SPSS 10.0 for Windows SAMPLE SESSION

Cross-Sectional Analysis

Short Course Training Materials

Designing Policy Relevant Research and

Data Processing and Analysis with SPSS 10.0 for Windows

4th Edition

Department of Agricultural Economics, Michigan State University

East Lansing, Michigan

March 2000

Components of the Cross-Sectional Training Materials

Section 0 - Introduction to the file structure for SPSS 10.0 for Windows (Data and Syntax Editors and Viewer (Output Navigator)). Must be read before starting the sample session

Section 1 - Basic functions

Section 2 - Table Lookup & Aggregation

Section 3 - Tables & Multiple Response Questions

Section 4 - Graphs, tables, publications and presentations, how to bring them into word processor.

Annexes

1. - Presentation of filters versus permanent selections, and graphing and data in chart options, SPSS 10.0 for Windows

2. - Six pages from the socio-economic survey of the smallholder survey in the Province of Nampula, Mozambique (NDAE Working Paper 3, 1992).

3. - Computer analysis of survey data - File organization for multi-level data by Chris Wolf, MSU Department of Agricultural Economics. This is downloadable as a separate document in English or French at http://www.aec.msu.edu/agecon/fs2/survey/index.htm.

Acknowledgments

Funding for this research was provided by the Food Security II Cooperative Agreement between the Department of Agriculture Economics at Michigan State University and the United States Agency for International Development, Global Bureau, Office of Agriculture and Food Security .

SPSS for Windows SAMPLE SESSION
SECTION 0 - Levels, time series and file structure for SPSS 10.0 for Windows
(Data, Syntax and Output windows)

Short Course Training Materials
Designing Policy Relevant Research and
Data Processing and Analysis with SPSS 10.0 for Windows
4th Edition

Department of Agricultural Economics, Michigan State University
East Lansing, Michigan
March 2000

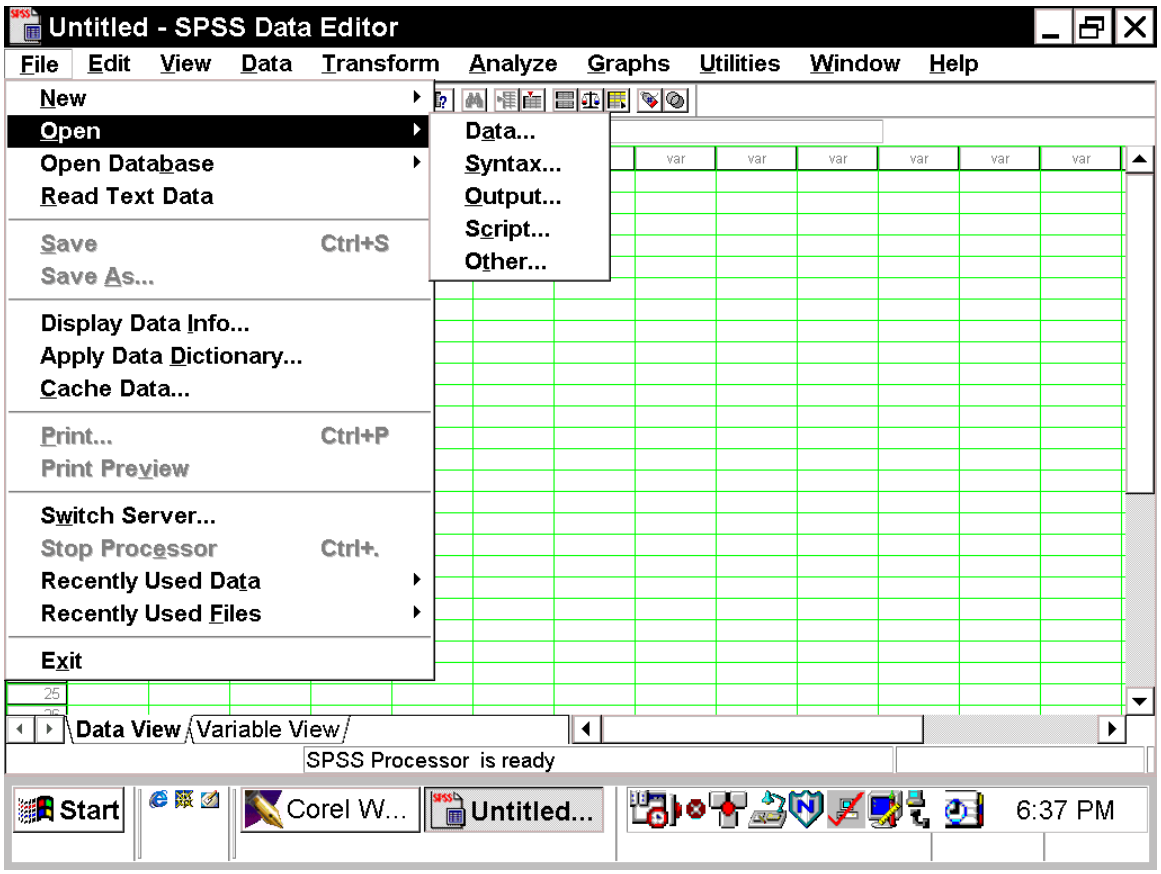
The following module introduces the basic concepts of levels, the notion of cross section analysis, and consequently, the methods of data organization. This module gives a brief description of the file structure of SPSS for Windows version 10.0. It is essential that you read through this module before starting the cross section session.

Files Used In SPSS 10.0 for Windows

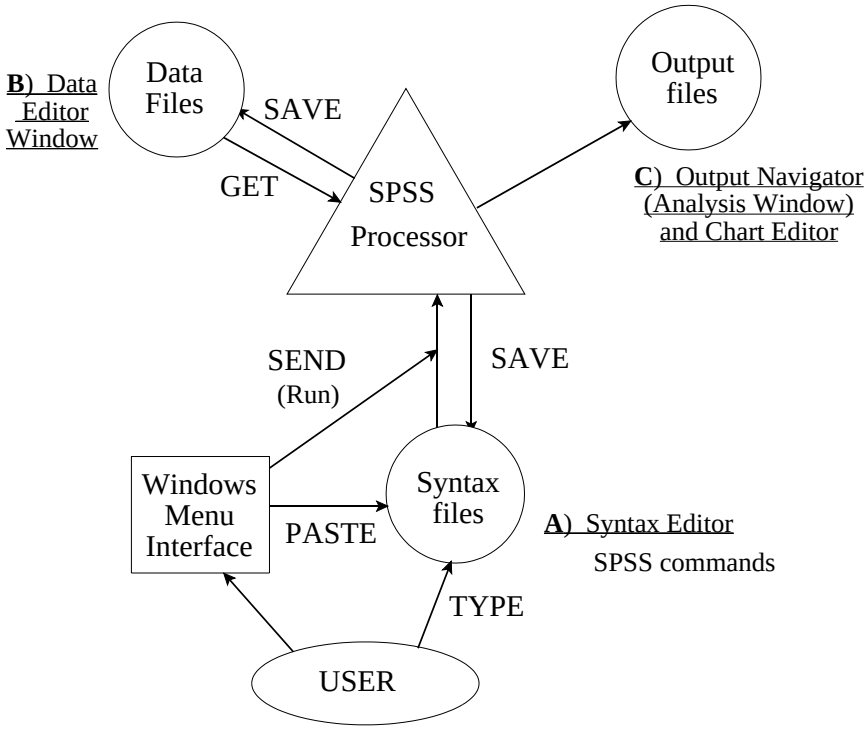
While using SPSS 10.0 for Windows in the manner taught in this tutorial, you are dealing with three different windows within the program—the Syntax Editor, the Data Editor window and the Viewer (including charts), the contents of each can be saved into the appropriate SPSS 10.0 for Windows file type.

When you open SPSS 10.0, in the upper left hand corner of the window, select **File**, then **Open** and you will have 5 options of file types from which to select:

Data	data files	(Extension *.sav)
Syntax	syntax or command files	(Extension *.sps)
Output	output files	(Extension *.spo)
Script	advanced programming files for use with Sax BASIC that are created automatically each time an **Output** is created	(Extension *.sbs)
Other	all files	(*.*)



It is important to recognize the significance of the different types of files and to understand the various commands you use to create and access the files.



A) The Syntax Editor

The Syntax Editor is the window where syntax or commands are written before they are submitted to the SPSS processor. To put commands in the Syntax Editor you can **type** the commands directly into the Syntax Editor or you can use the pull down menus and select **Paste** when you are finished customizing the command. There are four main uses of the Syntax Editor:

- To type commands directly or to paste commands from the Data Editor to be processed later by SPSS 10.0 for Windows,
- To send these commands to SPSS 10.0 for Windows for processing,
- To write or save these commands to a file for future use, and
- To retrieve files of commands that you have saved previously.

It is important to understand that the commands you put in the Syntax Editor will not be executed (no output will be produced) until you send the commands to the processor. The Syntax Editor is simply an area that helps you prepare the commands. To send the commands to the processor, you use the **Run** button in the Syntax Editor window toolbar of SPSS10.0 (or select **Run ... Current** from the Menus). Once you press the **Run** button, the computer sends the command(s) to the processor, which reads the commands written in the Syntax Editor and executes them. When all the commands have been processed, SPSS opens the Viewer to examine the results of your command. You can then switch back to the Syntax Editor and add new commands or edit old ones and execute these changes to observe different results.
It is good to start viewing the syntax of commands by using the **Paste** option rather than the **OK** option from the choices when you create a command in the Data Editor. If you wish to display the commands in your output file, do the following:

From the Menu (in any of the windows), select

Edit

Options ...

- Select the **Viewer** tab
  - Click in the box next to **Display commands in log**
  - Click on **OK** or **Apply**

When you have successfully completed each step in your analysis (or when you are ready to end an SPSS 10.0 for Windows session, even if it was not completely successful) you should save the commands to a file for future use. To save the commands, make the Syntax Editor active and select **Save** from the File menu. A file created from the Syntax Editor is called the *syntax (or command) file*. It is a file containing only commands; it never contains any of the data you may be analyzing with the commands. You must save your data separately, as described in the following section. We suggest that you use the default *extension* of `.SPS` when naming syntax files. `REP7.SPS`, `DEM-ALL.SPS`, and `SECTION1.SPS` are some examples.

By writing your commands to a syntax file, you can retrieve, look at, or modify sets of commands and rerun them. You can retrieve a syntax file by pulling down the **File** menu from any of the SPSS windows and selecting **Open**. Select **Syntax** and retrieve the filename under which you had last saved the file. Once you have opened a specific file, you can use the commands from the file, without having to recreate or type them again. If you make changes to the Syntax file that you wish to keep, make sure you save them to disk again.

B) The Data Editor Window

SPSS 10.0 for Windows stores your data in a *data file*. In addition to the values themselves, a data file contains such things as variable labels and value labels, formatting information, missing-value specifications, etc. Before you can do any data analysis in SPSS 10.0 for Windows, you must first tell SPSS to open a Data file. First select **File** from the menu, select **Open**, **Data** and highlight a data file. You have two choices at this point: 1) click on **Paste** to paste the command to the Syntax Editor and then run the command, or 2) run the command directly from the dialog box by clicking on the **Open** button). After running this command, the data in the file is available to SPSS 10.0 for Windows in the Data Editor window.

Two views of the data are available in the Data Editor window. **Data View** displays the actual values for the variables in the data file. **Variable View** displays the data dictionary which includes variable labels, value labels data type and other information. To switch between the views, click on the tab at the bottom of the screen.

You will often get a data file, compute new variables, make transformations, and finally save the modified set of data to use at another time. For example, you might retrieve a data file with land area per crop, add to it production per crop from another file, and then calculate yield. If you want to use these new production and yield variables at a later time, you must make sure that the data file is saved with the new variables in it. To save a data file, make the Data Editor the active window, select **Save As...** from the **File** menu and give the file a new name. Note, you **must** be in the Data Editor window to save your data unless you run a **SAVE OUTFILE** command from the Syntax Editor. You may choose to write over the old file by saving the file to the same file name.

C) The Viewer

SPSS 10.0 for Windows automatically writes all messages and output that result from the execution of your commands to the Viewer. For example, if you run a frequency command, then the frequency table you specify will be written to the Viewer. Similarly, if you generate a table or a graph, the table or graph will appear in the Viewer. To save the contents of the Viewer to a file, make the Viewer active, pull down the **File** menu and select **Save As...**. When you give the file a name, SPSS will automatically attach the *extension* `.SPO`. It is very important to save the *output file*. The Output file gives you access to your results after your SPSS 10.0 for Windows session has ended. For example, you can print the output of your session in order to examine the results and verify for errors. In the sample session, you will see how to save the contents of

the Viewer and give the file from each session a different name. One final note, you can manipulate the output produced just as if you were using a file manager (called Windows Explorer). In the Viewer, there are two panes: the one on the right contains the results, the one on the left shows an outline view of the contents. From within this pane, you manage the results by copying, moving or deleting the results, hiding a table or chart, renaming titles, inserting titles or text or a chart.

Summary of the Basic File Types

Syntax files (or command files) contain commands saved in the Syntax Editor. They do not contain output or data—only commands. Syntax files are made accessible to SPSS for Windows with an **Open..Syntax** command. Like in SPSS 6.1.3., the extensions are *.SPS (was *.LOG in SPSS/PC+ (DOS)).

Output files contain statistical output, data information and presentation (tables, graphs, charts), generated by the SPSS 10.0 for Windows processor, given selected commands. They do not contain data. Output files are made accessible to SPSS for Windows with an **Open..Output** command. The new extensions are *.SPO (was *.LIS in SPSS/PC+ (DOS) and *.LST in SPSS 6.1.3).

Data files contain data, including original survey variables plus new created variables through various SPSS 10.0 for Windows commands such as the `COMPUTE` or `AGGREGATE` commands. Data files are made accessible to SPSS for Windows with an **Open..Data** command. For SPSS 6.1.3. and 10.0., the extensions are *.SAV (was *.SYS for SPSS/PC+).

SPSS for Windows SAMPLE SESSION
SECTION 1 - Basic functions: SPSS files, Descriptives and Data Transformation

Introduction

This is a self-paced training aid designed to introduce the commands needed for some typical statistical survey analyses using **SPSS 10.0 for Windows**. This tutorial is intended to be a stand-alone training tool. To use it most effectively, you should ask a knowledgeable SPSS for Windows user to help you get started and to answer questions as you work independently through the session. It can also be used as a guide for classroom training.

A copy of the questionnaire on which the data is based can be found in the Mozambique project 1992 **NDAE Working Paper 3: A Socio-economic survey of the smallholder survey in the province of Nampula: Research Methods**, copies of the three tables which were made available and can be found at the end of the manual in the annex section (for further information please contact Dr. Michael Weber at webermi@pilot.msu.edu). Four portions of the questionnaire are referenced, each of which has a corresponding SPSS for Windows data file. Two other SPSS for Windows data files are required for conversion of units of measure.

Questionnaire Section	SPSS for Windows Data File
Main Household Section	C-HH.SAV
Table IA: Household Member Characteristics	C-Q1A.SAV
Table IV: Characteristics of Production	C-Q4.SAV
Table V: Sales of Farm Products	C-Q5.SAV
Conversion factors for computing kilograms	CONVER.SAV
Conversion factors for computing calories	CALORIES.SAV

This training consists of four sections, each of which should take approximately two hours. We recommend that you complete each section in a single sitting. These tutorial materials make the following assumptions:

- You know how to use Windows with a mouse
- The six data files listed above are stored in the directory c:\sample on your hard disk. If you have not done so already, you need to copy the files from sample.zip to this directory.
- Under **Options** in the **Edit** Menu the following items are set:
  list variables in the same order they are listed in the file
  list commands in the output window
  display the variable names rather than the variable labels
  Syntax Editor does not come up at the start of the session.

You can modify any of the settings that control how SPSS works from this screen as well.

Important: *Always remember to SAVE the changes to the data after each exercise and module, using a* ***new*** *file name. Also, you should save the syntax files and output files created during each session, using logical names, such as module1.sps or session1.spo. If you are not sure of any of the above, ask the person helping you to check them or check with the nearest computer service center or specialist.*

Open your SPSS software. If you have not read or completed Section 0, please do so now to clarify the concept of the **Syntax Editor**, where you **paste** or type commands, the **Viewer** where SPSS for Windows displays the results of your commands and the **Data Editor** window where the working data file and variable information are displayed.

Data Files and the Working File

Data from questionnaires that has been entered into SPSS 10.0 for Windows is stored in what are called *data files*. If we want to work with a set of data, we must open the corresponding data file, so that it is available to the program.

When a data file is opened, it is loaded from the disk into memory (the computer's "RAM"), making it the working file. This means that the data from this file is now available for you to use. Let's start with the questionnaire for Table IA: Household Member Characteristics. The data file that corresponds to it is C-Q1A.SAV. To open this file, perform the following steps:

1. From the **File** menu, select **Open...**, select **Data**
   This will open the Open File *dialog box.*
2. Change to the directory where your sample session data are and select the file c-q1a.sav.
3. Click on the **Paste** button to place the command in the Syntax Editor.
   The Syntax *Editor will now become the active window and you will see the text*
   ```
   GET
     FILE='C:\sample\C-Q1A.SAV'.
   ```
   in the Syntax *Editor.*
4. Click on the Run ▸ button on the Toolbar.

   Note that the `GET FILE` *command you just ran will be written to the Viewer.*

The data editor becomes the active window and the household-member data file is now in memory.

One key thing we often want to know about a data file is what variables it contains. We can find this out, along with other information, by using the **Variables...** command on the **Utilities** menu, in both the Syntax Editor and the Data Editor. It lets you browse through the variable definitions and variable labels. To do this, perform the following steps:

1. From the **Utilities** menu select **Variables...**
2. Select a variable name - the information about that variable will appear to the right.

This display shows definition information about each of the variables. We see the variable name, **district**, **vil**, **ca1**, **ca2**, **ca4**, **ca5**, **ca6**, and **univ**, the value labels for variables, the type of variable (numeric, string, date, etc.), the display width of the variable in characters, the number of decimal places (if Type is Numeric), and any values defined as user missing values.

Click on the **Close** button when you are finished exploring this window.

To write all of this information to your Viewer for later examination, do the following:

Pull down the **Utilities** menu and select **File Info**.
This command will execute immediately. The Viewer will become active and will contain a listing of all the variables with their definitions.

You can see the name of each of the variables, their labels, and the various formats, e.g. F8.2 means width 8 with two decimal places. This tool is an excellent way to begin to document the contents of the data file. You can copy this information to a word processing file to begin the documentation process.

If you want to look at the structure of each variable, there is a new way in SPSS 10.0. In the Data Editor window, select the **Variable View** at the bottom left on your screen, rather than the Data View. You can directly change the characteristics of your variables here, just as you can change values in your data in the Data View window. The example of the variable DISTRICT is shown in Table 1.1 on the next page, with a brief explanation of the choices in each column.

If you want to modify one of the parameters about a variable, click on the cell. If there are specific choices to be made, a small shaded box will appear in the right corner for that specific cell. Click on the box. in order to see the choices, add a new possibility, or view the other options. In some cases, as for Width, Decimals, and Column, instead of a box, arrows are shown to increase or decrease the size.

Example: For the variable **DISTRICT**, click on the column **Values**. Click in the cell for the variable (DISTRICT). You will see a small gray box  Click on this box.

A dialog box appears entitled: Value Labels

To add a new label of 4 associated with Nampula,

- enter **4** in the **Value** box and press the <tab> key,
- then enter Nampula in the **Value Label** box,
- click on the **ADD** bottom .
- Usually, you would select OK, but we don't want to keep this change so select Cancel.

You can use these steps to modify or delete an existing label. Highlight the specific label and then click **Change** or **Remove**.

Table 1.1. Basic Structure of Variable View in SPSS 10.0

Number of the variable	**Name**	**Type**	**Width**	**Decimals**	**Label**	**Values**	**Missing**	**Columns**	**Align**	**Measure**
Explana-tion	Variable name	Numeric or alpha-numeric (*String*)	Space required to write variable in the data set	Number of digits to the right of the decimal	Label for the variable	Labels for the values, e.g. labels for categorical variables,	Declared user missing values (example: -99), indicates cases that should be excluded from calculations	Display width of variable in Data View	Alignment of the data in Data View only: *Left, Right Center*	Measure-ment level of variable: *Scale, Nominal, Ordinal*
Example:										
1	district	Numeric	1	0	DISTRICT	1= MONAPO 2= RIBAUE 3= ANGOCHE	None	8	Right	Scale

[1] There are three categories of measurement level:

Scale: These are variables with values that are generally continuous or in intervals (integers) (e.g.: yield or age);
Ordinal: Values or alphanumeric variables that consist of categories with an intrinsic ordering (e.g. 1= short; 2=medium; 3 = tall);
Nominal: Values or alphanumeric variables that consist of categories with no intrinsic ordering **(eg. 1=man; 2=woman).**

Descriptive Statistics - involving one variable

The first thing to do when starting analysis is to get descriptive statistics (e.g. averages, maximum, minimum, and standard deviations) for all variables. This type of analysis helps you to find data entry errors, to give you a "feel" for what your data is like, to see that missing values have been defined correctly, etc. It may be tempting to skip this step for some data sets or for some variables, but this is an important step that will almost always save time later and improve analysis. For example, finding out the average age of all respondents may not be something you are interested in knowing, but if the average age turns out to be 91.3 yrs, this would alert you that something is probably wrong.

Basic descriptive statistics can be obtained from two common SPSS for Windows commands—**Descriptives** and **Frequencies**. **Descriptives** is used for continuous variables, while **Frequencies** is used for categorical variables.

A *continuous variable* is a variable that does not have a fixed number of values. A *categorical variable* is a variable that has a limited number of values that form categories. For example, look at the Annex Table IA: Household Member questionnaire. Variable **ca3** (age) is a continuous variable because age can take on many different values. Variable **ca2** (relation to head) is a categorical variable because its values are limited to the categories 1-6.

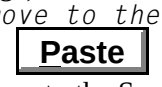
Start by examining the data in the file. Use the Data Editor window to scroll through your data file. To do this, perform the following steps:

1. Click on the Go To Data Editor  button on the Toolbar.
2. Scroll through the data.
   A period in a field indicates a missing value or sysmis.

This will give you a "feel" for what your data is like. It might also help point out obvious errors, e.g. a variable whose values are missing for all listed cases. Decide which of the variables are continuous and which are categorical (normally you would refer to the questionnaire to make this decision). You need to know this in order to select the right procedure to use for each variable. If you mistakenly perform a **Frequencies** on a continuous variable, you will probably get more output than you really want, with possibly hundreds of different "categories", one for each different value found. If you perform a **Descriptives** on a categorical variable, you will usually get meaningless results, since the average value of a variable that consists of categories has no real significance.

Descriptives

By examining the data, you should have found that variable **ca3** (age) is continuous and the remaining variables are categorical. To run descriptives on **ca3,** do the following:

1. From the **Analyze** menu select **Descriptive Statistics....Descriptives**
   This will give you the Descriptives *dialog box*
2. Select **ca3** (age) from the list on the left and click on the  button.
   ca3 will move to the **Variable(s)**: *box on the right*
3. Click on the  button to put the command into the Syntax window and automatically go to the Syntax window. (If the Syntax Editor did not become active, you can go there by clicking on the syntax button on the windows taskbar at the bottom.)
4. Execute the command by clicking on the Run  button located on the Toolbar. (Note that this time we did not have to move the cursor since it was already positioned in one of the lines of the **Descriptives** command.)

The Viewer will become active and the results of the command will be there. You will see that the mean for age (**ca3**) is 21.34 years.

5. Another useful way to examine a continuous variable is to run a Frequency command to view a histogram and the distribution of a variable. From the **Analyze...Descriptive Statistics...**select **Frequencies** .
6. Select **ca3** (age) from the list on the left and click on the ▶ button.
7. Click on **Charts** button and select Histograms, click on the **Continue** button.
8. Click on **Paste** to put the command into the Syntax Editor and make it active.
9. Execute the command by clicking on the Run ▸ button located on the Toolbar. View the distribution of ages in the data.

Save the Output File

Now that you have output in the SPSS Viewer, it is a good time to save that output file. Go to the Viewer window. Click on the **File...Save as...** on the SPSS toolbar at the top right. In the "File Name" box, type **Session1**. Make sure that the directory is the one where you want save the output. SPSS will automatically add the extension **.SPO** to indicate an output file.

Frequencies

Since the variables **ca1** (work on a farm or not), **ca2** (relation to head), **ca4** (sex), **ca5** (level of schooling) and **ca6** (marital status) are categorical, we will run a **Frequencies** on them. To run frequencies, do the following:

1. **Analyze...Descriptive Statistics...**select **Frequencies ...**
   This will give you the **Frequencies** *dialog box.*
2. Click the **Reset** button to clear the Variables box**.**
3. Select **ca1** from the list on the left and click on the ▶ button.
   ca1 *will move to the* **Variable(s):** *box on the right*
4. Repeat step 3 until **ca2, ca4, ca5** and **ca6** have all been moved to the **Variable(s):** box.
5. Click on **Paste** to put the command into the Syntax Editor and make it active.
6. Execute the command by clicking on the **Run** ▸ **b**utton located on the Toolbar.

The Viewer will become the active window. You will see, for example, the results for ca1 show that 70.7% of the household members work on a farm. The results for ca6 show that 38.0% of those surveyed are in monogamous marriages.

For a complete description of the output you receive from Descriptives and Frequencies refer to the SPSS for Windows Base System User's Guide Release 10.0 , pages 213-221.

Explore

Another command used to produce many types of descriptive statistics is the Explore command. One of the most useful outputs for this statistic is that it produces a list of data that can be considered as *outliers*. The Explore command can produce large amounts of output if used with its defaults. We will limit the output to statistics.

Run the **Explore** command on the variable **ca3** (age) using the following steps:

1. From the **Analyze...Descriptive Statistics** menu select **Explore...**
2. Select **ca3** from the list on the left and click on the ▶ next to Dependent List.
3. In the lower left corner of the dialog box is a box called Display. Click on the radio button (circle) next to Statistics.
   This will give us statistics only and no plots.

4. Next click on the **Statistics...** button.
   This will bring up the Explore: Statistics *dialog box.*
5. Click once on the square next to Outliers to put an ✓ in the box.
   You will notice there is already an ✓in the box next to Descriptives.
6. Click on the **Continue** button.
   This will bring you back to the Explore *dialog box.*
7. Click on **Paste** to put the command in the Syntax Editor and make it active.
8. Click on ▸ Run.

You see the Descriptives Table which shows you the standard descriptions and the Extreme Values table which shows you the five highest and five lowest values occurring for age (**ca3**). You can then determine if these values can be considered as *outliers*. The cases are identified by the case number. Refer to pages 223-230 of the SPSS for Windows Base System User's Guide Release 10.0 for an explanation of the **Explore** command.

Save the Syntax File

It is a good practice to frequently save your syntax files while you are working. You may need to re-run the commands on the same file after correcting a data entry error or if your computer "crashes" due to a problem with SPSS or another program. To save the file, make the Syntax Editor window the active window, select **File...Save as...** on the SPSS menu at the top right, in the File Name box, type the name **Session1**. It is useful to save the syntax file and the corresponding output file with the same name, just changing the extension. SPSS will automatically add the .SPS extension to the syntax file. Verify that the directory is the correct one. You must be in the Syntax Editor window to save the syntax file.

Apply what you've just learned about descriptive statistics by doing the following exercise.

Exercise 1.1: Run descriptive statistics on another sample file. Use the production questionnaire - Table IV, whose data is in file C-Q4.SAV.
Hints:

a. make C-Q4.SAV your working data file.
b. Use the **Descriptives** command for continuous variables, and **Frequencies** for categorical variables.
c. **Prod** is a categorical variable.
d. Quantities (**p1b**, **p2b**, ...) are continuous variables.
e. Units (**p1a**, **p2a**, ...) are categorical variables.
f. **p4** (month in which stocks ran out last year) **& p6** (month in which stocks will run out this year are categorical variables.

A small sampling of what you should find from running these frequencies and descriptive statistics follows:

PRODUCT

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	cotton	83	4.9	4.9	4.9
	peanuts	144	8.5	8.5	13.4
	rough rice	155	9.2	9.2	22.6
	bananas	50	3.0	3.0	25.5
	sweet potato	12	.7	.7	26.2
	cashew liquor	24	1.4	1.4	27.6
	sugar cane liquor	11	.6	.6	28.3
	dried cashew	2	.1	.1	28.4
	sugar cane	13	.8	.8	29.2
	cashew nut	130	7.7	7.7	36.9
	coconut	45	2.7	2.7	39.5
	beans	279	16.5	16.5	56.0
	manteiga beans	7	.4	.4	56.4
	sunflower	5	.3	.3	56.7
	oranges	13	.8	.8	57.5
	cashew fruit	44	2.6	2.6	60.1
	manioc	338	20.0	20.0	80.0
	sorghum	124	7.3	7.3	87.4
	maize	192	11.3	11.3	98.7
	"ossura"	5	.3	.3	99.0
	tobacco	4	.2	.2	99.2
	tomato	13	.8	.8	100.0
	Total	1693	100.0	100.0	

Descriptive Statistics

	N	Minimum	Maximum	Mean	Std. Deviation
PROD THIS YR - # OF UNITS	1670	.0	5000.0	26.353	163.436
PROD NORMAL YR - # OF UNITS	1598	.5	5000.0	22.815	159.510
STOCK ENTERING HARVEST - # OF UNITS	173	.0	30.0	2.523	4.575
STORED FOR CONS THIS YR - UNIT	1227	1	9	2.25	1.82
STOCK FOR SEED - # OF UNITS	869	.0	100.0	4.938	6.876
Valid N (listwise)	147				

Descriptive Statistics - involving two or more variables

Crosstabs

The **Crosstabs** command produces tables showing the distribution of cases according to their values for two or more categorical variables.

Look at the household member questionnaire in the annex section, Annex Table IA. One thing you might be interested to know is how the gender of the respondents varied by their relationship to the head of household. This would tell you, for example, how many females are heads of households. The **Crosstabs** command will produce this type of summary. Make the household member file, C-Q1A.SAV, the working data file.

1. Click on the open folder button in the top left of the Data Editor Taskbar
2. Select the file c-q1a.sav.
3. Click on **Paste** to place the command in the Syntax Editor and make it active.
4. Place the cursor anywhere on the line containing the "GET" command and click on the Run button on the Toolbar.

To use the **Crosstabs** command do the following:

1. Select **Analyze...Descriptive statistics** from the menu**.**
2. Select **Crosstabs...**
   This will bring up the **Crosstabs** *dialog box.*
3. Select **ca2** (relation to head) from the list on the left and click on the ▶ next to Row(s):
4. Select **ca4** (sex) from the list on the left and click on the ▶ next to Column(s):
5. Click on the **Cells...** button
   This will bring up the Crosstabs: Cell Display dialog box
6. In the **Counts** section, click on the box next to **Observed** to place an ✓ in it, if there is not already one there.
7. In the **Percentages** section click on the boxes next to **Row** and **Column** to put ✓'s in them.
8. Click on **Continue**
9. Click on **Paste**
10. Run the command in the Syntax Editor**.**

The Crosstabs:Cell Display dialog box specifies which statistics you want displayed in each cell of the table—in this case we wanted counts, row percentages, and column percentages. (Row percentages sum to 100 across all the cells in a row, while column percentages sum to 100 across all the cells in a column. By default the **Crosstabs** command just gives counts.) The table produced by this command tells you that there are 21 female heads of households, and that 6.1% of the total number of heads of households are female.

Means

The **Compare Means** command is somewhat similar to **Crosstabs,** but it gives statistics about continuous variables. It shows how the mean and other statistics for a continuous variable differ by the values of one or more categorical variables. Another way to look at the relationship between **Crosstabs** and **Compare Means** is that, **Crosstabs** is a way of getting **Frequencies**-type output broken down by categories of one or more other variables, while **Compare Means** is a way of getting **Descriptives**-type output broken down by categories of one or more other variables.

Suppose we want to know how the age of the respondents varied by their relationship to the head of household. If we did this with **Crosstabs** we would get a table with dozens of cells for the different ages represented, which would be an unusable format. Instead we will use **Compare Means.**

1. Select **Compare Means** from the **Analyze** menu
2. Select **Means...**
3. Select **ca3** (age) and click on the ▶ next to Dependent List:
4. Select **ca2** (relation to head) and click on the ▶ next to Independent List:
5. Click on **Paste**
6. Run the command from the Syntax Editor

This command will calculate means of the dependent variable (age), which should normally be a continuous variable. The means will be calculated separately for each different value of the independent variable, which should be a categorical variable, such as relation to household head.

From this output you find that the average age of heads of households is 41.5 years while the average age of their spouses is 33.2 years.

Data Transformations

After examining the results of the descriptive statistics you will often want to do data transformations. A data transformation is an operation that takes existing variables and either changes their values in a systematic way or uses their values to calculate new variables. The following example shows a common data transformation; the conversion of a continuous variable to a categorical variable.

The information we received from the **Means** command is interesting, but it might also be useful to see the actual distribution of the ages into groups or categories, so we can tell, for example, how many heads of household are older than 60. Since the age variable, **ca3**, is continuous, we cannot do this directly—first we have to transform it. Let's suppose we're interested in four categories: 0-10 years old, 11-19 years, 20-60 years, and over 60 years of age.

Recode

To categorize a variable, you use the **Recode** command. Categorizing a continuous variable makes detailed information more general. If you want to keep the detailed information as well as the new general information, you must recode the variable into a different variable. If you recode into the same variable the original values will be lost.

In this particular file, if you use the **Recode Into Same Variable** command to transform **ca3** (age), **ca3** will take on the new categorical values assigned in the **Recode** statement, and the original ages will be lost. Since we want to preserve the original ages and store the categorized values in a separate variable, we will **Recode Into A Different Variable.**

Let's **Recode** the variable **ca3** into a new variable called **age_gp** for age groups**.**

1. Select **Recode** from the **Transform** menu
2. Select **Into Different Variables...**
3. Select **ca3** from the list on the left
4. Click on the ▶ next to **Input Variable -> Output Variable:** box
   ca3 should move to the **Input Variable->Output Variable: box** *and the name of the box will change to* ***Numeric Variable -> Output Variable.***
5. Click once in the empty box next to **Name:** in the **Output Variable** section to put the cursor there.
6. Type **age_gp** in the box.
7. Click once in the empty box next to **Label:** in the **Output Variable** section.
8. Type **Age Group** in the box.
9. Click on **Change** to have the variable name and label changes take effect.

10. Click on **Old and New Values...**
    The **Recode into Different Variables: Old and New Values** *dialog box will appear.*
11. In the Old Value section click on the circle next to **Range: [ ] through [ ]**
    Your cursor should be in the first box.
12. Type **0** in the first box
13. Press **<Tab>** and type **10** in the second box.
14. Press **<Tab> twice.**
    Your cursor will now be in the box next to **Value:** *in the* **New Value** *section.*
    OR you may press the "Alt" key leaving your finger on the key while you press the "l" key to bring you to the "New Value" box.
15. Type **1** for the first age group.
16. Click once on **Add**
17. Click on the first box after **Range:** and repeat steps 11 through 16 to recode ages **11 thru 19** to **2** and ages **20 thru 60** to **3.**
18. To recode ages **61** and up to **4,** click on the circle next to **Range: [ ] through highest**
19. Enter **61** in the box and repeat steps 14 through 16 using 4 for the value.
20. Click on **Continue**
21. Click on **Paste**
22. Select the following text in the Syntax Editor

```
RECODE
  ca3
  (0 thru 10=1)  (11 thru 19=2)  (20 thru 60=3)  (61 thru Highest=4)  INTO
  age_gp .
VARIABLE LABELS age_gp 'age group'.
EXECUTE .
```

23. Run the command

Recode changes the values for **age_gp** to the codes we want to use—1,2,3, and 4. We will switch to the Data Editor to view that the changes were made.

To switch to the Data Editor window (*we will use a different method than we used earlier*):

1. Click on **Window** from the menu and select **c:\sample\q1a.sav**.
2. Scroll through the window with the scroll bars.

SPSS's standard format for displaying a numeric variable includes two decimal places, which is inappropriate for a variable we know will always have an integer value. To change the display format of **age_gp** to the same format as our other variables, go to the Variable View window.

1. Switch to the Data Editor window if you are not already there.
2. Select the Variable View window from the bottom left bar.
3. Click once on the box for the variable **age_gp** on line 12 in the column for Type. "Numeric" will appear in the box.
4. Click on the small gray box **...** that appears to the right in the small box and the Variable Type box for the variable **age_gp** will appear.
5. In the box next to Width: type 1.
6. In the box next to Decimal Places: type 0
7. If the circle next to Numeric is not selected, select it.
8. Click on **Continue**
   You do not have the option to **Paste** *this command and the changes are automatically made when you select.*

You could also have changed the size of the variable in the "**Width**" and "**Decimals**" column. You must first change the decimals to 0 and then change the width to 1.

These changes tell SPSS for Windows to display **age_gp** with a width of 1 digit and no decimal places. When you **Recode** a new variable, it does not have *Value Labels*. The statistical output from SPSS always includes the names of the variables being analyzed, but sometimes the name of a variable does not tell us as much as we would like to know. Since names are limited to eight characters, they may not be descriptive enough for us to remember the complete question from the questionnaire (e.g. the variable **ca1** is work on a farm or not). The name also does not tell us what the individual values of a categorical variable refer to (e.g.**ca4** is sex and a value of 1 indicate man and 2 indicates woman) . To make the output more understandable, we add *Variable Labels* and *Value Labels*. To avoid confusion and mistakes, you should always add labels for any computed variable that you are going to save for later use. The best time to add labels is immediately after you create the new variable, because if you postpone it you may forget. The **Recode** command facilitates this by allowing you to add the **Variable Label** when you do the recode. To add the **Value Labels** use the following steps:

1. You should still be in the *Variable viewer* window from the last set of steps.
2. In the box in the **Label** column for the variable **age_gp**, you should see the text "Age Group" because it was included in the command..
3. If there is no text in the Label: box, enter the text "Age Group" there.
4. Go to box for **age_gp** in the Values column, where it says "None".
5. Click on the small gray box [..] once to enter the Value Labels dialog box.
6. Type **1** in the Value box, then hit <Tab> to move to the Value Labels box and type **0 to 10** in that box.
7. Click on [Add]
   You will have noticed there are two other options available as well, [Remove] to delete a value and value label set, and [Change] to modify the value label for a specific value.
8. Repeat steps 6 and 7 using the following information:

Value:	Value Label:
2	**11 to 19**
3	**20 to 60**
4	**61 and older**

9. Click on [OK]
10. Go to the Data View window and you will see that **age_gp** is now displayed as a single digit.
11. Select **Variables...** from the **Utilities** menu.
12. Click on **age_gp** to verify the changes you just made.
13. Click on [Close] when you are finished.

This new variable is not yet part of the data file stored on disk. We must save the file in order for this variable to be included permanently in a new data file. It is a good practice to save a file under a different name in case we want to go back to a previous version of a file. For this reason we will use the **Save As** command from the **File** menu with the new file name **Q1A-AGE.SAV**.

1. Make sure the Data Editor window is the one in front (the active window).
2. From the **File** menu select **Save As...**
   The cursor should be in the box under File name: *above the Save as type: SPSS (*.SAV) drop-down box. Typing while that area is highlighted will wipe out the current text.*
3. Type **q1a-age** (The .sav extension will be added automatically.)
4. Paste and run the command.

Now each time the data file Q1A-AGE.SAV is opened, the **age_gp** variable will be included.

You might want to analyze this new categorical variable using the Crosstabs command to determine how many people in each age group are heads of households, spouses, or children.

1. Use **Analyze...Descriptive Statistics... Crosstabs**... from the menus.
2. Use **age_gp** for Rows and **ca2** (relation to head) for Columns.
3. Check the proper selections in the Cells choices at the bottom, for we want both Row and Column percentages.
4. Paste the command and run it.

From this, you can see that 12% of heads of households are 61 years of age or older. Also, that of the people 61 years or older, 83.7% are heads of households.

Compare the information you get from this **Crosstabs** analysis with the information from the **Compare Means** command performed on **ca3** (age) earlier. To do this, we will explore SPSS's ability to switch between the Syntax, Viewer, and Data windows.

To switch to the Viewer:

1. From the **Window** menu select Session1 - SPSS.Viewer
2. Scroll back through the window with the scroll bars.
3. Find the Crosstabs table and compare with the Compare Means table.

To switch to the Syntax Editor:

1. From the **Window** menu select Session1 - SPSS Syntax Editor.
2. Scroll through the window with the scroll bars.

To switch to the Data Editor:

1. From the **Window** menu select q1a - SPSS Data Editor.
2. Scroll through the window with the scroll bars.

Please note it is also possible to switch from one window to another by clicking on the SPSS icons in the Windows taskbar, found by default at the bottom of the screen (the taskbar may be moved to any sides of the screen).

Apply what you have learned about data transformations and descriptive statistics by doing the following exercise.

Exercise 1.2: Using the Household Data and Questionnaire (latter available in the annex), find out the number of households in each district that have 1-4, 5-7, and more than 7 persons per household. One way to find out this information is to create the following table.

Hints:
a. Use the file C-HH.SAV.
b. Recode **h1** into **hhsize** using the following groups: (1 thru 4) (5 thru 7) (8 thru Highest).
c. Add a variable label and value labels.
d. Run **Crosstabs** on this variable by **district**.

Household size * DISTRICT Crosstabulation

			DISTRICT			
			MONAPO	RIBAUE	ANGOCHE	Total
Household size	1.00	Count	65	48	74	187
		% within Household size	34.8%	25.7%	39.6%	100.0%
		% within DISTRICT	60.7%	40.3%	64.3%	54.8%
		% of Total	19.1%	14.1%	21.7%	54.8%
	2.00	Count	39	56	36	131
		% within Household size	29.8%	42.7%	27.5%	100.0%
		% within DISTRICT	36.4%	47.1%	31.3%	38.4%
		% of Total	11.4%	16.4%	10.6%	38.4%
	3.00	Count	3	15	5	23
		% within Household size	13.0%	65.2%	21.7%	100.0%
		% within DISTRICT	2.8%	12.6%	4.3%	6.7%
		% of Total	.9%	4.4%	1.5%	6.7%
Total		Count	107	119	115	341
		% within Household size	31.4%	34.9%	33.7%	100.0%
		% within DISTRICT	100.0%	100.0%	100.0%	100.0%
		% of Total	31.4%	34.9%	33.7%	100.0%

Looking at the results, you can see for Monapo for example, 34,8% of all 1 to 4 member households (group 1) are found within Monapo and that 60,7% of all households in Monapo have 1 to 4 members in a household.

Before exiting SPSS for Windows we should save the contents of the Viewer. The output window contains all of the command and the results of these commands. It is useful to keep this output in a file so you can review it later, print it or include it in a report.

1. Make the Viewer the active window using its icon in the Windows taskbar.
2 From the **File** menu select **Save As...**
3 Enter the filename **session1**
The .spo extension will be added to the name automatically.
4. Click on **Save**

To exit SPSS for Windows:

1. From the **File** menu select Exit SPSS
A dialog box will prompt you to save the contents of C:\sample\c-hh.sav
2. Click on **No**
A dialog box will prompt you to save the contents of Syntax Editor to Syntax1.
3. Click on **Save** and give it a filename such as Module1.sps
SPSS for Windows will exit.

For some types of analysis the data files may need to be restructured to a different level. The data from the four questionnaires—household, member, production and sales—are in four separate data files because the data are at different levels. The household data is at the most general, or highest, level - one case per household. The other three files contain more detailed data, which is usually thought of as being at a lower level - there are multiple cases per household. If you are not familiar with the concept of levels of data, read "Computer Analysis of Survey Data -- File Organization for Multi-Level Data" by Chris Wolf, before continuing on with this section.

The analysis we did in Section 1 was done at each level separately, using just the variables in a single file at a time. However, other types of analysis require combining data from more than one file. Let's look at an example.

Suppose we want to create a table of calories per adult equivalent produced per day from the principal food crops. Furthermore, we want to see how this varies by district and calorie-production quartile.

TABLE:1 Food Production in calories per adult equivalent per day

	Calorie Production Quartile			
Districts	1	2	3	4
Monapo				
Ribaue				
Angoche				

The data in their current form cannot answer the question, therefore, many transformations are required to produce this table. This is a typical example of the complications you will encounter in real-world data analysis. This entire section will be devoted toward the goal of creating this table.

To begin, let's first take a look back at some of the files that we have and at the variables we need to use from each of these:

- C-Q1A.SAV: This file contains data on household member characteristics. It is at the household-member level. We need to use the variables **ca3** (age) and **ca4** (sex) in this exercise to compute the number of adult equivalents per household.
- C-Q4.SAV: This file contains data on crops produced by the household. The variables we need to calculate the total production of the household are:

    a. **prod** - contains the codes for the agricultural crop produced.
    b. **p1a** - contains the codes for the unit in which the production was measured (100 kg sack, 50 kg sack, etc).
    c. **p1b** - contains the number of units produced this year.

Note that the unit of production is not a standard unit for each crop. For example, a "100 kg sack", as the term is used in Mozambique, weighs 100 kg only when the sack is filled with corn. When it is filled with manioc root, it weighs much less than 100 kg. Thus, we need *conversion factors* to be able to convert each of the units in which production was actually measured to our standard unit, which is the kilogram.

- CONVER.SAV: This is a *table-lookup file*. This file was created specifically to handle the problem of converting non-standard units to a standard unit. For each product-unit combination there is a conversion

factor to convert the measurement to equal the weight in kilograms. In other words, there is a different conversion factor for each product-unit combination. For example, the conversion factor for a 50 kg sack of rice is _____; for a 50 kg sack of cotton it is _____, while a 50 kg sack of manioc root is _____. The variables in this file are:

a. **prod** - product (crop) code
b. **unit** - unit of measure
c. **conver** - conversion factor (equal to the number of actual kilograms for the combination of **prod** and **unit**)

Below, a sample of data from CONVER.SAV shows that
rice (**prod**=7) measured in a 20 liter can (**unit**=8) weighs 19 kg;
rice (**prod**=7) measured in a 50 kg bag (**unit**=24) weighs 53 kg;
beans (**prod**=30) measured in a 20 liter can weighs 17 kg;
beans (**prod**=30) measured in a 50 kg bag weighs 47 kg.

prod (Product)	**unit** (unit)	**conver** (conversion factor)
...	...	...
7	8	19
7	24	53
...	...	...
30	8	17
30	24	47
...	...	...

- CALORIES.SAV: This also is a *table-lookup file*, created for convert kilograms of food into calories of food. It contains two variables:

a. **prod** - the product (crop)
b. **calories** - number of calories per kilogram of each of the crops

With this information in hand, we can now think about the specific steps we must take to create the table we want. Logically, there are three steps:

1. We need to know how many calories each household produced for the year. We can generate a file with this information using data we have stored in three places—the production file, C-Q4.SAV, and two table-lookup files, CONVER.SAV and CALORIES.SAV.
2. We need to know how many adult equivalents are in each household. We can generate a file with this information using data from the member file, C-Q1A.SAV.
3. We need to combine the results from steps 1 and 2 into one file so we can compute calories produced per adult equivalent per day.

Step 1: Generate a household level file containing the number of calories produced per household.

In executing this step, we must keep three things firmly in mind.

First, all production is currently measured in non-standard units whose weight is different for each product. Thus, we must first convert all production into kilograms.

Second, we want to know calories produced by each household, not kilograms. Thus, after converting all production to kilograms, we must convert it again to calories.

Third, an examination of file shows that we have data for each product produced by the household. But we want to know the total calories produced by the household, not the total calories from each separate product. After we convert all production to calories, we must sum the calories within each household to arrive at the household total.

With these points firmly in mind, let's begin by opening C-Q4.SAV.

1. Select **File/Open/Data...**
2. Select the file name c-q4.sav
3. Paste and run the command.

Let's first convert all production of the crops into kilograms. To find the conversion factor appropriate for each case in the production file (C-Q4.SAV), we need to look up the product and unit in the CONVER.SAV file. We will create a new file where each case has both the data from the production file and a variable containing the conversion factor for that product-unit combination. In SPSS for Windows, the command to do this is **Data/ Merge Files**/ **Add Variables**.

The input files for a merge must be sorted by the key variable(s) (those variables you are using to match the cases). Since we have a unique conversion factor for each product-unit combination, both our product variable and our unit variable are key variables. The CONVER.SAV file is already sorted by **prod** and **unit**. We must sort the currently working production file the same way, while taking account of the fact that the unit variable is named **p1a** and not **unit**. To sort the cases:

1. From the **Data** menu select **Sort Cases...**
   The Sort Cases *dialog box will come up.*
2. Select **prod** and click on ▶
3. Select **p1a** and click on ▶
4. Paste and run the command.

The files are now ready to be merged. **Merge Files** requires at least two files as input. In this case, the two files are working data file and CONVER.SAV. We are doing a "File - Table" merge where the second file is our "Lookup Table". The file created by **Merge Files** will become the working data file, replacing the current one.

1. From the **Data** menu select **Merge Files**, then select **Add Variables...**
   The Add Variables: Read File *dialog box will come up.*
2. Select the filename conver.sav
3. Click on **Open**

The variables used to match cases must have the same names. We must select **p1a** from the "New Working Data File" and move it into the box for excluded variables. We will rename it to **unit** and we will be able to use it as a variable to match the cases.

4. Select **p1a** from the list under New Working Data File: and click on ◀
5. Click on **Rename...**
   This will allow you to rename **p1a** *to* **unit** *to match the conversion file.*
6. Next to New Name: type **unit**
7. Click on **Continue**

We cannot select the variables to match by until we indicate we want to match cases on key variables.

8. Check the box next to Match cases on key variables in sorted files
9. Click on round button next to External file is keyed table
10. Select **prod** from the Excluded Variables: list
11. Click on ▶ next to Key Variables: (bottom, right)
12. Repeat steps 10 and 11 for **unit**
13. Paste the command
    A warning will come up telling you the data files must be sorted. Since we have sorted the files...

14. Click on **OK**
    A dialog box will ask you if you want to save the contents of the data *window. We do not want to save it, the new file can take its place, so...*
15. Click on **NO**
16. Select and run the command. Be sure to include `EXECUTE`.

The above steps tell SPSS for Windows to merge the working data file (active in your Data Editor window) and the CONVER.SAV file, (using CONVER.SAV as a table lookup) to add the unit variable to our working data file. Since the key variables need to have the same names in both files we renamed **p1a** (the unit variable for our working file) to **unit** (**p1a** will remain **p1a** in c-q4.sav).

Key Variables are required in any Merge when one of the files is being used as a keyed table. Our key variables specify doing the lookup by product and unit, because we have a different conversion factor for each product-unit combination. If we had used only **prod**, SPSS would expect each product to have only a single conversion factor, with the same value regardless of the unit of measurement used. For example, it would expect the same conversion factor for rice whether it was in a 100 kg bag or a 20 liter can. This would be incorrect.

The new working file produced by the merge now contains the needed conversion factor variable, **conver**. For every product-unit combination, **conver** is equal to the number of kilograms in that unit. It is always important to verify if the Merge was successfully completed. Return to the Data Editor window and look at some cases to verify that
the conversion factors match the products. For example, a 20-liter can when filled with maize grain actually has 18 kilograms of maize grain, thus check to see that when PROD=47 and UNIT=8, CONVER=18.

We can now calculate total kilograms by multiplying the number of units (**p1b**) by this conversion factor.

1. From the **Transform** menu select **Compute...**
2. Under Target Variable: enter **qprod_tt** (for total quantity of production in kg)
3. Click on **Type & Label** in order to add a label for **qprod_tt** here, if you wish, then select **Continue**.
4. From the list on the left of the Compute Variable window, select **p1b** and click on ▶ to put it in the right hand window, the numeric expression box.
5. Type * or select the button in the dialog box to add the multiplier sign next to **p1b**.
6. From the list on the left select **conver** and click on ▶
7. Paste, select and run the command

Next, we need to look up how many calories per kilogram each product contains. This information is in the table-lookup file CALORIES.SAV. This file has two variables—product and number of calories per kilogram. The key variable is product. In order to add the calorie-conversion variable to the working data file we need to do another merge with keyed table lookup. This time the key variable only needs to be the product variable. The data file has already been sorted by product (see the previous merge), so we don't need to sort it again.

1. From the **Data** menu select **Merge Files** then **Add Variables...**
2. Select the file calories.sav, **Open**
3. Check the Match cases... box
4. Check the External file is keyed table box
5. Put **prod** in the Key Variables: box
6. Paste the command
7. Clear the warnings as necessary
8. Select and run the command

The new working data file produced by the merge now contains the needed calorie variable, **calories**, but check to make sure. Maize grain (PROD=47) should have 3590 calories per kilogram in the **calories** variable. We can now compute total calories produced.

1. Use **Transform**/**Compute...**
2. Use **kprod_tt** as the Target Variable: (for total calories produced)
3. Click on **Type & Label** in order to add a label for **kprod_tt** here, if you wish, then select **Continue**.
4. Click in the Numeric Expression box and enter this equation **qprod_tt * calories**
5. Paste, select and run the command

This gives us a new working data file with total calories produced per product for each household. We are only interested in the seven staple food crops:
corn (prod=47), nhemba bean (prod=30), manteiga bean (prod=31), manioc (prod=41), rice (prod=6), sorghum (prod=44), and peanuts (prod=5)).

We can find these product coding by looking at **prod** in the questionnaire. Since we are only interested in those products, we can filter for just those cases. To make only these cases active we use **Select Cases**. **Select Cases** selects a subset of the cases based on particular criteria. **Select Cases** can either filter out the unselected cases or delete the unselected cases. If you delete the unselected cases you can return to the original file as long as you do not save the current working file under the same name as the original file. If you filter out the unselected cases (which we will be doing since it is a safer method) you can always unfilter the data which will activate all of the cases in the file.

1. From the Date Editor window, select **Data/Select Cases**
   You should see the Select Cases *dialog box.*
2. Select the round button next to If condition is satisfied
3. Click on **If...** under If condition is satisfied
4. Click **in** the box, to the right of ▶, **not** on the button itself .
5. Enter the following text (without hard returns):

```
PROD = 47 | PROD = 30 | PROD = 31 | PROD = 41 | PROD = 6 | PROD
= 44 | PROD = 5
```

   The "|" are symbols for the word OR. We are telling SPSS to select all cases with **prod** *equal to 47 or* **prod** *equal* 30 or **prod** *equal* 31...
6. Click on **Continue**
7. Select the round (radio) button next to Filtered (so that the cases with the other products are only temporarily not being used and may be brought into an analysis later.
8. **Paste** the command
9. Select the text (highlight it) in the Syntax Editor from `the line with USE ALL` to `the line with EXECUTE` and run the command.

Only cases with these product codes will now be used for analysis and saving the file to a new name. This subset of the data will be in effect until we open another file or use the **Data/Select Cases** to **Select All cases** (unfilter the cases).

Now, we need to know how many calories were produced per household for all staple food products combined. To do this, we need to sum, for each household, the values of **kprod_tt** for all of the food crops the household produced. In other words, we need to create a new household-level file from the current product-level file with one case per household. SPSS uses the term "AGGREGATE" to collapse the number of cases at one level to a new level. We will sum all the cases for household-product to one case for household.

To create the new household-level file, we use **Aggregate**. **Aggregate** will create a new data file with one case per household and **kprod_tt** summed across the products for each household. It always uses the

working data file as the file to be aggregated. We already have the production file open, so we are ready to aggregate.

1. From the **Data** menu select **Aggregate...**
   The Aggregate Data *window will appear.*
2. Select **district**, **vil**, and **hh,** respectively, for the Break Variable(s):
3. Select **kprod_tt** as the Aggregate Variable(s):
4. Click on **Name & Label...**
5. Change the default name **kprod__1** *to* **kprod_tt**
6. Enter the following label: Calories Produced in Staple Foods
7. Click on **Continue**
8. Click on **Function...**
9. Select Sum of values and click on **Continue**
10. Select Replace working data file, **OK**
11. Select Paste the command
12. Click on **No** to not save the contents of data window NewData
13. Run the command.

If we had selected Create new data file instead of Replace working data file, the new aggregated data file would have been stored on disk, and would not have become our working file. We would have had to open the data file to access it.

The Break Variable(s) specify the variables to be used for combining cases in the aggregated file. Any cases from the original file that have identical values for all of the break variables will be combined into a single case in the aggregated file. We want the aggregated file to have one case per household, so we use the variables that identify a household in our survey—**district**, **vil**, and **hh**.

Aggregate Variable(s) creates a new variable **kprod_tt**, which we calculate by summing **kprod_tt**, total calories produced, across all cases (the different food crops) for each household. The only variables which are contained in an aggregated file are the break variables and any new aggregated variables created (e.g. **kprod_tt**).

The new working data file now contains what we need, total number of calories from staple foods produced per household. To be sure this new variable exists, do a **Descriptives** on **kprod_tt**. You should find that the average number of calories produced per household per year is 4,483,964.7.

Save this data file using the **Save As...** command.

1. Make the Data Editor window active.
2. Use **Save As...** from the **File** menu
3. Name the file **hh-file1**
4. Paste and run the command.

Step 2: Generate a household level file containing the number of adult equivalents per household.

The data needed to calculate adult equivalents per household is in the member file, C-Q1A.SAV.

1. Click on the open folder button on the SPSS Data Editor Taskbar
2. Select the file name c-q1a.sav
3. Paste and run the command.

The rules we will use for calculating adult equivalents for this survey are:

Males, 10 years and older	= 1.0
Females, 10 to 19 years old	= 0.84
Females, 20 years and older	= 0.72
Children, under 10 years old	= 0.60

This says that, on average, a female 10 to 19 years old needs only 84% as many calories as a male 10 years or older, and that children under 10 need only 60% as many calories as the typical male older than 10. Thus for example, a child (male or female) under age 10 gets counted as .60 adult equivalents. For each person (case) in the member file we need to look at their sex, **ca4**, and their age, **ca3**, to calculate their adult equivalent.

Compute.../If... allows us to do this. The adult equivalent variable being created is **ae**.

1. From the **Transform** menu select **Compute...**
   The Compute Variable *window will appear.*
2. For the Target Variable: enter **ae**
3. Select the **Type & Label** box and enter Adult equivalent in the Label.Click on **Continue**
3. In the Numeric Expression: box enter a **1**
4. Click on **If...**
5. Select the radio button for Include if case satisfies condition:
6. Enter the statement **ca4 = 1 & ca3 >= 10**
7. Click on **Continue**
8. Paste the command but don't run it yet.
9. Repeat steps 1, and 3-8 replacing the previous information with the following. You are not obliged to use the menus within SPSS. Once you have a set of commands that you have pasted to the Syntax editor, it becomes much easier at this stage to simply copy and paste the same command within the Syntax editor itself and then changing the variables names. It is quicker. For those who cannot perform the copy/paste manoeuver here within the Syntax editor, simply repeat the steps above as indicated.

Numeric Expression	If... Statement
0.84	**ca4 = 2 & ca3 >= 10 & ca3 <= 19**
0.72	**ca4 = 2 & ca3 >= 20**
0.6	**ca3 < 10**

10. Select all of the `If` statements and run.

Your syntax should look like this:

```
IF (ca4 = 1 & ca3 >= 10) AE = 1 .
IF (ca4 = 2 & ca3 >= 10 & ca3 <= 19) AE = 0.84 .
IF (ca4 = 2 & ca3 >= 20) AE = 0.72 .
IF (ca3 < 10) AE = 0.60 .
VARIABLE LABELS AE 'ADULT EQUIVALENT' .
EXECUTE .
```

To verify that the new adult equivalent variable, **ae**, has been calculated, display a frequency table for it.

1. You will need to select **Analyze/Descriptive Statistics/Frequencies...**
2. Use **ae**
3. Paste and run

You should see there are 1524 total cases. Ideally there should be four values represented in the table —1, .72, .84, and .60— and no missing cases. You can see we have nine missing cases. This tells us that our data file is missing either the age or the sex for nine people. This is something that should have been identified during the cleaning process. At this point a researcher should go back to the original questionnaires and try to fix this. Since we can't do this, we will use an alternative method.

If we leave these values missing, the sizes of our households will appear to be slightly smaller than they actually are, which will distort our results. We could avoid this problem by eliminating the households of those nine individuals from our analysis, but then we can't use the information about the food production from those households. Instead, we will try to make a reasonable assumption about those nine missing members. We know that the adult-equivalent values range from a low of .6 for children to a high of 1.0 for adult males, which is not a very wide range. To find out the average adult-equivalent value for our sample...

1. **Analyze/Descriptive Statistics/Descriptives...**
2. Variable is **ae**
3. Don't forget to paste before you run the command

This shows that the mean value of **ae** for all individuals is .79, with a standard deviation of only .17. We will assume that the nine individuals with missing age or sex codes are all "average" individuals, and assign them the adult-equivalent value of .79. (Warning: be very cautious about "filling in" missing data this way, because careless use of this technique can give you misleading results. We are using this as an illustration of SPSS commands, not recommending that you do this routinely to compensate for missing data.)

1. **Transform/Recode/Into Same Variables...**
   Recode into Same Variables *dialog box will appear.*
2. Move **ae** to Variables:
3. Click on **Old and New Values...**
4. Select System-missing
5. Select Value: in the New Value section and enter **.79** in the box
6. Click on **Add**
7. **Continue**
8. Paste, select and run

Now we need to calculate the number of adult equivalents for each household. The current file is at the member level, but the values we need are for the household level. Again we use **Aggregate** to go from the member level to the household level. The new variable **ae_tt** will be calculated by summing **ae** across all members of a household.

1. From the **Data** menu select **Aggregate...**
2. Move **district**, **vil**, and **hh** to Break Variable(s):
3. Move **ae** to Aggregate Variable(s):
4. Click on **Name & Label...**
5. In the Name: box enter **ae_tt**
6. In the Label: box enter **Adult Equivalents**
7. **Continue**
8. **Function...**
9. Select Sum of values
10. **Continue**
11. Select Replace working data file
12. Paste, clear warnings and run.

Aggregate creates a new working file. The new working data file is at the household level, with one case per household. The variable **ae_tt** is the total adult equivalents for that household. To verify that this variable was created, do a **Descriptives** on **ae_tt**.

1. **Analyze/Descriptive Statistics/Descriptives...**
2. Paste and run.

You should find that the average adult equivalent over all households is 3.49.

This completes step 2. Save this file as HH-FILE2.SAV.

1. Make sure Data Editor window is active
2. **File/Save As...**
3. Filename hh-file2
4. Paste and run.

Step 3: We need to join the two files created in steps 1 & 2 together in order to compute calories produced per adult equivalent.

Now we have HH-FILE1.SAV containing the calorie-production data for all households, and we have HH-FILE2.SAV containing the adult-equivalent data for all households. We need to combine these files case-by-case to get both sets of data in a single file. To do this, we use **Merge Files**, but this time neither of the files are keyed tables.

We noted earlier that key variables are required for any merge that includes a keyed table lookup. When you're joining two files at the same level, as we're about to do, it may not seem important to include key variables, but it is. The key variables determine which cases are to be combined.

*You should never use **Merge Files** without Key Variables because without them you have no guarantee that SPSS will combine the right cases.* The command will execute without any warnings or error messages, but the results may be incorrect.

Note: hh-file2.sav is still the working file

1. **Data/Merge Files/Add Variables...**
2. Use file hh-file1.sav for the Read File
3. **Open**
4. Select Match cases on key variables...
5. Select Both files provide cases
6. Key Variables: are **district**, **vil**, and **hh** respectively
7. Paste, clear warning, select and run.

Merge Files created a new working data file. The two variables you need in order to compute calories produced per adult equivalent are now in the working file. Total calories produced (**kprod_tt**) per household for the year divided by total adult equivalents per household (**ae_tt**) divided by 365 days per year gives us calories produced per adult equivalent per day (**kprod_ae**).

1. **Transform/Compute...**
2. Target Variable: **kprod_ae**
3. **Type & Label...**
4. Label: **Calories produced per adult equivalent per day**
5. **Continue**
6. Numeric Expression: enter **kprod_tt/ae_tt/365**
7. Paste, select and run

Before we can produce the table we want, we have to create one more variable, denoting which calorie-production quartile each household falls in within their district. **Rank Cases** can do this for us. **Rank Cases** computes a new variable for each case, showing how that case ranks within a group according to the

value of another variable. In this case, we want to classify each household by how it ranks within its district in terms of calories produced per **ae**. Specifically, for each district, we want to break the households into four groups of equal size (quartiles), from lowest to highest calorie production. A new variable containing values from 1 to 4 will indicate to which quartile each household belongs.

1. **Transform/Rank Cases...**
2. Move **kprod_ae** to Variable(s):
3. Move **district** to By:
4. **Rank Types...**
5. Unselect Rank
6. Select Ntiles: 4
7. **Continue**
8. Paste and run
9. Note the new variable name in the Viewer; it should be NKPROD_A

The first thing we specify is the variable containing the values to use for the ranking—in this case **kprod_ae**. Then we need the By variable to specify the variable(s) that define the groups—in this case **district**. **Rank Cases** has a number of different methods of ranking. We're using one of the simplest—/NTILES(4) tells SPSS for Windows to break the variable into quartiles. From this command, SPSS for Windows will create a new variable that will contain the rankings and generate a name for for that variable.

We can now use **Means** to get the numbers to fill in our table.

1. **Analyze/Compare Means/Means...**
2. Move **kprod_ae** to Dependent List:
3. Move **nkprod_a** to Independent list: layer 1 of 1
   nkprod_a *came from the* **Rank Cases** *procedure.*
4. **Next**
5. Move **district** to Independent List: layer 2 of 2
6. Paste and run

You should note that mean for the entire population is 4014.5183 and the mean for the 2nd quartile in Ribaue is 2517.4551. The output from **Compare Means** gives you the numbers necessary for the table, although they are not formatted exactly as we showed the table at the beginning of this section. In Section 3 you will learn how to produce the same results but in a nicer-looking table format.

Save this file as HH-FILE3.SAV.

1. Make the Data Editor window active
2. **File/Save As...**
3. Filename is hh-file3
4. Paste and run

You should now save the contents of the Syntax Editor to a permanent command file for later use.

1. Make the Syntax Editor active
2. **File/Save As...**
3. Use the filename session2
   The .sps extension will be added automatically.

This file now contains all the commands from the Syntax Editor. *Whenever you do any substantial amount of work, you should always save the contents of the Syntax Editor to a command file.* You may have noticed that throughout the Sample Session we could have run the commands and by clicking on **OK** instead of **Paste**. Pasting commands into the Syntax Editor and then running them, rather than running them

directly, gives you documentation on your work and enables you to run the exact same analysis over again at a future date. Documenting now can save many steps later.

So now let's see how you would retrieve the command file you just created. To exit SPSS for Windows:

1. **File/Exit SPSS**
   SPSS will prompt you to save the contents of the windows that have not been saved; in this case the Viewer.
2. Save the Viewer as session2

Start SPSS for Windows again. To open our command file:

1. **File/Open/Syntax...**
2. Select the file session2.sps
3. **OK**
   The Syntax *window* c:\sample\session2.sps *will be active*

You can then re-execute these same commands or edit them as you wish.

Your SESSION2.SPS should look similar to this, with the exception that documentation comments have been added to this example, using an "*" at the beginning of each comment:

```
*session 2 - Produce table on food production in calories per
  adult equivalent per day.
*M. Beaver - 6/6/2002.

GET
  FILE='C:\sample\C-Q4.SAV'.

********Step 1 ********.
*preparing to merge conver file with this file - must sort by matching
  variables.

SORT CASES BY
  prod (A) p1a (A) .
MATCH FILES /FILE=*
 /RENAME  p1a=unit
 /TABLE='C:\sample\CONVER.SAV'
 /BY prod unit.
EXECUTE.

*calculating total quantity produced in kgs.

COMPUTE qprod_tt = conver * p1b .
VARIABLE LABELS qprod_tt 'COMPUTE  qprod_tt = conver * p1b (COMPUTE)' .
EXECUTE .

*merging in calorie conversion value.

MATCH FILES /FILE=*
 /TABLE='C:\sample\CALORIES.SAV'
 /BY prod.
EXECUTE.

*calculating total calories produced.

COMPUTE kprod_tt = qprod_tt * calories .
VARIABLE LABELS kprod_tt 'COMPUTE  kprod_tt = qprod_tt * calories (COMPUTE)'
  .
EXECUTE .
```

```
*setting filter to select only staple foods.

USE ALL.
COMPUTE filter_$=(prod = 47 | prod = 30 or prod = 31 or prod = 41 or prod = 6
  or prod = 44 or prod = 5).
VARIABLE LABEL filter_$ 'prod = 47 | prod = 30 or prod = 31 or prod = 41 or'+
 ' prod = 6 or prod = 44 or prod = 5 (FILTER)'.
VALUE LABELS filter_$  0 'Not Selected' 1 'Selected'.
FORMAT filter_$ (f1.0).
FILTER BY filter_$.
EXECUTE .

*check to be sure correct products are selected.

FREQUENCIES
  VARIABLES=prod
  /ORDER=  ANALYSIS .

*aggregating to the household level to sum total calories produced.

AGGREGATE
  /OUTFILE=*
  /BREAK=district vil hh
  /kprod_tt 'Calories produced in staple foods' = SUM(kprod_tt).

*verify variable is created and value is reasonable.

DESCRIPTIVES
  VARIABLES=kprod_tt
  /STATISTICS=MEAN STDDEV MIN MAX .

*save household level file.

SAVE OUTFILE='C:\sample\hh-file1.sav'
  /COMPRESSED.

**** Step2 - calculate adult equivalents ****.

GET
  FILE='C:\sample\C-Q1A.SAV'.

IF (ca4 = 1 & ca3 >= 10) ae = 1 .
VARIABLE LABELS ae 'Adult equivalent' .
IF (ca4 = 2 & ca3 >= 10 &ca3 <= 19) ae = 0.84 .
IF (ca4 = 2 & ca3 >= 20 ) ae = 0.72 .
IF ( ca3 < 10 ) ae = 0.6 .
EXECUTE .

*checking to see if compute is correct.
list ca4 ca3 ae / cases=20.
freq ae.

*get the mean for the total population.

DESCRIPTIVES
  VARIABLES=ae
  /STATISTICS=MEAN STDDEV MIN MAX .

*replace sysmis with the mean for the total population.
```

```
RECODE
  ae  (SYSMIS=.79)  .
EXECUTE .
freq ae.

*aggregating to the household level summing adult equivalents.

AGGREGATE
  /OUTFILE=*
  /BREAK=district vil hh
  /ae_tt 'Adult Equivalents' = SUM(ae).
DESCRIPTIVES
  VARIABLES=ae_tt
  /STATISTICS=MEAN STDDEV MIN MAX .

SAVE OUTFILE='C:\sample\hh-file2.sav'
  /COMPRESSED.

**** Step 3 - join hh calorie and hh ae together ****.

MATCH FILES /FILE=*
 /FILE='C:\sample\hh-file1.sav'
 /BY district vil hh.
EXECUTE.

* calculate calories produced per adult equivalent per day.

COMPUTE kprod_ae = kprod_tt / ae_tt / 365 .
VARIABLE LABELS kprod_ae 'Calories produced per adult equivalent per day' .
EXECUTE .

RANK
  VARIABLES=kprod_ae  (A) BY district  /NTILES (4) /PRINT=YES
  /TIES=MEAN .
MEANS
  TABLES=kprod_ae  BY nkprod_a  BY district
  /CELLS MEAN COUNT STDDEV  .

SAVE OUTFILE='C:\sample\hh-file3.sav'
  /COMPRESSED.
```

Exercise 2.1: Produce similar output using calories retained (production minus sales) instead of calories produced. It will show calories retained per adult equivalent per day from the total of the same six food crops. The output should be broken down by district and calorie production quartile.

Hints:
a. The procedure is very similar to the work that we just completed.
b. Sales come from c-q5.sav.
c. Check the file for the appropriate variable for the quantity of sold production. Note that the product codes are the same as for c-q4.sav. Also check for the variables by which to sort.
d. Retrieve the commands from generating the previous table and check each step for needed changes. There will be changes of product code, file names, and variables.
e. Computing the calories sold involves the same basic steps as computing the calories produced. (Step 1)
f. Merge this newly created file, (the file containing calories sold), with the file containing calories produced, hh-file3.sav.

g. Keep in mind that only 256 households sold products, but all 343 households produced and retained calories. If the calories-sold variable is missing, it means the household did not sell food, so it should be recoded to zero.
h. Compute calories retained = calories produced - calories sold.
i. Rank into quartiles.
j. Use the **Compare Means** command to show calories retained by **district** and **quartile**.
k. Save the data file.
l. There's no need to save the contents of the Syntax Editor, from the exercise, to a file.
m. Execute the newly created syntax file, select all and run

This is an example of the output you should produce:

Report

KRET_AE

NKRET_AE NTILES of KRET_AE by DISTRICT	DISTRICT DISTRICT	Mean	N	Std. Deviation
1	1 MONAPO	1148.045	27	409.61445
	2 RIBAUE	1232.803	29	350.22596
	3 ANGOCHE	912.7559	28	384.74681
	Total	1098.877	84	401.03778
2	1 MONAPO	2211.383	27	205.71992
	2 RIBAUE	2145.845	30	202.81580
	3 ANGOCHE	1698.510	29	168.49973
	Total	2015.575	86	297.99128
3	1 MONAPO	3314.857	28	477.12339
	2 RIBAUE	3126.358	30	329.89358
	3 ANGOCHE	2405.008	29	336.48560
	Total	2946.574	87	547.14537
4	1 MONAPO	7619.102	27	3557.135
	2 RIBAUE	5759.039	30	1649.584
	3 ANGOCHE	4954.763	29	2426.824
	Total	6071.803	86	2821.271
Total	1 MONAPO	3570.975	109	3032.696
	2 RIBAUE	3081.416	119	1902.739
	3 ANGOCHE	2506.498	115	1957.991
	Total	3044.234	343	2370.146

Tables

Using **Tables** you can calculate various statistics and present them in a variety of ways that are completely under your control. Unlike other SPSS for Windows procedures, **Tables** allows you to do the following:

- to choose how you want to assemble variables and statistics for display in rows, columns, and layers. (The variables can be stacked or nested. Stacked means that the more than one variable can be displayed in the rows below one another or in columns next to each other. Nested means that all of the values for one variable are displayed below the individual values of another variable.)
- to manipulate table structure, content, and presentation format.
- to include flexible percentages, specifying the base for the percentages (their denominator) so that they add to 100% across rows, columns, subtables, or whole tables.
- display up to 60 characters for variable labels and value labels.

There are 4 types of tables under the Custom Tables command:

Basic tables - all tools available under tables are applied uniformly to all variables
General tables - nesting, stacking, statistics, or totals can be applied differently to different variables
Multiple response tables - variables where respondent can give more than one response to a question
Table of Frequencies - special purpose table - frequency of categorical variables that have the same categories, e.g. Most important ... Least important

Let's compare the **Crosstabs** procedure with the **Tables** procedure for crosstabulation.

Open the member file we created that contains the age variable, Q1A-AGE.SAV.

1. **File/Open/Data...**
2. Select q1a-age.sav
3. Paste, select and run.

First do a simple crosstabulation using the **Crosstabs**.

1. **Analyze/Descriptive Statistics/Crosstabs...**
2. Move **ca2** to Row(s):
3. Move **age_gp** to Column(s):
4. **Cells...**
5. Select Observed in the Counts section
6. Select Row in the Percentages
7. **Continue**
8. Paste and run.

Below is the output.

RELATION TO HEAD * age group Crosstabulation

			age group				
			0 to 10	11 to 19	20 to 60	61 and older	Total
RELATION TO HEAD	HEAD	Count		6	296	41	343
		% within RELATIO TO HEAD		1.7%	86.3%	12.0%	100.0%
	WIFE/HUSBAND	Count		25	280	5	310
		% within RELATIO TO HEAD		8.1%	90.3%	1.6%	100.0%
	SON/DAUGHER	Count	503	184	31		718
		% within RELATIO TO HEAD	70.1%	25.6%	4.3%		100.0%
	MOTHER/FATHE	Count			5	1	6
		% within RELATIO TO HEAD			83.3%	16.7%	100.0%
	OTHER RELATIV	Count	70	55	16	2	143
		% within RELATIO TO HEAD	49.0%	38.5%	11.2%	1.4%	100.0%
Total		Count	573	270	628	49	1520
		% within RELATIO TO HEAD	37.7%	17.8%	41.3%	3.2%	100.0%

Let's use **Basic Tables**.to produce the same table:

1. **Analyze/Custom Tables/Basic Tables...**
2. Move **ca2** to Down:
3. Move **age_gp** to Across:
4. Paste and run.

		age group			
		0 to 10	11 to 19	20 to 60	61 and older
RELATION TO HEAD	HEAD		6	296	41
	WIFE/HUSBAND		25	280	5
	SON/DAUGHER	503	184	31	
	MOTHER/FATHER			5	1
	OTHER RELATIVE	70	55	16	2

This is a **Basic Table**, using its default settings. The row labels correspond to the value labels for variable **ca2** (relation to head). The column labels are the value labels which you designated for the variable **age_gp**. If you want to customize the table for your needs, **Basic Tables** can become much more complex. Let's create a more complex table...

Modifying label

1. Select **Variable View** in the Data Editor
2. In the **Label** column, click on the cell for **ca2**.
3. Delete the text

This will blank out the label for **ca2** *so it will not appear in the first column.*

Creating table

1. **Analyze//Custom Tables/General Tables...**
2. Move **ca2** to Rows:
3. Move **age_gp** to Columns:
4. Select **age_gp**, click on **Edit Statistics...**

Modifying statistics

1. Select **Count** under Cell Statistics:
2. Change Label: to **N**
3. Check to see that the Width: = **5**
4. Click **Change**
5. Select Row % from under Statistics:
6. Change the Label: to **%**
7. Check to see that the Width: = **5**
8. **Add**
9. **Continue**

Adding Total

1. **Insert Total**
2. Select age_gpTotal, click on **Edit Statistics...**
3. Select the round button next to Custom Total Statistics
4. Select **Count** under Statistics:
5. Change Label: to **N**
6. Change Width: to **5**
7. **Add**
8. **Continue**
9. Select **ca2**, click on **Insert Total**

Modifying format

1. **Format...**
2. Set Empty Cell Appearance to Zero
3. **Continue**

Adding title

1. **Titles...**
2. In the Title box:
3. Type **Table 1: SPSS for Windows Sample Session**
4. Type **Age Breakdown by Relation to Head**
5. In the Caption Box type **Source: Nampula family sector household survey, 1991**.
6. In the Corner box: type **Relation to Head**
7. **Continue**

Paste the command and run it.

Table 1: SPSS for Windows Sample Session
Age-Breakdown by Relation to Head

	age group								Total
	0 to 10		11 to 19		20 to 60		61 and older		
Relation to Head	N	%	N	%	N	%	N	%	N
HEAD	0	.0%	6	1.7%	296	86.3%	41	12.0%	343
WIFE/HUSBAND	0	.0%	25	8.1%	280	90.3%	5	1.6%	310
SON/DAUGHER	503	70.1%	184	25.6%	31	4.3%	0	.0%	718
MOTHER/FATH	0	.0%	0	.0%	5	83.3%	1	16.7%	6
OTHER RELAT	70	49.0%	55	38.5%	16	11.2%	2	1.4%	143
Total	573	37.7%	270	17.8%	628	41.3%	49	3.2%	1520

Source: Nampula family sector household survey, 1991

This is the table produced from those commands. Note: You may change the table properties, formats, pivot the tables, etc.

The corner label does not show automatically. To get the label to show, select the table by double-clicking on it in the Viewer. Right-click on the same table and select **Table Properties...** in the pop-up menu. Select the ***Nested*** Radio button in the *Row Dimension Labels* box. Click on OK. And then click outside the table to resume your work in the SPSS Viewer. You should now see the title in the corner box.

Save this output file from the Viewer using the **Save As...** command.

1. Make the Viewer window active.
2. Use **Save As...** from the **File** menu
3. Name the file **session3.spo** and save.

This may seem like a lot of effort to produce a single table. For a one-time application it may seem easier to create the table in a word processor. However, if this is something you are going to do repeatedly, the benefits of having SPSS for Windows produce the "pretty" table far outweigh the effort to create the table. This becomes important with periodic data, such as monthly prices, where each month the table should be updated.

The following is a comparison of computing averages using **Compare Means** and using **Custom Tables**, based on an example from section 2.

1. **File/Open/Data...**
2. Select hh-file3.sav
3. Paste, select and run
4. **Analyze/Compare Means/Means...**
5. Move **kprod_ae** to Dependent List:
6. Move **nkprod_a** to Independent list: layer 1 of 1
7. **Next**
8. Move **district** to Independent List: layer 2 of 2
9. Paste and run

It produces the following output:

Report

Calories produced per adult equivalent

NTILES of KPROD_AE	DISTRICT	Mean	N	Std. Deviation
1	MONAPO	1221.7281	27	416.1286
	RIBAUE	1484.0298	29	422.1161
	ANGOCHE	1272.0519	28	486.2593
	Total	1329.0592	84	452.2224
2	MONAPO	2494.8048	27	377.1214
	RIBAUE	2517.4551	30	366.0805
	ANGOCHE	2431.9673	29	296.8005
	Total	2481.5167	86	345.8224
3	MONAPO	3968.1419	28	621.3403
	RIBAUE	4000.8905	30	549.8340
	ANGOCHE	3640.3535	29	453.2870
	Total	3870.1717	87	562.9770
4	MONAPO	9150.0222	27	4686.2114
	RIBAUE	7520.2527	30	2158.8635
	ANGOCHE	8364.3191	29	4054.9027
	Total	8316.5516	86	3764.1698
Total	MONAPO	4206.4675	109	3813.5641
	RIBAUE	3900.7967	119	2559.3106
	ANGOCHE	3950.2610	115	3390.5114
	Total	4014.5183	343	3271.4011

This is the information we needed to fill in the numbers of our table in section 2. Let's use **Custom Tables** to produce output that looks similar to the table we were shooting for throughout section 2. Let's add the Minimum and Maximum to the table for more information.

1. Change the labels for the following Variables by going to the Vraibale Views window in the Data Editor. In the column for Label, double click on the box for the given variables below and change the text in the box according to the indications below:

Variable	**New Label**
kprod_ae	delete existing label and leave blank
district	type in District
nkprod_a	delete existing label and leave blank

2. **Analyze/Custom Tables/Basic Tables...**
3. Move **kprod_ae** to Summaries:
4. Move **nkprod_a** to Down:
5. Move **district** to Across:
6. **Statistics...**
7. Select Mean, use the label **MEAN**, Format: **ddd.dd**, Width: **5** and Decimals: **0**
8. **Add**
9. Select Maximum, use **MAX,** Format: **ddd.dd**, Width: **5** and Decimals: **0**
10. **Add**
11. Select Minimum, use the label **MIN**, Format: **ddd.dd**, Width: **5** and Decimals: **0**

12. **Add**
13. **Continue**
14. **Layout...**
15. In the Statistic Labels section, select, Down the Left side, **Continue**
21. **Titles...**
22. Type in the Title box: **Table 1: Food Production in Calories**
23. Press <Enter>, then type **per Adult Equivalent per Day**
24. Enter in the Corner box: **Production Quartile**.
25. **Continue**, paste and run

This produces the following table:

Table 1: Food Production in Calories
Per Adult Equivalent per Day

Production Quartile		District		
		MONAPO	RIBAUE	ANGOCHE
1.00	Mean	1222	1484	1272
	Max	1956	1938	1952
	Min	294	429	354
2.00	Mean	2495	2517	2432
	Max	3169	3120	2961
	Min	1973	2030	2024
3.00	Mean	3968	4001	3640
	Max	5067	4834	4563
	Min	3176	3141	2996
4.00	Mean	9150	7520	8364
	Max	28466	13124	20485
	Min	5107	4984	4692

Some of you might not have the **Production Quartile** showing in the Corner box. You need to have the Corner box "Nested" which you can do through the *Pivot Table Editor*. Select the table by double-clicking on it in the Viewer. Right-click on the same table and select **Table Properties...** in the pop-up menu. Select the ***Nested*** Radio button in the *Row Dimension Labels* box. Click on OK. And then click outside the table to resume your work in the SPSS Viewer. You should now see the title in the corner box.

Summaries specifies which variables are continuous. This is usually the most important variable, one for which the statistics will be computed. In the above example the variable **kprod_ae** is a continuous variable for which minimums, means, and maximums are calculated for each category of **nkprod_a.**

Subgroups determines how to group the cases into rows and columns of the table. Variables named in **Subgroups** should always be category variables (variables that have a finite number of values).
Statistics indicates what statistics should be computed using the continuous (Summaries) variables.

If SPSS for Windows reports an error for a **Custom Table** it usually has to do with using an incorrect variable for the requested function or variable width being too large for the table. If the table is obviously wrong from the first screen, use <F3> to stop further processing. If you get one of these errors check to be sure that you listed a continuous variable under Summaries and that variable widths and column widths are adequate.

A simple way to print a table you have just created, is to select the table(s) in the **Viewer** and print.

1. Make the Viewer active
2. Select the table you wish to print
3. **File/Print.../**. The **Selection** button should be chosen. Then select **OK**.

Exercise 3.1: Produce a similarly formatted table using calories retained as you did in Exercise 2.1. **Include totals** by production quartile (you will have to use **Basic Tables**). Your table should look similar to this:

Table 1: Food retention in calories
Per adult equivalent per day

Production Quartile		District			Tota
		MONAPO	RIBAUE	ANGOCHE	
1					
	MEAN	1148	1233	913	109
	MAX	1806	1783	1391	180
	MIN	224	429	208	20
2					
	MEAN	2211	2146	1699	201
	MAX	2544	2556	1936	255
	MIN	1807	1790	1396	139
3					
	MEAN	3315	3126	2405	294
	MAX	4303	3730	3055	430
	MIN	2555	2566	1984	198
4					
	MEAN	7619	5759	4955	607
	MAX	20874	9465	12675	2087
	MIN	4360	3731	3064	306

Multiple Response Question

The information sought by analysts often requires questions that allow the respondent to select multiple answers. A single SPSS for Windows variable cannot record the answers to this type of question adequately, because a variable can have only one value for each case. The solution is to record each possible response in a different variable. The responses can be analyzed separately using commands you have already seen (**Frequencies**, **Crosstabs**), but ideally we want to analyze these related variables jointly. This can be done with a special grouping feature of **Custom Tables/General Tables**. SPSS for Windows allows two different grouping methods, to handle two different ways of asking this type of question.

If a survey question asks the respondent to "check all that apply" from a set of ten choices, ten variables must be used to code the responses. You would need to create a variable for each of the ten possible responses. Each variable would have a value to indicate whether the response was checked (1), or not checked (0). These variables are called *multiple dichotomy* variables. Refer to the Tables manual for more detail.

On the other hand, if the survey question asks the respondent to "list up to 4 choices" from a set of ten, four variables must be used to code the responses. The set of possible responses would be numbered from 1 to 10, and the user would enter up to four of those numbers. These values would be entered for the four variables in SPSS for Windows. These variables are called *multiple response* variables. Again refer to the Tables manual for more detail.

Question 35 of the household questionnaire is an example of a multiple response question. It asks about crops grown principally to be sold. Each household is asked to specify up to three main crops which are coded into variables **h35a**, **h35b**, and **h35c**. The crop codes are provided for five of the most common crops. The question is left open-ended, however, since a code of 6 is allowed for other crops, which are written down by name.

Because the question is open ended, more categories were added to these variables. This is something that is done after all surveys are completed. Someone manually goes through all surveys and assigns a code to each of the different crops respondents came up with for "6-other". Codes and value labels are then assigned for each crop. As you will see with the following commands, eleven different crops ended up being coded for question 35.

The best way to analyze this question is to use **Custom Tables** with Multiple Response Sets. You could use **Frequencies** on each of the variables individually, but you would then have to sum the results by hand. **Custom Tables** will calculate these summary statistics for you if you create a group variable using the Multiple Response Sets. Open the household data file.

1. **File/Open/Data...**
2. Select c-hh.sav
3. Paste, select and run

To create the table do the following:

1. **Analyze/Custom Tables/General Tables...**
2. **Mult Response Sets...**
3. Select **h35a**, **h35b**, **h35c** and move to Variables in Set:
4. Select Categories in the Variables Are coded As box
5. Name: **crops**
6. Label: **Crops grown principally to be sold**
7. **Add**, **Save**
8. move **$crops**, in Mult Response: to Rows:
9. Move **district** to Columns:
10. Select **$crops**
11. **Insert Total**
12. Select **$cropsTotal**
13. Change label to **Total Cases** in Total Label:
14. Paste and run.

The Syntax editor should show this:

```
TABLES
 /FORMAT BLANK MISSING('.')  /MRGROUP  $crops 'Crops grown principally to be' +
' sold'  h35a h35b h35c
  /GBASE=CASES
  /FTOTAL= $t000001 "Total Cases"
  /TABLE=$crops + $t000001  BY district > (STATISTICS)  .
```

Mult Response creates a group variable **$crops** from the three variables **h35a**, **h35b**, and **h35c**. Insert Total produces the summary statistics. As discussed in the previous section, the format for the table can be customized using the options in the Custom Table: General dialog box..

The output table should look like this:

		DISTRICT		
		MONAPO	RIBAUE	ANGOCHE
Crops grown principally to be sold	COTTON	63	24	3
	PEANUTS	13	2	70
	SESAME			3
	SUNFLOWER		1	
	RICE	5	2	78
	MAIZE, BEANS	7	18	16
	BANANA		2	2
	MANIOC		2	5
	SUGAR CANE	3	1	
	TOBACCO		1	
	SWEET POTATO			1
	CASHEW NUT	1		
Total Cases		75	44	90

Save this output file with all the tables and output in it using the **Save As...** command.

1. Make the Viewer window active.
2. Use **Save** from the **File** menu to automatically save under the name Session 3.spo.

SPSS for Windows SAMPLE SESSION
SECTION 4 - Graphs, tables, publications and presentations, how to bring them into word processor

The objective of this module is to give you the tools necessary to prepare reports, i.e. to learn how to move SPSS results into other applications. While it is possible to move SPSS text output, tables, charts and other graphics, into word processors and spreadsheets under Windows and DOS, this module will uniquely focus on a chart or table as an example. The methods used in this example would be quite similar for other SPSS results.

This module will not look at how to prepare presentations or publications. An additional module should be developed to look at these questions more precisely and should include such concepts as titles, text boxes, image insertion and watermarks among other basic concepts. Nonetheless, the tools presented in this module, showing how to transfer SPSS results to a word processor, is paramount and key to diffusing and disseminating reports and results in a presentable manner.

The method is simple: once the SPSS results such as a chart or a table are produced (it is always better to save the output as well), it can be printed or incorporated into reports prepared using word processors or publishing programs. Incorporating tables and charts from SPSS can be done using a simple copy and paste procedure (the **Copy** command must be made from an output file in the Viewer). Proceed with a simple copy and paste procedure of an SPSS table of from the output file Sess3.spo. Find the following table in that file:

1. Go to **File/Open/Output...** in SPSS 10.0 for Windows
2. Select Session3.spo in the sample folder where you saved your output from the sample session (*.spo extension)
3. Click on Open
4. Locate the table indicated in the Viewer. Click once on it with left hand side mouse to select it.
5. Select **Edit/Copy** through the menu system (or click the "right" mouse button and select "Copy").
6. Now open your word processor software if it is not already open.

The following instructions are for WordPerfect 8.0 and Word 97.

		DISTRICT		
		1 MONAPO	2 RIBAUE	3 ANGOCHE
Crops grown principally to be sold	1 COTTON	63	24	3
	2 PEANUTS	13	2	70
	3 SESAME			3
	4 SUNFLOWER		1	
	5 RICE	5	2	78
	6 MAIZE, BEANS	7	18	16
	7 BANANA		2	2
	8 MANIOC		2	5
	9 SUGAR CANE	3	1	
	10 TOBACCO		1	
	11 SWEET POTATO			1
	12 CASHEW NUT	1		
Total Cases		75	44	90

With your word processor open, Select **Edit**

A) If you click on the **"Paste"** button, the table is pasted in "RTF" (rich text format). It can then be edited as any other table. (If you have problems, try using the "picture" paste below.)

B) If you choose to use the "**Paste Special**" from the "Edit" menu, you are given three choices,
1) the RTF format - same as using "Paste"
2) unformatted text - pastes the last text that you copied
3) picture - pasted as a graphic image

The preferable choice is picture. The picture cannot be edited in the word processor, but it usually retains the exact look of the SPSS table. If you wish to change the table, you must return to the SPSS Viewer, make the changes, then copy it from within Viewer and return to your word processor to paste it as a picture again.

If you have copied a graphic, then you should use the "Paste Special" and select "bitmap" if you are using WordPerfect. Either bitmap or Enhanced metafile works for Word.

The "Copy Object" choice from within the Viewer works well for Word, but not at all for WordPerfect

Is there is something you do not like on the table? Would you like to add a title? You should all modifications to the graphs or tables from within SPSS, NOT in the word processor (editing a picture within the word processor invokes the graphic editor for that word processor and may very difficult to manipulate).

You can also copy tables and paste them into spreadsheet programs if you are more comfortable making formatting changes there. There are many formatting aspects that can be controlled from within SPSS. Check the Tables Manual for instructions on using Pivot Table and all the other ways to change the structure and look of a table.

Unfortunately, if the original chart or table file changes (say you were preparing a paper with a graph for real prices and you need to add another month or change data), to update your report you must erase the old tables and paste new copies.

The process is basically the same for Graphics, such as pie charts and histograms, but there is more flexibility in the ways to save the file, along with more difficulties in getting just the look you want. As an example, we will look at the distribution of cashew tree ownership across households in the Mozambique data, using a histogram.

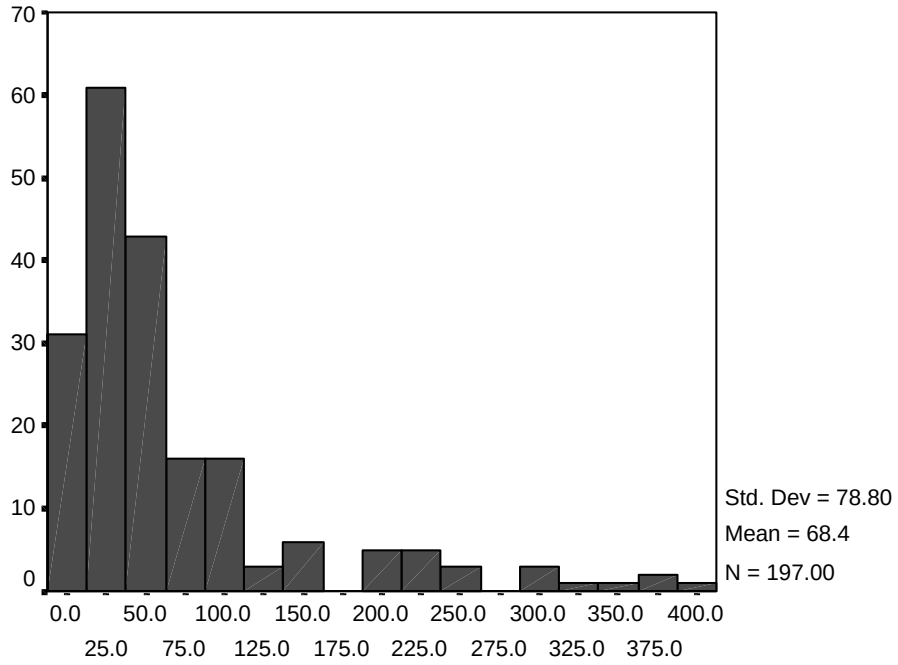
Open the member file we created that contains the tree ownership variable, C-HH.SAV.

1. **File/Open/Data...**
2. Select C-HH.sav
3. Paste, select and run.

Create the Histogram chart using the variable H57 (number of trees owned):

4. Select **Graphs/Histogram....**
5. Find H57 (Number of cashew trees) in the variable list and move it into the Variables box.
6. Paste, select and run.

You should get a histogram chart like this:



Next, you will copy the Chart in order to put it into your word processor. We have found that there are differences between word processors. The following example is given for Word Perfect.

7. Go to the Viewer and click once on the Graph to select it. If you want to edit it, adding Titles or changing fonts or other actions, click twice for the editing facility. When you are done close the editing window by clicking elsewhere in the output file. Them click once on the graph to select it.
8. Use CTRL-C (pressing CTRL and C at the same time) to copy the graph into memory.
9. Open Word Perfect and place your cursor where you would like the top left-hand corner of the graph to appear.
10. Use CTRL-V (pressing CTRL and V at the same time) to paste the graph into Word Perfect.

You will not be able to edit this graph, other than the size, placement, wrapping of text and other basic aspects of the fit into Word Perfect.

You can save a chart as a separate file by double-clicking on the Chart in the Viewer. Then select **File/Export Chart** , selecting BMP as the file type. You can then use **Insert/file** in Word Perfect in order to bring the BMP graphic into a file or **File/Open**. Afterwards, you can change the size but nothing else within the file.

For MicroSoft Word, we have found that it works best to copy the graph using **Edit/Copy object** in the SPSS Viewer and then in MS Word, using **Edit/Paste special/Picture** to paste the chart in. See the SPSS Base 10.0 Users Guide pp.144-155 for more options.

Exercise 4.1.

Select another table from your Sess3.SPO file. Repeat steps 1 to 9 to create a Word Perfect or Word document with the table in it. Practice making various changes to the table in SPSS and copy the output to your word processor document.

Annexes

The following annexes were prepared for users of the sample session to have a brief reference guide, to explain the various functions of the SPSS commands most commonly used in the sample session, to describe the numerous options available to the user within the various menus and finally, to help manipulate results in the Output navigator.

ANNEX 1

Filters Versus Permanent Selections.

You can filter or delete cases that don't meet the selection criteria. In Section 2 of the cross-sectional training, we filtered the data but we did not delete any cases. When you set a filter from the **Data/Select cases** command, unselected cases are filtered by default. Filtered cases remain in the data file but are excluded from analysis. You can see which cases are filtered out by looking at the far left column of the Dat View window, where the case numbers are given. Numbers with a slash through them have been filtered and will not be included in an analysis or reporting. SPSS creates a filter variable, FILTER_$, to indicate filter status. Selected cases have a value of 1; filtered cases have a value of 0. To turn filtering off and include all cases in your analysis, select All cases in the **Data/Select cases** command. If you want to delete specific cases from the data set, use the **Data/Select cases** command, complete an IF statement for those cases that you want to keep, and then select the choice **deleted** in the **Unselected Cases Are** box. Be sure to save this file under a new name or you will permanently delete the cases from the data file.

The Three Line Charts and Three Data in Charts Options.

The **Graph/Line** command allows you to make selections that determine the type of chart you obtain, simple, multiple and drop-line. In the menu, select the icon for the chart type you want, and select the option under Data in Chart Are that best describes your data. You can see a description of the three available Data in Chart types below. A category axis on a chart is an axis that displays values individually, without necessarily arranging them to scale. (A scale axis, in contrast, displays numerical values to scale.) Bar charts, line charts, and area charts usually have one category axis and at least one scale axis. Scatterplots and histograms do not have a category axis.

The Missing Values options are available only when the new chart will display or summarize more than one variable (not including variables that define groups):

- Exclude cases listwise excludes a case from the entire chart if is has a missing value for any of the variables summarized.

- Exclude cases variable by variable excludes a case separately from each summary statistic calculated. Different chart elements may be based on different groups of cases.

Display groups defined by missing values is available only when you use a categorical variable to define groups for a new chart. If selected, each missing value for the grouping variable (including the system-missing value) will appear as a separate group in the chart. If not, cases with system-missing or user-missing values for the grouping variable are excluded from the chart. It is recommended to always uncheck this box as it is not of interest to show on a graph the missing values or sysmisses.

Simple lines

Summaries for Groups of Cases
Categories of a single variable are summarized. The y-height of the points is determined by the Line Represents option.
A single Category Axis variable.

Summaries of Separate Variables
Two or more variables are summarized. Each point represents one of the variables.
Two or more Line Represents variables.

Values of Individual Cases
A single variable is summarized. Each point represents an individual case.

A single Line Represents variable.

Multiple lines

Summaries for Groups of Cases
Categories of one variable are summarized within categories of another variable. The y-height of the points is determined by the Lines Represent option.
A Category Axis variable (Category Variable 1).
A Define Lines by variable (Category Variable 2).

Summaries of Separate Variables
Two or more variables are summarized within categories of another variable.
Two or more Lines Represent variables (Var 1, Var 2).
A Category Axis variable (Category Variable).

Values of Individual Cases
Two or more variables are summarized for each case.
Two or more Lines Represent variables (Var 1, Var 2).

Manipulating Output in SPSS 10.0 for Windows.

Numerous modules could be dedicated to working with the Output navigator. Section 4 only discussed simple cutting and pasting of results. One suggestion would be to follow the tutorial within SPSS to learn about the countless possibilities and options which are available to the SPSS user in the Output navigator. Your results have never looked this good! Easier and faster data exploration and to ability to drag icons in the navigator outline and content panes on the left, expand and collapse the outline - see the output you want; multi-dimensional pivot tables, swapping and hiding rows and columns, new and numerous styles for charts and tables, colors, fonts, line styles, text attributes; no loss of any custom formatting, dragging output from SPSS to a word processor (in windows metafile format); change a title directly within the output, right click for pop-up menus as shortcuts, and much more.

You may have trouble viewing the complete output following a SPSS command like **Frequencies** or **Tables**. It may run hundred and thousands of cases but will only show the first 50 for example. To view all of the specific output in this case, simply double click or right click on the selected output and choose Open. This will open a separate window called a pivot table. Then scroll down to see the output in whole. You may also edit the table here as well. Enjoy using the various options given to you to modify the styles, formats, colors, text attributes and so on.

ANNEX 2

Socio-Economic Survey of Family Sector Farms
in the Province of Nampula
(Angoche, Monapo e Ribaúe)
July/August 1991
Departamento de Preços e Mercados
Food Security Project

Name of Household Head _____

Household Number _____ HH

Aldeia _____ VIL

Distrito _____ DIST

(Subset of questions from original questionnaire)

I.**HOUSEHOLD CHARACTERÍSTICS**

H1 _____ 1. How many persons are in this household?

H4 _____ 4. Has your family always lived in this village?
1=yes 2=no

H8 _____ 8. Is your family registered as "deslocada"?
1=yes 2=no

H19 _____ 19. Do you presently have lands in fallow?
1=yes 2=no

H21 _____ 21. What is the total area of these fallowed parcels? (hectares)

H24 _____ 24. Do you have lands that you have completely abandoned?
1=yes --> question 25 2=no --> question 27

H25 _____ 25. What is the total area of these abandoned lands? (hectares)

H26 _____ 26. What was the principal motive for abandoning these lands?
1=no security
2=lands lost fertility
3=lack of labor
4=insect attacks
5=other

[**We would like to ask you about the food crops you grow**.]

H29 _____ 29. Over the last five years, have you increased or decreased the amount of land in food crops?
1=increased 2=decreased 3=no change

H31 _____ 31. During a normal year, is your farm production sufficient to feed your entire family?
1=yes 2=no

[**We would like to ask you about the cash crops you grow on your farm?**]

H34 _____ 34. Do your grow any crops that are principally destined for the market?
1=yes 2=no

35. Which crops are grow principally to be sold? (List the three most important)

H35A ___ 1=cotton 4=sunflower
H35B ___ 2=peanuts 5=rice
H35C ___ 3=sesame 6=other

H36 _____ 36. Over the last five years, have you changed the area grown in these cash crops?
1=increased
2=decreased
3=no change

H39 _____ 39. Do you normally grow cotton?
1=yes 2=no

H52 _____ 52. Since your involvement with the cotton companies, have you reduced your area dedicated to food crops, such as maize and manioc?
1=yes 2=no

IV. **PRODUCTION**

H56 _____ 56. Do you have cashew trees?
1=yes 2=no

H57 _____ 57. How many trees do you presently have? (number)

H57A ___ 57A. Of these trees, from how many did you harvest during the last year? (number)

V. AGRICULTURAL SALES

We would like to ask about the marketing of your agricultural products since August of 1990.
64. Over the last five years, have you increased the quantities marketed of the following crops:

H64A ___ a. maize 1=yes 2=no
H64B ___ b. manioc 1=yes 2=no
H64C ___ c. rice 1=yes 2=no
H64D ___ d. cotton 1=yes 2=no
H64E ___ e. peanuts 1=yes 2=no
H64F ___ f. beans 1=yes 2=nao
H64G ___ g. sorghum 1=yes 2=nao
H64H ___ h. cashew nuts 1=yes 2=no

H65 _____ 65. Compared with five years ago, has the marketing of these products been more difficult or easier?
1=more difficult --> question 66
2=easier --> question 67

H66 _____ 66. If more difficult, why?
1=fewer buyers
2=transportation problems
3=security problems
4=low prices
5=lack of consumer goods
6=other _____

H67 _____ 67. If easier, why?
1=more buyers
2=better transportation
3=better security
4=attractive prices
5=more consumer goods
6=other_____

H83 _____ 83. Does your family usually receive traditional gifts or participate in exchange relations?
1=yes 2=no

H84 _____ 84. If yes, how often?
1=only when there is a lack of food
2=only during feasts and rituals
3=frequently

XI. **TYPICAL CONSUMPTION PATTERNS**.

H86 _____ 86. How many meals did these people have yesterday? (Number of meals)

H89 _____ 89. Do you consider these meals adequate to maintain the health of all the household members?
1=yes 2=no

We would also like to ask you about your diet during the hungry period (January to May).

H91 _____ 91. How meals do you customarily prepare daily during hungry period?

H92 _____ 92. In general, are these hungry period meals adequate to maintain the health of all household members?
1=yes 2=no

H96 _____ 96. During the hungry period, was there always food available to purchase from the market or from your neighbors?
1=yes 2=no

I. HOUSEHOLD CHARACTERISTICS

Table IA: Household Characteristics

Name	Family Member Number	This person works on-farm or off-farm 1=yes 2=no	Relation to Head 1=head 2=spouse 3=child 4=parent 5=other kin 6=other	Age	Sex 1=m 2=f	Level of Schooling (enter the last completed year) 0=illiterate 12=post-high school 98=no formal schooling but literate	Marital Status 1=monogamous 2=polygamous 3=single 4=widowed 5=divorced 6=emigrant wife (husband out longer than six months
	MEM	CA1	CA2	CA3	CA4	CA5	CA6
	1		Head				
	2						
	3						
	4						
	5						
	6						
	7						
	8						
	9						
	10						
	11						

IV. PRODUCTION

Table IV: Characteristics of Production

Product		**Quantity harvested**		**Quantity harvested in a normal year**		**Existing stocks at harvest time**		**Month in which last year's stock ran out**	**Amount to be stored from this year's harvest for consumption**		**How long will this year's stocks last?**	**Quantity reserved for seed**	
1=corn 2=beans 3=manteiga beans 4=manioc 5=rice 6=sorghum 7=cotton	8=peanuts 9=cashew nuts 10=cashew drink 11=cane drink 12=coconut 13=coconut drink others	Unit 1=sack 100 2=sack 50 3=kilo 4=liter 5=can 20	Qt	Unit 1=sack 100 2=sack 50 3=kilo 4=liter 5=can 20	Qt	Unit 1=sack 100 2=sack 50 3=kilo 4=liter 5=lata 20	Qt	(enter the month)	Unit 1=sack 100 2=sack 50 3=kilo 4=liter 5=can 20	Qt	(enter the month or "all year", if appropriate)	Unit 1=sack 100 2=sack 50 3=kilo 4=liter 5=can 20 other	Qt
PROD		**P1A**	**P1B**	**P2A**	**P2B**	**P3A**	**P3B**	**P4**	**P5A**	**P5B**	**P6**	**P7A**	**P7B**

V. AGRICULTURAL SALES

Table V: Sales of Farm Products

Sale	Crop	Quantity sold		Period of sale	Motive for sale at this time	Buyer	Locale of sale	Distance from the farm	Why sold to this buyer	Value of Sales		Who in the household is responsible for the sale
	1=corn 2=manteiga bean 3=beans 4=manioc 5=rice 6=cotton 7=peanuts 8=cashew nut 9=cashew drink 10=cocos others	Units 1=sack 100 2=sack 50 3=kilo 4=liter 5=can 20	No. of Unit	1= planting (Aug-Dec.) 2= hungry period (Jan-April) 3=this year's harvest 4= various times	1=needed money 2=buyers available 3=consumer goods available 4=attractive price	1=lojista 2=wholesaler 3=AGRICOM 4=ambulante 5=brigada 6=company	1=farmgate/ house 2=village 3=locality 4=district 5=province	(enter the kms between farmer and point of sale)	1=the only one available 2=always sell to this one 3=best price 4=transportation provided 5=carries consumer goods	meticais	Unit 1=unit price 2=total value	1=husband 2=wife
VE	**V1**	**V2A**	**V2B**	**V3**	**V4**	**V5**	**V6**	**V7**	**V8**	**V9A**	**V9B**	**V10**
1												
2												
3												
4												
5												
6												
7												
8												
9												

N.B. Not all of the variables that appear in the printed table are in file C-Q5.sav. Only variables VEN, V2a, V2b, V9a and V9b were kept for this exercise. The PROD variable replaces the V1 variable.

ANNEX 3

Computer analysis of survey data - File organization for multi-level data, by Chris Wolf, MSU Department of Agricultural Economics

This is downloadable at http://www.aec.msu.edu/agecon/fs2/survey/index.htm as a separate document in English or French.