SPSS 22 for Windows TUTORIAL

Cross-Sectional Analysis

Short Course Training Materials
Designing Policy Relevant Research and
Data Processing and Analysis with SPSS 22 for Windows*
1st Edition

Margaret Beaver

Department of Agricultural, Food and Resource Economics
Michigan State University
East Lansing, Michigan
February 2014

Components of the Cross-Sectional Training Materials

Section 0 - Introduction to the file structure for IBM SPSS Statistics (Data and Syntax Editors and Output Window)). You should read this section before starting the main tutorial.

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 a word processor.

Annexes

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

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

References

On the Food Security Group web site at MSU there are several survey research training materials which you might find helpful. The website is http://fsg.afre.msu.edu/index.htm. The Survey Research Training Materials link can be found by scrolling down to the end of the page.

Two papers discuss levels of data:

1) Computer analysis of survey data – File organization for multi-level data by Chris Wolf, MSU Department of Agricultural Economics. This document can be downloaded as a separate document in English or French
2) Data Preparation and Analysis by Margaret Beaver and Rick Bernsten. June 2009. (CDIE reference number pending)

Another article of interest which contains guidelines to manage the data, data verification techniques and preparation of data for analysis is:

Survey Data Cleaning Guidelines: (SPSS and Stata). 1st Edition. Margaret Beaver. MSU International Development Working Paper 123. April 2012.

Acknowledgments

Funding for this research was provided by the Food Security III 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.

Table of Contents

IBM SPSS Statistics TUTORIAL
SECTION_0 - File structure for SPSS 22 for Windows
(Data, Syntax and Output windows)

IBM SPSS Statistics 22 opening dialog box

The opening screen for version 22 of the program IBM SPSS gives the user several different choices to begin using the program. On the right side, we can find out what is new with the version of the program under the "What's New:" section. You can cycle through the new items by clicking on the arrow buttons to the right or left . The next box below is "Modules and Programmability": which lists the modules that have been installed based on the license code. The last box on the right side gives us immediate access to tutorials to help understand how to use the program.

On the left side of the dialog box a new dataset or database query can be started under "New Files". Below that is a list of datasets that have recently been opened. You can choose to open one of those datasets or click on the yellow file folder icon to open another dataset not in the list.

If you do not wish to see this opening dialog box, you can place a tick mark in the box next to "Don't show this dialog box in the future", which is in the lower left hand corner of the dialog box. In the lower right hand side, if you click on **OK**, the file highlighted in the upper left is opened. You can also click on **Cancel** to close the dialog box and do nothing.

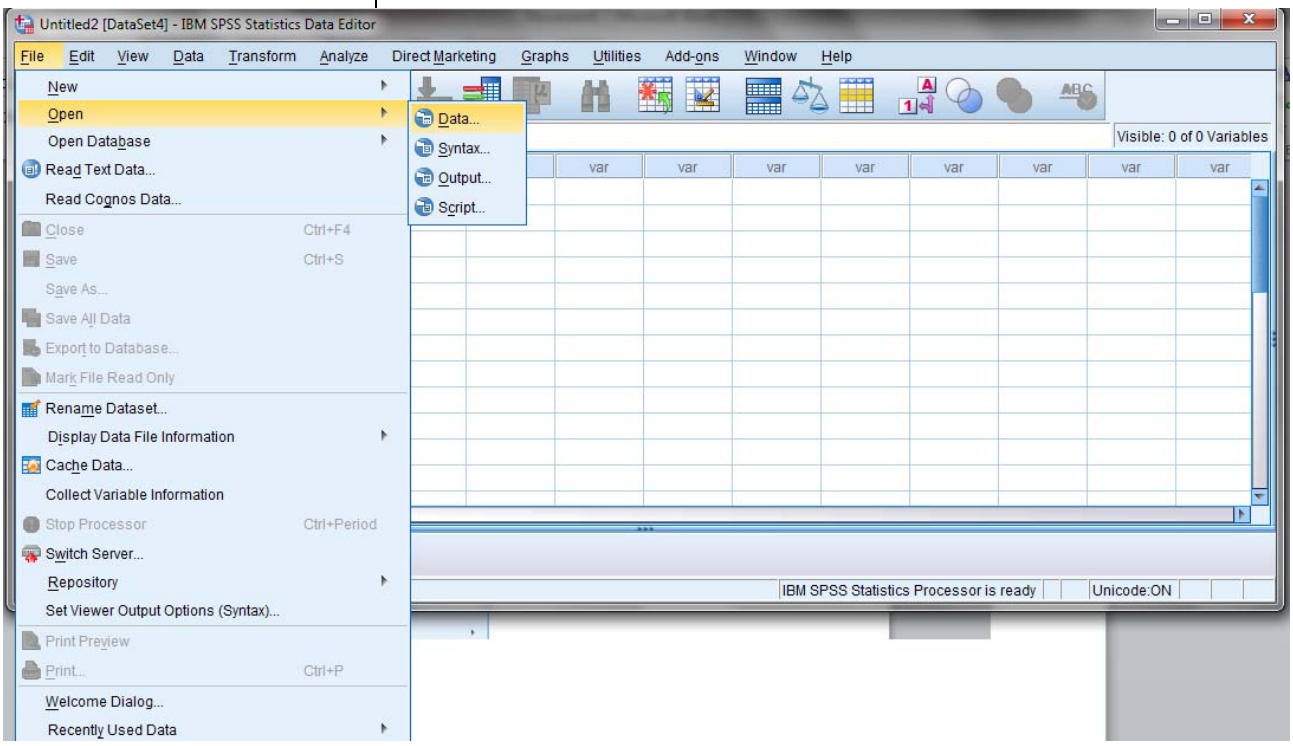
File Types Used in SPSS

This section gives a brief description of the file structure of SPSS for Windows version 22. It is essential that you read through this section before starting the tutorial.

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

When you open SPSS, in the upper left hand corner of the window, select **File**, then **Open.** You will have 4 options of file types from which to select:

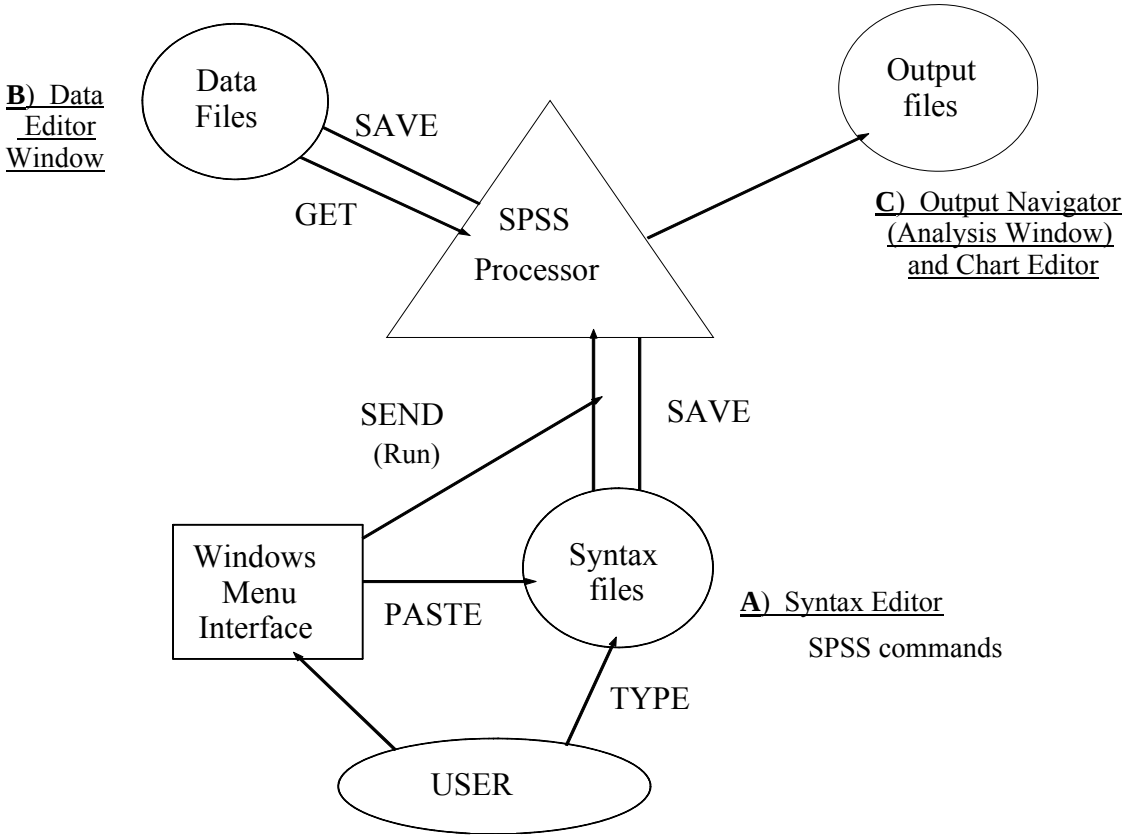
Data	data files	(Extension *.sav)
Syntax	syntax or command files	(Extension *.sps)
Output	output files	(Extension *.spv)
Script	advanced programming files for use with Sax BASIC that are created automatically each time an **Output** is created	(Extension *.sbs)



In the Title bar at the top of the screen you see '**Untitled1 [DataSetx] IBM SPSS Statistics Data Editor**". With this version, more than one data set can be opened within the same SPSS session. Each file that is opened using the menus is given a name which can be used to reference

the dataset in the syntax file. If you are running a syntax file developed in versions before SPSS 15, a name, e.g. [DataSet0], will not be assigned to the dataset. You will see '**Untitled1 []** '. If the dataset does not have a name, the dataset will close automatically if you open another dataset. All datasets must have names in brackets to remain open.

It is important to recognize the significance of the different types of files and to understand the commands you will 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 from the Data Editor 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 22 for Windows,
- To send these commands to the program, SPSS Statistics, 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 **Selection** symbol in the Syntax Editor window toolbar (or select **Run ... Selection** from the Menus). Once you press the **Selection** button,  the computer sends the command(s) to the processor, which reads the commands that were selected from the Syntax Editor and executes them. When all the commands have been processed, SPSS opens the Output Window for you to examine the results of your commands. You can then switch back to the Syntax Editor to 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 Data Editor choices when you create a command using the menus. The commands will also be displayed in your output file.

When you have successfully completed each step in your analysis (or when you are ready to end an SPSS 22 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

SPSS Statistics 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, and measurement level. Before you can do any data analysis in SPSS 22, 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 in the Data Editor window.

Two views of the data are available in the Data Editor window. Look in the lower left of the screen where you can see two tabs. The first - Data View - displays the data in the variables. The second - Variable View - displays the data dictionary. The dictionary includes the names of the variables, the type of variable, the format, the variable labels, value labels, if any values are declared as user missing, column width, alignment, measure, and role. To switch between the views, click on either of the tabs at the bottom of the screen on the left.

You will often open 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 the 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 or you can give the file a new name.

C) The Output Window

SPSS automatically writes all messages and output that result from the execution of your commands to the IBN SPSS Statistics Output window. For example, if you run a frequency command, the frequency table will be written to the Output window. Similarly, if you generate a table or a graph, the table or graph will appear in the Output window.

To save the contents of the Output window to a file, make the Output window active by clicking on that window, pull down the **File** menu and select **Save As...**. When you give the file a name, SPSS will automatically attach the *extension* .spv. It is very important to save the *output file* if you want to review what you have done at a later time. Note that this file cannot be viewed in earlier versions of the SPSS program.

The Output file gives you access to your results after your SPSS 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 Output window 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 (e.g. Windows Explorer). In the Output window, there are two panes: the one on the right contains the results from a command, 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 using the menus with **File / Open / Syntax** . The default extension name is *.SPS. You can have more than one syntax file open. The "active" syntax window is the one with a red plus in the upper left-hand corner of the title bar.

Output files contain statistical output, data information and presentation (tables, graphs, charts), generated by the SPSS 22 for Windows processor, given selected commands. They do not contain data. Output files are made accessible to SPSS for Windows using the menus with **File / Open / Output** where the file is placed in the SPSS Output window. The default extension name is *.spv . You can have more than one output file open. The "active" output window is the one with a blue plus in the upper left-hand corner of the title bar.

Data files contain data, including original survey variables plus new variables created using various SPSS 22 for Windows commands such as the COMPUTE or AGGREGATE commands. Data files are made accessible to SPSS for Windows using the menus with an **File / Open / Data** which places the file in the Data Editor. The default extension name is *.SAV. A data file can also be opened using syntax commands. With version 22, more than one data file can be open in the same session. The "active" data set is the one with a red plus in the upper left-hand corner of the title bar.

SPSS data files can be saved in compressed format. If you have very large files, using this format can reduce the size of the file. The extension is .zsav and can be picked from the dropdown box for Save as type in the Save data as dialog box. Only SPSS version 21 and later can open this type of file. Data files can be saved to many different formats included Stata, and Excel. The dropdown box shows the different types.

Syntax command rules

All commands must end with a period (full stop) as a command terminator. SPSS uses the period as an indication that the command is complete. The command can continue to several lines without the user needing to specify that the command continues to the next line. SPSS will read all lines until it comes to a period (full stop). A blank line following a command with also act as a command terminator.

Upper and lower case does not make a difference for the actual command. SPSS will read the command

FREQUENCIES

or it can read

Frequencies

with the same result. Either command will be executed. SPSS will automatically translate everything to uppercase before executing the command. Most commands can be abbreviated to 3 or 4 characters, e.g. **fre** is the same as **frequencies**.

SPSS Statistics SAMPLE SESSION
SECTION 1 - Basic functions: SPSS files, Descriptives and Data Transformations

Introduction

This is a self-paced training aid designed to introduce the commands needed for some typical statistical survey analyses using **SPSS Statistics 22 for Windows**. This tutorial is intended to be a stand-alone training tool. To use it most effectively, you should ask a knowledgeable SPSS 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 are 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**. Three tables were made available and can be found at the end of the manual in the Annex 2 (for further information please contact Dr. Michael Weber at webermi@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 to standard units.

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 of your choice on your hard disk. If you have not done so already, you need to unzip the files from sample.zip to this folder.
- Under **Options….** in the **Edit** Menu the following items are set:

  In the General tab check to see that
  - Under the section for Variable lists the radio button next to **Display names** should be selected and the radio button next to **File** should be selected. This means the names rather than the variable labels will be displayed in dialog boxes and the variables are ordered by the way

they were defined rather than alphabetical order.

In the Viewer tab

- The box to the left of **Display commands in the log** is checked (lower left corner of the dialog box)

In the Output tab

- Names and Labels are selected for both the Outline Labeling and Pivot Table labeling for the Variable labels
- Values and Labels are selected for both the Outline Labeling and Pivot Tale labeling for the Variable Value labels

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

After making these selections, click on **OK** to set the options.

Important: *Always remember to SAVE the changes to the data after each exercise and section, 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.spv. 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 Output window where SPSS 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 have been entered into SPSS are 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.

SPSS writes the folder structure where a file has been saved into the syntax file whenever commands are pasted. Example:

```
GET
  FILE='C:\Users\name\Documents\sample\C-Q1A.SAV'.
```

There is a command called FILE HANDLE where you can specify the default folder where you want to work. The FILE HANDLE uses a temporary variable name to store the folder information. You use this temporary variable name (replacing the actual folder structure specification) to reference the folder. Using this command at the beginning of the syntax file will facilitate sharing of your syntax files with other people, where they will only need to edit the FILE HANDLE command before running your syntax file. An example of this command is:

FILE HANDLE command

FILE HANDLE *sample* /NAME='C:\Users\beaverm\Documents\sample'.

"*sample*" is the temporary variable that stores the folder reference 'C:\Users\beaverm\Documents\sample'.

When a data file is opened, it is loaded from the disk into memory making it the working file. This means that the data from this file are 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 Data *dialog box.*
2. Change to the folder 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. Two lines of commands were written to the Syntax Editor.
   If the Syntax Editor *window was not open, it will now become the active window and you will see the command to open the data file. If there was a syntax windows already open, you will need to switch to that window to see the command.*

```
GET
  FILE='C:\Users\name\Documents\sample\C-Q1A.SAV'.
DATASET NAME DataSet1 WINDOW=FRONT.
```

4. Edit the syntax file to add the FILE HANDLE command. Replace the directory reference in the GET FILE command to use "sample". Also edit the DATASET command line to change the name of the dataset to be the name of the data file. Change "DataSet1" to "c_q1a" so that it looks like
   `DATASET NAME demog WINDOW=FRONT.`

5. We also want to add comments to define what the purpose of the syntax file is. Above the command to open the data file you can type what the purpose of the syntax file is, your name and the date you created the syntax file as well as any other comments that will help you remember what the syntax file is for. Example:

   * purpose - SPSS tutorial session 1 - basic descriptives and data transformatiaons
   * author and current date
   (example: /* beaver – 6 Jan 2014 */)

The syntax should look like:

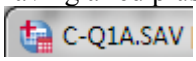
```
*SPSS tutorial session 1 – basic descriptives and data transformations
*Author and date (Beaver – 6 January 2014

FILE HANDLE sample /NAME='C:\Users\name\Documents\sample'.

GET
  FILE='sample\C-Q1A.SAV'.
DATASET NAME demog WINDOW=FRONT.
```

6. **Now, you must block all the lines** and then click on the Selection  button on the Toolbar.

Note that the commands, FILE HANDLE, GET FILE *and* DATASET NAME*, that you just ran will be written to the Output window.*

The Data Editor becomes the active window and the household-member data file is now in opened. Because SPSS 22 can open multiple data files in the same session, each file will be given a "DATASET NAME" which allows us to specify which data file (if more than one is open) should be used when a command is run in the Syntax Editor. If you have multiple datasets open, only one will be considered "active" by the program. The "active" dataset is identified by having a red plus appearing in the upper left hand corner of the title bar. 

It is recommended that you change the dataset name from the default that SPSS gives it to a unique name so that you can always know what the dataset name is and you can consistently reference it with the DATASET commands.

NOTE: If you do not specify which dataset to use with the "DATASET ACTIVATE" command, SPSS will use whichever dataset has the red plus sign in the upper left corner.

DATASET commands

Since more than one data file can be opened in the same SPSS session, the command - DATASET – is used to manage the data files. The DATASET command has several key word options. They are:

Command	Example
DATASET NAME name [WINDOW={ASIS }] {FRONT}	GET FILE='c:\data\spssdata.sav'. DATASET NAME *file1* window = front. SORT CASES BY ID. GET FILE 'c:\data\moredata.sav' DATASET NAME *file2* window = front. SORT CASES BY ID.
DATASET ACTIVATE name [WINDOW={ASIS }] {FRONT}	GET FILE='c:\data\spssdata.sav'. DATASET NAME *file1*. COMPUTE AvgIncome=income/famsize. GET DATA /TYPE=XLS /FILE='c:\data\exceldata.xls'. DATASET NAME *file2* window = front. COMPUTE TotIncome=SUM(income1, income2, income3). DATASET ACTIVATE *file1*.
DATASET DECLARE name [WINDOW={MINIMIZED}] {HIDDEN } {FRONT }	DATASET DECLARE corrmatrix. REGRESSION /DEPENDENT=var1 /METHOD=ENTER= var2 to var10 /OUTFILE=CORB(corrmatrix). DATASET ACTIVATE corrmatrix.

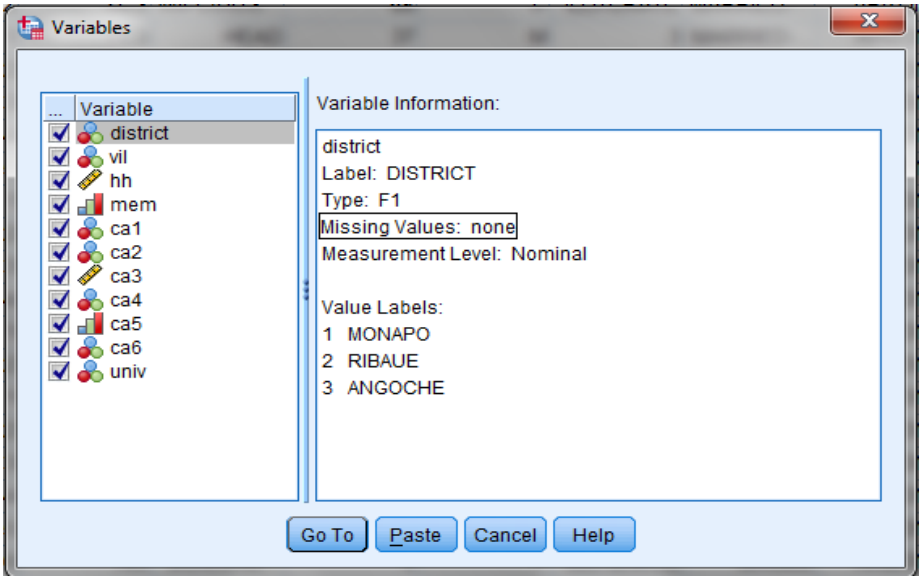
Command	Example
DATASET COPY name [WINDOW={MINIMIZED}] {HIDDEN } {FRONT }	DATASET NAME original. DATASET COPY males. DATASET ACTIVATE males. SELECT IF gender=0. DATASET ACTIVATE original. DATASET COPY females. DATASET ACTIVATE females. SELECT IF gender=1.
DATASET CLOSE {name} {*} {ALL}	DATASET CLOSE file1.
DATASET DISPLAY.	The DATASET DISPLAY command displays a list of currently available datasets. The only specification is the command name DATASET DISPLAY.

Utilities / Variables

One key piece of information we 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, which can be found in all three SPSS windows. You can 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 choice is also on the tool bar  . This dialog box shows definition information about each of the variables. We see the variable names, **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. The symbol to the left of the variable denotes whether the variable type is ordinal, nominal or scale – level of measurement.



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

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

Pull down the **File** menu, choose **Display Data files Information** and select **Working File**.

This command will execute immediately. The Output window will become active and will contain a listing of all the variables with their definitions.

DISPLAY DICTIONARY command

The SPSS command is

```
DISPLAY DICTIONARY
```

Switch to the Output window to look at the results of this command. You can see the name of each of the variables, the label associated with the variable, and the format as well as other information about each variable. For example: F8.2 means width of 8 with two decimal places. The width is computed by adding the number of digits to the left of the decimal plus the decimal plus the number of digits to the right of the decimal. With a format of F8.2, five digits are displayed to the left and two to the right of the decimal plus the decimal = width of 8 (5+1+2). The DISPLAY DICTIONARY command 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.

CODEBOOK command

The codebook command reports the dictionary information -- such as variable names, variable labels, value labels, missing values -- and summary statistics for all or specified variables and multiple response sets in the active dataset. For nominal and ordinal variables and multiple response sets, summary statistics include counts and percents. For scale variables, summary statistics include mean, standard deviation, and quartiles.

To obtain the codebook information:

1. From the **Analyze** menu select **Reports** and then **Codebook**
2. If you want to include the file name and location, click on the Output tab and place a tick mark on all the options under "File Information".
3. Click on the **Paste** button to place the command in the Syntax Editor. Switch to the Syntax Editor and run the command.

Note that the command has several different subcommands:

```
CODEBOOK  district [n] vil [n] hh [s] mem [o] ca1 [n] ca2 [n] ca3 [s] ca4 [n]
        ca5 [o] ca6 [n]
  /VARINFO POSITION LABEL TYPE FORMAT MEASURE ROLE
        VALUELABELS MISSING ATTRIBUTES
  /FILEINFO NAME LOCATION CASECOUNT
  /OPTIONS VARORDER=VARLIST SORT=ASCENDING MAXCATS=200
  /STATISTICS COUNT PERCENT MEAN STDDEV QUARTILES.
```

Example output from this command:

File Information

File Name		C-Q1A.SAV
Location		C:\Users\beaverm\Documents\sample
Number of Cases	Unweighted	1524
	Weighted	1524

district

		Value	Count	Percent
Standard Attributes	Position	1		
	Label	DISTRICT		
	Type	Numeric		
	Format	F1		
	Measurement	Nominal		
	Role	Input		
Valid Values	1	MONAPO	444	29.1%
	2	RIBAUE	602	39.5%
	3	ANGOCHE	478	31.4%

Variable View

Another method to use if you want to look at the structure of each variable, is to select the Variable View at the bottom left of the Data Editor 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. An example of this view is shown in Table 1.1 on the next page for the variable DISTRICT, 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 to see the choices, add a new value, 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. 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 22

Number of the variable	**Name**	**Type**	**Width**	**Decimals**	**Label**	**Values**	**Missing**	**Columns**	**Align**	**Measure**[1]	**Role**[2]
	Variable name	Numeric or alpha-numeric (*String*)	Space required by the variable	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*	Some dialogs support predefined roles that can be used to pre-select variables for analysis. When you open one of these dialogs, variables that meet the role requirements will be automatically displayed in the destination list(s).
Example:											
1	district	Numeric	1	0	DISTRICT	1= MONAPO 2= RIBAUE 3= ANGOCHE	None	8	Right	Nominal	Input

[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).**

[2] There are several types of role choices. You can explore the command by looking at the help in SPSS. Available role subcommands are:
Input. The variable will be used as an input (e.g., predictor, independent variable).
Target. The variable will be used as an output or target (e.g., dependent variable).
Both. The variable will be used as both input and output.
None. The variable has no role assignment.
Partition. The variable will be used to partition the data into separate samples for training, testing, and validation.
Split. Included for round-trip compatibility with IBM® SPSS® Modeler. Variables with this role are not used as split-file variables in IBM® SPSS® Statistics.

Descriptive Statistics - involving one variable

The first step at the beginning of analysis is to run descriptive statistics (e.g. counts, 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 are like, and to be sure 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 you time later and improve your 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, you would be alerted 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 (scale) variables, while **Frequencies** is used for categorical (nominal and ordinal) variables.

Continuous / categorical variables definition

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 Annex 2, 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. If you are in the Syntax Editor, click on the Go To Data  button on the Toolbar.
2. Scroll through the data.

```
A period in a field indicates a missing value
 or sysmis.
```

Scrolling through the data will give you a "feel" for what is in the data file. 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 may have no real significance.

DESCRIPTIVES command

By examining the data, you should have found that variable **ca3** (age) is continuous (or scale) 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 arrow  button

ca3 will move to the **Variable(s)**: *box on the right*

3. Click on the  button to place the command into the Syntax window. (If the Syntax Editor did not become active, you can go there by clicking on the syntax icon on the windows taskbar at the bottom of your screen.)

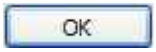
4. Execute the command by clicking on the Selection  button located on the Toolbar. (Note that this time we did not have to move the cursor since it was already positioned in the last line of the **Descriptives** command.)

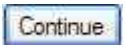
The Output window will become active and the results of the command will be there. You will see that the mean for age (**ca3**) is 21.34 years.

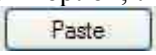
FREQUENCIES command with a chart

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. Remove the check mark ✓ next to the words "Display Frequency Table". An information box pops up to tell you that you have turned off all output and that you must select an item from the charts or statistics dialog box or no output will be displayed. Click on 

8. Click on the  button and select the radio button next to "Histograms" and check ✓ the "Show normal curve on histogram" option; then click on the  button.

9. Click on  to put the command into the Syntax Editor and switch to the Syntax Editor to make it active.

10. Execute the command by clicking on the Selection  button located on the Toolbar.

In the Output window you can see the output of the histogram of the distribution of ages.

Save the Output File

Now that you have output in the SPSS Output window, it is a good time to save that output file. Switch to the Output window if you are not in that window already. Click on the **File...Save as...** on the SPSS toolbar at the top left. 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 **.spv** to indicate an output file.

FREQUENCIES command

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 those variables. To run a

frequencies, do the following:

1. **Analyze...Descriptive Statistics...**select **Frequencies ...** *The* Frequencies *dialog box will open.*
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. You could also include district and vil.
5. Click on **Paste** to paste the command into the Syntax Editor. Switch to the Syntax window to make it active.
6. Execute the command by clicking on the **Selection**  **b**utton located on the Toolbar.

The Output window will become the active window. The first table in the output is called the Statistics. The Statistics table is important in that it tell you how many cases are valid and how many are missing. If there should be no missing values, this output immediately tells you there might be a problem with the data. The remaining output is a frequency table for each individual variable. 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.

Explore (EXAMINE) command

Another command used to produce many types of descriptive statistics is the Explore command. One useful output for this statistic is that it produces a list of cases that can be considered *outliers*. This command also produces graphs of the distribution of data using a stem and leaf chart or a histogram. The default is a stem and leaf chart. The Explore command can produce large amounts of output if used with its defaults. We will limit the output to statistics. You can explore the other options on your own. Within each of the dialog boxes, there is a **HELP** button on the right which will explain the statistic.

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 switch to make it active.
8. Click on Selection 

You see the Descriptives Table which shows you the standard descriptives 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.

Go To Case

To find a case by the case number, in the Data Editor, select **Edit...Go to Case**, or you can click on the icon in the tool bar,  . A dialog box opens, type the case number and click on **OK**.

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...** from the SPSS menu at the top left. 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; however each will have a different 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.

Exercise 1.1:

Apply what you've just learned about descriptive statistics. Run descriptive statistics on another sample file. Use the production questionnaire - Table IV, whose data are in the file C-Q4.SAV.

Hints:

a. make C-Q4.SAV your working data file. Note that SPSS did not close the data file that was open. It opened the new file and gave it a label of "Dataset2".
   You will see the text

```
GET
  FILE=''C:\Users\name\Documents\sample\C-Q4.SAV'.
  DATASET NAME DataSet2 WINDOW=FRONT.
```

Remember to change the reference to the directory to "sample" and the name of the dataset so that if you need, you can reference the dataset specifically. Change the name to ***prod****, e.g.*

```
    DATASET NAME prod WINDOW=FRONT.
```

Your syntax should read

```
GET
  FILE='sample\C-Q4.SAV'.
  DATASET NAME prod WINDOW=FRONT.
```

*You now have 2 datasets open. The data file with a red **plus** will be the "active" dataset. You must be sure that dataset "*C_Q4.sav"* with the dataset name of "prod" has the red "plus" when you switch to the syntax window to run commands that are specifically for that 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:

prod PRODUCT

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	3 cotton	83	4.9	4.9	4.9
	5 peanuts	144	8.5	8.5	13.4
	6 rough rice	175	9.2	9.2	22.6
	8 bananas	50	3.0	3.0	25.5
	9 sweet potato	12	.7	.7	26.2
	10 cashew liquor	24	1.4	1.4	27.6
	11 sugar cane liquor	11	.6	.6	28.3
	13 dried cashew	2	.1	.1	28.4
	17 sugar cane	13	.8	.8	29.2
	21 cashew nut	130	7.7	7.7	36.9
	23 coconut	45	2.7	2.7	39.5
	30 beans	279	16.5	16.5	56.0
	31 manteiga beans	7	.4	.4	56.4
	35 sunflower	5	.3	.3	56.7
	38 oranges	13	.8	.8	57.5
	39 cashew fruit	44	2.6	2.6	60.1
	41 manioc	338	20.0	20.0	80.0
	44 sorghum	124	7.3	7.3	87.4
	47 maize	192	11.3	11.3	98.7
	50 "ossura"	5	.3	.3	99.0
	67 tobacco	4	.2	.2	99.2
	68 tomato	13	.8	.8	100.0
	Total	1693	100.0	100.0	

Descriptive Statistics

	N	Minimum	Maximum	Mean	Std. Deviation
p1b PROD THIS YR - # OF UNITS	1670	.0	5000.0	26.353	163.4359
p2b PROD NORMAL YR - # OF UNITS	1798	.5	5000.0	22.817	179.5101
p3b STOCK ENTERING HARVEST - # OF UNITS	173	.0	30.0	2.523	4.5746
p5b STORED FOR CONS THIS YR - # OF UNITS	1231	.0	1460.0	17.612	86.1036
p7b STOCK FOR SEED - # OF UNITS	869	.0	100.0	4.938	6.8755
Valid N (listwise)	171				

Descriptive Statistics - involving two or more variables

CROSSTABS command

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

Look at the household member questionnaire in Annex 2, Table IA. One thing you might be interested to know is how the gender of the respondents varied by relationship to the head of household, i.e., 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. If you still have the data file open that was used for the exercise, you can close that file by clicking on the "X" in the upper right hand corner of that data file. However, it would be better to place commands in the syntax window to close the file. As listed above under options for the "DATASET" command, we can type a command to activate the first file, C-Q1A.SAV, and close the file that we used for the exercise.

Switch to the Syntax Editor and below the last command in the file, type the following:

```
DATASET ACTIVATE demog.
DATASET CLOSE prod.
```

The first command makes the first data file we opened (c-q1a.sav, which was given the dataset name of "demog") the active data file and closes the second data file (c-q4.sav, which was given the dataset name of "prod"). Block these commands and run them.

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 a ✓ 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. There is an option to display a Clustered bar chart. Place a ✓ in the box.
10. Click on **Paste**
11. 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. By default the **Crosstabs** command just gives counts. 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 down all the cells in a column. 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 (Row percents). The table also shows that of the females in the sample, 2.9% are heads (Column percents). Below the table a bar chart appears giving a visual display of the distribution of gender within relationship to head.

MEANS command

The **Compare Means** command is somewhat similar to **Crosstabs,** but it gives statistics about continuous variables controlled by a categorical variable. 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 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 table. 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. The default output for this command is to provide means, number of observations and standard deviation. To obtain more statistics, click on the **Options** button in the upper right side of the dialog box. Select a statistics (e.g. median) from the left hand box and move it to the right hand box. Click on **Continue.**
6. Click on **Paste.**
7. Run the command from the Syntax Editor.

This command calculates means of the dependent variable (ca3), which is normally a continuous variable. The means will be calculated separately for each different value of the independent variable, which is a categorical variable ca2, "relation to household head".

From this output you find that the average age of head of households is 41.53 years while the average age of the spouse is 33.19 years.

Data Transformations

After examining the results of the descriptive statistics you might 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 next 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 grouped into 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 into a Different Variable (RECODE command)

To categorize a continuous variable, you use the **Recode** command. Categorizing a continuous variable makes detailed information more general. You want to keep the detailed information as well as the new general information. Therefore, you must recode the variable into a new 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. We want to preserve the original ages and store the categorized values in a separate variable. We will use the menu choice - **Recode Into A Different Variable.**

Let's recode the variable **ca3** into a different variable called **age_gp** for age groups**.**

1. Select **Recode Into A Different Variables** from the **Transform** menu
2. Select **ca3** from the list on the left
3. 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.
4. Click once in the empty box next to **Name:** in the **Output Variable** section to put the cursor there.
5. Type **age_gp** in the box.
6. Click once in the empty box next to **Label:** in the **Output**

Variable section.

7. Type **Age Group** in the box for the label.
8. Click on **Change** to move the variable name into the **Numeric Variable -> Output Variable:** box.
9. Click on **Old and New Values...**
   The Recode into Different Variables: Old and New Values dialog box will appear.
10. In the Old Value section click on the circle next to **Range: _____ through _____**
11. Type **0** in the first box
12. Press **<Tab>** and type **10** in the second box.
13. Press **<Tab> twice.**
    *Your cursor will now be in the box next to **Value:** in the **New Value** section.*
14. Type **1** for the value of the first category.
15. Click once on **Add**
16. 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.**
17. To recode ages **61** and higher to **4,** click on the circle next to **Range: _____ through highest**
18. Enter **61** in the box and repeat steps 14 and 17 using 4 for the value.
19. Click on **Continue**
20. Click on **Paste**
21. 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.
```

22. Run the selected commands.

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 look 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-q1a.sav [demog] - SPSS Data Editor**.
2. Scroll through the Data Editor using the scroll bars.

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

1. Switch to the Data Editor window if you are not already there.
2. Select the Variable View tab from the bottom left.
3. The variable **age_gp** is on line 12.
4. First, in the cell under the Decimal column, type 0.
5. Second, in the cell under the Width column, type 1.

These changes tell SPSS for Windows to display **age_gp** with a width of 1 digit with no decimal places. This procedure can also be done with syntax, which we highly recommend. Should you need to rerun your syntax, the formatting will be done with the syntax file.

FORMATS command

Switch to the Syntax Editor. At the end of the commands, type the following:

```
FORMATS age_gp (F1.0).
```

Now the command is in the syntax and it not required that you manually change the format. In the parentheses F stands for fixed. 1 is equal to the width display, and 0 is the number of decimals. To learn about other formats, place your cursor in the line where the formats command is and click on the tool  (Syntax Help).

When you Recode into a new variable, it does not have *Value Labels*. The statistical output from SPSS can include 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.

```
Note: with SPSS 13 and later, variable names
are no longer limited to 8 characters.
However, if you share your data files and
syntax with someone who is using an earlier
version of SPSS, that person will not be able
to open the data files or run the syntax if you
use longer variable names.
```

VARIABLE LABELS command

Names of variables 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 indicates M (male) and 2 indicates F (female) . 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 create the recode command.

The command format is:

VARIABLE LABELS var1 ‘label associated with var1’.

It is not available from the menus.

Adding value labels cannot be done from the menus. To add the **Value Labels** manually use the following steps:

1. Switch to the Data Editor and click on the tab for *Variable View*

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, 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 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. Click to the Data View tab to look at the variable. **age_gp** is now displayed as a single digit when value labels are off and value labels should show when value labels are on.
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.

VALUE LABELS command

However, the best way to include value labels is to add the command in the syntax file. You must type the command. An example is below.

```
VALUE LABELS age_gp
    1 '0 to 10'
    2 '11 to 19'
    3 '20 to 60'
    4 '61 and older'.
```

Measurement Level (VARIABLE LEVEL command)

Measurement level of the variable is important for this version of SPSS to be able to specify the correct statistic in the command to produce Tables and also for the graphing module. Measurement level was discussed on page 16. The measurement level is either nominal, ordinal or scale. To set the measurement level the command is "VARIABLE LEVEL". The "EXECUTE" command is required for this passive command.

```
VARIABLE LEVEL age_gp (ORDINAL).
EXECUTE.
```

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. It is a good practice to save a file under a different name to preserve the original data 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 you are in the Data Editor window (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,** switch to the Syntax Editor, replace the directory reference with "sample" 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, switch and run it.

From the output, 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 obtained 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, Output window, and Data windows.

To switch to the Output window:

1. From the **Window** menu select Session1.spv – IBM SPSS Statistics 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 side 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 (available in Annex 2), find out the number of households in each district that have 1-4, 5-7, and more than 7 persons per household.

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**.

hhsize Household groups * district DISTRICT Crosstabulation

			district DISTRICT			
			1 MONAPO	2 RIBAUE	3 ANGOCHE	Total
hhsize Household groups	1 1 thru 4	Count	65	48	74	187
		% within hhsize Household groups	34.8%	25.7%	39.6%	100.0%
		% within district DISTRICT	60.7%	40.3%	64.3%	54.8%
		% of Total	19.1%	14.1%	21.7%	54.8%
	2 5 thru 7	Count	39	56	36	131
		% within hhsize Household groups	29.8%	42.7%	27.5%	100.0%
		% within district DISTRICT	36.4%	47.1%	31.3%	38.4%
		% of Total	11.4%	16.4%	10.6%	38.4%
	3 8 thorugh highest	Count	3	15	5	23
		% within hhsize Household groups	13.0%	65.2%	21.7%	100.0%
		% within district 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 hhsize Household groups	31.4%	34.9%	33.7%	100.0%
		% within district DISTRICT	100.0%	100.0%	100.0%	100.0%
		% of Total	31.4%	34.9%	33.7%	100.0%

Looking at the results, for group 1 (households with a member size from 1 to 4) 34.8% are in Monapo, 25.7% in Ribaue and 39.6% in Angoche (row percents). In the district, Monapo, 60.7% of all households have 1 to 4 members in a household,, 36.4% have 5 to 7 members and 2.8% have 8 or more members. The % of total shows the percent each category combination is of all the cells in the table.

Before exiting SPSS for Windows we should save the contents of the Output window. 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 Output window the active window using its icon in the Windows taskbar.
2 From the **File** menu select **Save As...**
3 Enter the filename **session1**
   The .spv extension will be added to the name automatically.
4. Click on ***Save***

To exit SPSS for Windows, switch to the Data Editor:

1. From the **File** menu select **Exit**
2. A dialog box will prompt you to save the contents of Syntax Editor . Click on ***Yes***
3. A dialog box will prompt you to save the contents of ….\sample\c-hh.sav. Click **No**
   SPSS Statistics will close.

SPSS for Windows SAMPLE SESSION
SECTION 2 - Restructuring Data Files - Table Lookup & Aggregation

Introduction

For some types of analyses 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. See Annex 3.

The analysis we did in Section 1 was done at each level separately, using just the variables in a single file. 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 harvested. 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 at 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 kgs 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 53; for a 50 kg sack of cotton it is 17.5, while a 50 kg sack of manioc root is 33.33. 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. The weight is different for each product. Thus, we must first convert all production into a standard unit which will be kilograms.

Second, we want to know calories produced by each household, not kilograms. After converting all production to kilograms, we must convert it into 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 for specific food crops, not the total calories from each separate product. After we convert all production to calories, we must select those crops and then sum the calories within each household to arrive at the household total.

This tutorial assumes that no data files are open at this point. The Data Editor has no data.

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 the command. A syntax file is opened.
   ```
   Remember to add all the information at the top of the
   syntax file to explain what the syntax file is about and
   include the "FILE HANDLE" command that sets the folder
   where you are working.(Page 13)
   ```
4. Change the directory reference to say "sample" and edit the dataset name to "prod". Run the two commands.
   ```
   Remember to block both commands.
   ```

First we want to 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 add a variable to the active file where each case has both the data from the production file and a variable containing the conversion factor for that product-unit combination.

If you are working with large datasets, the merge can be done more quickly if the table lookup file is indexed. To index the lookup file containing the conversion values, open the CONVER.SAV file. An option is given to create an index for this table lookup file to facilitate and increase the speed to merge the files.

1. Select **File** / **Open / Data...**
2. Select the file name conver.sav
3. Paste the command, switch to the Syntax editor, change the path reference, edit the dataset name to "conver", block the commands and run them. This file is the active dataset now.
4. From the **Data** menu select **Sort Cases...**
   ```
   The Sort Cases dialog box will come up.
   ```
5. Select **prod** and click on  to move it into the Sort by: box.
6. Select **unit** and click on  to move it into the Sort by: box.

7. Place a ✓next to the box labeled Save file with sorted data
8. Click on the FILE button to give the file a new name. In the File name: box type **CONVER_index** and click on the SAVE button.
9. Back in the Sort variables dialog box, place a ✓ next to Create an index .
10. Paste the command. Edit the folder reference to indicate only the file handle name, then run the command.

Note: If you are working with smaller datasets it is not necessary to index the lookup file.

The file is saved. Now make the production file the active file and close the conver_indexed data file. In the syntax file type

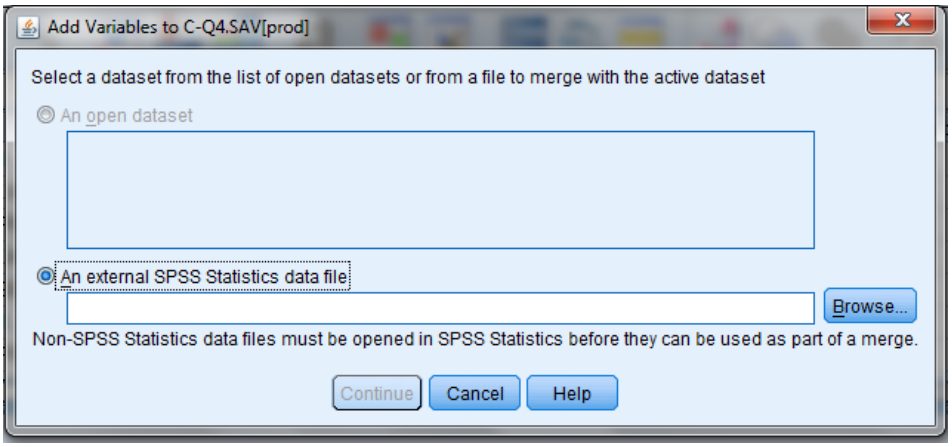
```
DATASET ACTIVATE prod.
DATASET CLOSE conver.
```

Block these two commands and run them.

Merge files –file-table lookup merge (STAR JOIN 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 the working data file and CONVER_index.SAV. We are doing a merge where the second file is our "Lookup Table". The variable we are adding from the CONVER_index.SAV file will be placed at the end of the active dataset with this command .

1. From the **Data** menu select **Merge Files**, then select **Add Variables...** *The* Add Variables to C-Q4.SAV [c_q4]: *dialog box will open. We have no other open datasets so the default is to use an external SPSS data file.*



2. Click on the radio button next to An external SPSS Statistics data file. Click on the **Browse** button to select the file conver_index.sav
3. Click on **Open** .
4. Click on **Continue .**

The variables used to match cases must have the same names. We must select **unit** from the "New Active Dataset" and move it into the box for Excluded Variables. We will rename it to **p1a** to match the variable name in the production file that contains the unit of measure.

5. Select **unit** from the list under New Active Dataset: on the right and click on  to move it into the Excluded Variables box.
6. Click on **Rename**...
   This will allow you to rename **unit** *to* **p1a** *to match the working data file name for this variable.*
7. Next to New Name: type **p1a**
8. Click on **Continue**

We cannot select the variables to match by until we select how we want to match cases.

9. Check the box ✓ next to Match cases on key variables in sorted files
10. SPSS assumes you are doing a table lookup merge so it pre selects Non-active dataset is keyed table for you. Since the keyed table is not our active file, this is fine.

Now we need to tell SPSS what the key variables are to match by.

11. Select **prod** from the Excluded Variables: list
12. Click on  next to Key Variables: (bottom, right)
13. Repeat steps 11 and 12 for **unit -> p1a**
    Note that the variable "unit" no longer exists, it has been renamed to p1a.
14. Paste the command and switch to the Syntax Editor. Fix the folder reference to match the FILE HANDLE variable . Run the command.

This command is quite different from the earlier versions of SPSS but works the same way as the MATCH FILES with one exception. The MATCH FILES required both files be sorted by the key variables. This command does not require that the files are sorted. This is an active command and does not require the "EXECUTE".

This command is the equivalent of a SQL left outer join. The active file variables are specified by a prefix of "t0" and the lookup file variables are prefixed with "t1". Multiple lookup files can be merged at the same time with different key variables. Another enhancement is that if you want to match by string variables, the string size of the variables in the two files does not have to be the same width. Password protection has been added with version 22. Read the HELP on this command for further information.

```
STAR JOIN
  /SELECT t0.district, t0.vil, t0.hh, t0.p1b, t0.p2a, t0.p2b, t0.p3a, t0.p3b, t0.p4,
     t0.p5a,
    t0.p5b, t0.p6, t0.p7a, t0.p7b, t0.univ, t0.dia, t0.mes, t0.ano, t1.conver
  /FROM * AS t0
  /JOIN 'sample\CONVER_index.SAV.sav' AS t1
    ON t0.prod=t1.prod
    AND t0.p1a=t1.unit
  /OUTFILE FILE=*.
```

The above steps tell SPSS to merge the active data file (active in your Data Editor window) and the CONVER_index.SAV file, (using CONVER_index.SAV as a table lookup) to add the *conver* variable to our active data file. Since the key variables need to have the same names in both files we renamed **unit** (the variable from the conversion file) to match the name in our working dataset **p1a** . You can still use the MATCH FILES command in SPSS 22 – it is just not available from the menus.

The STAR JOIN command reorders the variables so that the key variables that were used in the merge appear first.

Key Variables are required in a Merge where one of the files is being used as a keyed table. Our key variables specify doing the lookup by product and p1a, because we have a different conversion factor for each product-p1a (or unit) combination. If we had used only **prod**, SPSS would expect each product to have only a single conversion factor, 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 active dataset now contains the needed conversion factor variable, **conver**. For every product-unit combination, **conver** contains the value required to convert the quantity harvested to kilograms. It is always important to verify if the merge was successfully completed. Switch to the Output window, if you are not there, and check the LOG for error messages. If there is an error message, the merge was not done correctly. Return to the Data Editor 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.

CAUTION: You can only run a Merge (MATCH FILES) command once. If the merge did not work, generally, you must open the original data file, and run all the commands up to the merge command, fix the problem with the merge command and then run the merge.

COMPUTE command

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

1. From the **Transform** menu select **Compute Variable...**
2. Under Target Variable: enter **qprod_tt** (for total quantity of production in kg)
3. Click on **Type & Label** to add a label for **qprod_tt**. Click on the radio button next to "**Use expression as label**", 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 the three commands and run them.

Switch to the Data Editor and scroll to the right to the end of the variables to find the new variable, which is always added at the end of the file. Look to be sure you see numbers in this new variable. If you only see periods, you have forgotten to include the "**EXECUTE**." command when you blocked the syntax. You can check by looking in the message area at the bottom right of the Data Editor. If you see "**Transformations Pending**", you need to run the "**EXECUTE**" command.

Verify that the compute command has done what you expected it to do. Did you multiply **the quantity produced** by the **conversion factor**? Try calculating a few numbers by hand. Household 3 in district 1 – vil 1 produced 16 – 100 kg sacks of cotton, If you multiply 16 by 35, do you get 560?

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 (prod). To add the calorie conversion variable to the active data file we need to do another merge with keyed table lookup. This time the key variable only needs to be the product variable.

1. From the **Data** menu select **Merge Files** then **Add Variables...**
2. Click on the radio button next to An external SPSS data file. Click on the **Browse** button to select the file calories.sav
3. Click on **Open** .
4. Click on **Continue** .
5. Check the box ✓ next to Match cases on key variables in sorted files
6. Click on radio button next to Non-active dataset is keyed table
7. Select **prod** from the Excluded Variables: list
8. Click on  next to Key Variables: (bottom, right)
9. Paste the command
11. Select and run the command

The dataset now contains the needed calorie variable, **calories**; check the output to be sure there are no error messages and check to see that the variable exists at the end of the dataset. Some products do not have any calories, so you should expect to see missing values. Maize grain (PROD=47) should have 3590 calories per kilogram in the **calories** variable. We are now ready to compute total calories produced.

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

SELECT IF command

We now have a variable that contains the total calories produced per product for each household. We are only interested in the seven staple food crops:

(prod=5) peanuts,
(prod=6) rice,
(prod=30) nhemba bean,
(prod=31) manteiga bean,
(prod=41) manioc,
(prod=44) sorghum, and
(prod=47) maize

We can find these product codes 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 the command **Select Cases**.

Select Cases selects a subset of the cases based on particular criteria. The data 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 turn a filter on (which we will be doing because it is a safer method) you can always turn the filter off to make the whole dataset available for further analysis.

1. From the Date Editor window, select **Data** / **Select Cases**
   You should see the Select Cases *dialog box.*
2. Select the radio 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 = 5 | PROD = 6 | PROD = 30 | PROD = 31 | PROD = 41 | PROD = 44 | PROD = 47
   The "|" is a symbol for the word OR. We are telling SPSS to select all cases with **prod** = *47 or* **prod** = 30 *or* **prod** = 31...
6. Click on **Continue**
7. Under the OUTPUT section of the dialog box, the radio button next to Filter out unselected cases should already be selected.
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.

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

SPSS creates a variable called **filter_$** which contains values of 0 and 1. 0 = not selected, 1 = selected. Those cases with a 0 will have a slash in the case number column at the left.

Only cases with these product codes will now be used for all active commands. Note that the filter command does not affect any COMPUTE statements (passive command). All cases will be used with a COMPUTE command, even if the filter has been set. This subset of the data will be in effect for analysis until we turn the filter off. To turn the filter off, you would choose **Data / Select Cases / All cases** (unfilter the cases).

We are now ready to calculate the total calories produced per household for these specific staple food products. To do this, we need to sum, for each household, the values of **cprod_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 household-product level file which will contain only one case per household. SPSS uses the term "AGGREGATE" to collapse the number of cases at one level to a new higher level. We will sum all the cases for household to one case for household.

AGGREGATE command

To create the new household-level file, we use **Aggregate**. **Aggregate** will create a new data file with one case per household where the variable **cprod_tt** is summed across the products for each household.

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 **cprod_tt** from the left hand side list of variable and move it to the Summaries of Variables(s): box
4. The default function is to compute a mean. We want to sum the values. We must change the function. Click on **Function...**
5. Under "Summary Statistics", click on the radio button next to Sum and click on **Continue**
6. Click on **Name & Label...**
7. Change the default name **cprod_tt_sum** *to* **cprod_tt**
8. Enter the label: Calories Produced in Staple Foods
9. Click on **Continue**
10. In the "**Save**" section of the dialog box, select the radio button next to Create a new dataset containing only the aggregated variables. In the "dataset name" box, type hh_file1
11. Paste the command.

In the Syntax Editor *you see the commands:*

```
DATASET DECLARE hh_file1.
AGGREGATE
  /OUTFILE='hh_file1'
  /BREAK=district vil hh
  /cprod_tt 'Calories Produced in Staple Foods'
  = SUM(cprod_tt).
```

These two commands are required. The "dataset declare" command creates a new dataset called hh_file1. The "aggregate" command places the new data in the new dataset.

12. After the last command, add another command to make this new dataset the active dataset. Type in the syntax window:
```
DATASET ACTIVATE hh_file1.
```
13. Block all three commands (DATASET DECLARE, AGGREGATE and DATASET ACTIVATE. Run these commands.

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 **cprod_tt**, which we calculate by summing **cprod_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. **cprod_tt**).

The original file we started with (C-Q4.sav [prod]) is still loaded in memory. In the taskbar, you can see that the new dataset with a dataset name of hh_file1 has a red plus on the icon. This is our new active dataset. The first icon is the Output window, the second is the original dataset we opened which contains the production data, the third is the Syntax Editor, the fourth is the new dataset which is *untitled3[hh_file1]. This dataset has the red plus and is our active dataset.

This dataset has not yet been saved to disk, which is why we see *Untitledx in the title bar – this dataset only has a dataset name [hh_file1] to permit SPSS to reference it to distinguish it from the other dataset which is open.

The new dataset contains the variables we need for our analysis: total number of calories from staple foods produced per household, plus the key variables to identify a household (district, vil, hh). There should be only one case per set of key variables, i.e. one case per household.

Let's look at the aggregated variable. Run a **Descriptives** on **cprod_tt**. You should find that the average number of calories produced per household per year is 4,483,964.7. Look at the LOG file in the Output window. You should see the following:

```
DATASET ACTIVATE hh_file1.
DESCRIPTIVES
  VARIABLES=cprod_tt
  /STATISTICS=MEAN STDDEV MIN MAX .
```

SPSS inserted a command to make the new dataset the active dataset. (DATASET ACTIVATE hh_file1.)

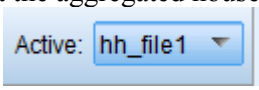
We want to save this dataset using the **Save As...** from the **Fil**e menu.

1. Switch back to the Data Editor. Be sure that the red plus sign is in the title bar of this dataset with a name of hh_file1.
   If this dataset does not have the red plus, you will be saving the c_q4.sav file to a new name.
2. Select **Save As...** from the **File** menu

3. Name the file **hh_file1**
4. Paste, switch to the syntax editor, edit the directory structure on this "SAVE AS" command to replace the directory structure with our variable that contains the directory structure – "**sample**", block the command and run it.

We want to close the production file (C-Q4.sav). Its dataset name is "prod". In the Syntax Editor, type

DATASET CLOSE prod.

Be sure that the aggregated household file is active. In the syntax editor you should see 

in the second tool bar line. Run the DATASET CLOSE command. C-Q4.SAV should no longer be an open dataset in SPSS. You should have only one dataset open with a dataset name of hh_file1. The file name on disk is hh_file1.sav.

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 Toolbar
2. Select the file name c-q1a.sav
3. Paste the command, change the FILE HANDLE reference name to "sample", change the dataset name to "demog" and run, blocking the commands.
4. Close down the dataset called hh_file1 by typing the command to close the dataset into the syntax file and run it.

   DATASET CLOSE hh_file1.

There should only be one dataset open - the file c-q1a.sav.

The rules we will use to calculate 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

The adult equivalents are indicating what percent of a standard calorie load is needed to sustain a person. For example, males 10 years and older need 100% of the calories. A female 10 to 19 years old needs only 84% as many calories as a male 10 years or older, and children under 10 need only 60% as many calories as the typical male 10 years and older. For each person (case) in the member file we need to look at sex, **ca4**, and age, **ca3**, to calculate the adult equivalent for that person.

COMPUTE / IF (IF command)

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

1. From the **Transform** menu select **Compute Variable...**

   The Compute Variable *window will appear.*
2. For the Target Variable: enter **ae**

3. Select the **Type & Label** button and enter Adult equivalent for the Label. Click on **Continue**
4. In the Numeric Expression: box enter a **1**
5. Click on the **If...** button.
6. Select the radio button for "Include if case satisfies condition:"
7. Enter the statement **ca4 = 1 & ca3 >= 10**
8. Click on **Continue**
9. Paste the command but don't run it yet.
10. Repeat steps 1, and 3-8 replacing the previous information with the following.

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

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 make editing changes to the variables names and criteria. If you prefer to use the menus, follow the steps above.

11. 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 .
```

Check the Output window to be sure there are no error messages in the LOG.

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

1. Select **Analyze / Descriptive Statistics / Frequencies...**
2. Select **ae**
3. Paste and run

There are 1524 total cases. Also, 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. SPSS will not compute a value for a variable if any of the components of the expression are user or system missing. Missing values should have been identified during the cleaning process and noted. At this point a researcher should go back to the original questionnaires to try to fill in the missing data. Since we can't do this, we will use an alternative method.

If we leave these values missing, the adult equivalent value of those households will be smaller than they actually are, which will distort our results somewhat. 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 adult equivalents. We know that the adult-equivalent values range from a low of 0.6 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 to select is **ae**
3. Don't forget to paste before you run the command

The results show that the average value of **ae** for all individuals is .79, with a standard deviation of only 0.17. 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. 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.)

Recode into the Same Variable

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 the radio button next to 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.

Run a frequencies command to verify the change. You can type the command in the syntax editor and run it from there. SPSS does not require the full command to be typed. Generally only 3 letters are required. In this instance "FRE" is sufficient.

```
FRE ae.
```

AGGREGATE command

Now we need to calculate the total number of adult equivalents for each household. The current file is at the member level; however, the values we need should be at the household level. Again we use **Aggregate** to collapse the data from the member level to the household level. A 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** and click on **Continue**
7. Click on the **Function...** to change the function to sum.
8. Select Sum and click on **Continue**
9. In the "**Save**" section of the dialog box, select the radio button next to Create a new dataset containing only the aggregated variables. In the "dataset name" box, type hh_file2
10. Paste the command.

In the Syntax Editor *you see the commands:*

```
DATASET DECLARE hh_file2.
AGGREGATE
  /OUTFILE='hh_file2'
  /BREAK=district vil hh
  /ae_tt 'Adult eduivalents' = SUM(ae).
```

These two commands are required. The "dataset declare" command creates a new dataset called hh_file2. The "aggregate" command places the new data in the new dataset.

11. After the last command, add another command to make this new dataset the active dataset. Type in the syntax window:
    ```
    DATASET ACTIVATE hh_file2.
    ```
12. Block the three commands and run.

The active dataset will be hh_file2 which shows the total adult equivalents for the household. The original file we started with (C-Q1A.sav [c_q1a]) is still loaded. In the taskbar, you can see it does not have a red plus on the icon.

We no longer need the c-q1a file open. We can close it using the syntax editor. Type the command:

DATASET CLOSE demog.

Be sure that the red plus is in the new dataset called hh_file2, before you run this command from the syntax editor.

NOTE: *If the file you are trying to close is the "active" file, the data file will not close, it will only delete the dataset name so that in the title bar you will see c_q1a[]. If this is the file that is active when you try to run the DATASET CLOSE command, you will have to give it a dataset name (from the menus – File, Dataset Name), then activate the hh_file2 dataset, then run the syntax from the syntax window.*

The variable **ae_tt** is the total adult equivalents for that household. Run a **Descriptives** on **ae_tt**.

1. **Analyze / Descriptive Statistics / Descriptives...**
2. Variable to select is **ae_tt**
3. Paste and run.

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

Look at the LOG section in the Output window. You should see the following:

```
DATASET ACTIVATE hh_file2.
DESCRIPTIVES
  VARIABLES=ae_tt
  /STATISTICS=MEAN STDDEV MIN MAX .
```

This completes step 2. Save this file to disk as **hh_file2.sav**.

1. Be sure that the red plus sign is in the title bar of this new dataset.

If this dataset does not have the red plus, you will be saving the c_q1a.sav file to a new name.

2. **File / Save As...**
3. Filename hh_file2
4. Paste, edit the directory structure to replace the actual directory with the temporary variable containing that information "sample" and run.

Step 3: Join the two files created in steps 1 & 2 together to compute calories produced per adult equivalent per day.

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, household by household, 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.

Merge files – file-file merge (MATCH FILES command)

*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.*

You are now ready to merge the two household level files. Both files must be sorted in the order of the variables that you will use to match the two file. The aggregate command assures us that the files have been sorted by the variables that identify a household. As long as you have not sorted the files differently and then saved, there is no need to open and resort them. To merge them, click on

1. **Data / Merge Files / Add Variables...**
2. Click on the radio button next to An external SPSS data file. Click on the **Browse** button to select the file hh-file1.sav
3. Click on **Open** .
4. Click on **Continue** .
5. Check the box ✓ next to Match cases on key variables in sorted files
6. To be able to match the two files we need to place a ✓ in the box next to Cases are sorted in order of the key variables in both datasets *Note that after this box is checked, the radio option next to Both files provide cases has been selected.*
7. Move the variables: **district**, **vil**, and **hh** respectively, into the Key Variables: box.
8. You will see a Warning box reminding you that both files must be sorted by the key variables or the merge will file. Click on OK.
9. Paste, clear warning, select both the MATCH FILES command and the EXECUTE command and run.

Merge Files added the variable for total calories to the active dataset. The two variables you need to compute calories produced per adult equivalent are now in the same file; the title bar still indicates the name of the data file is hh_file2.sav with a dataset name of [hh_file2].

Total calories produced (**cprod_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 (**cprod_ae**).

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

RANK CASES command

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, showing how each 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 adult equivalent per day. 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 **cprod_ae** to Variable(s):
3. Move **district** to By:
4. Click on **Rank Types...**
5. Remove the check mark next to Rank
6. Select Ntiles: 4
7. **Continue**
8. Paste and run

   The Output window should pop up where you can see a table describing the new variable that has been created - **Ncprod_a.**

The first step was to specify the variable to use for the ranking—in this case **cprod_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) which tells SPSS to break the variable into quartiles. From this command, SPSS will create a new variable that contain the ntile rankings and generate a name for that variable.

MEANS command

We can now use **Means** to produce the values to fill in our table.

1. **Analyze / Compare Means / Means...**
2. Move **cprod_ae** to Dependent List:
3. Move **ncprod_a** to Independent list: layer 1 of 1

   ncprod_a *came from the* **Rank Cases** *procedure.*
4. Click on **Next**
5. Move **district** to Independent List: layer 2 of 2
6. Paste and run.

You should note that the mean for the entire population is 4,014.518 and the mean for the 2nd quartile in Ribaue is 2,517.455. The output from **Compare Means** gives you the numbers necessary for the final 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.

If you want to remove the count and standard deviation from the output table, you can go back to the command and

7. Click on **Options** , select "number of cases" and "standard deviation" from the "Cell Statistics" box and move them back into the "Statistics" box. Click on **Continue** .
8. Paste and run.

Save this dataset as **hh_file3.sav**.

1. Make the Data Editor window active
2. **File / Save As...**
3. Filename is hh_file3
4. Paste, change the directory name to the variable "sample" and run

You should 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 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 for your work and enables you to run the exact same analysis over again at a future date. Documenting now can save much time and many steps later.

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

1. **File / Exit**
   SPSS will prompt you to save the contents of the windows that have not been saved; in this case the Output window*.*
2. Save the Output window 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 the listing below, 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 in quartitles by district.
*Beaver - January 2007.

*set the file handle.

FILE HANDLE sample /NAME='C:\Documents and Settings\aec_user\My
Documents\sample'.

GET
  FILE='sample\sample\C-Q4.SAV'.
DATASET NAME prod WINDOW=FRONT.

********Step 1 ********.

*create an index for the lookup file to facilitate the merge.
*need necessary to do unless the files are very large.

SORT CASES BY prod(A) unit(A)
  /OUTFILE='sample\CONVER_index.sav' INDEX=YES SIZE=DEFAULT.

*merge in the lookup file to standardize to kgs from other units.
STAR JOIN
  /SELECT t0.district, t0.vil, t0.hh, t0.p1b, t0.p2a, t0.p2b, t0.p3a, t0.p3b, t0.p4,
t0.p5a,
    t0.p5b, t0.p6, t0.p7a, t0.p7b, t0.univ, t0.dia, t0.mes, t0.ano, t1.conver
  /FROM * AS t0
  /JOIN 'sample\CONVER_index.SAV.sav' AS t1
    ON t0.prod=t1.prod
    AND t0.p1a=t1.unit
  /OUTFILE FILE=*.
*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.

STAR JOIN
  /SELECT t0.p1a, t0.district, t0.vil, t0.hh, t0.p1b, t0.conver, t0.qprod_tt, t0.p2a,
t0.p2b,
    t0.p3a, t0.p3b, t0.p4, t0.p5a, t0.p5b, t0.p6, t0.p7a, t0.p7b, t0.univ, t0.dia,
t0.mes, t0.ano,
    t1.calories
  /FROM * AS t0
  /JOIN 'Documents\sample\CALORIES.SAV' AS t1
    ON t0.prod=t1.prod
  /OUTFILE FILE=*.
```

```
*calculating total calories produced.

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

*setting filter to select only staple foods.

USE ALL.
COMPUTE filter_$=( prod = 5 | prod = 6 | prod = 30 or prod = 31 or prod = 41 or
  or prod = 44 or | prod = 47).
VARIABLE LABEL filter_$ 'prod = 5 | prod = 6 or prod = 30 or prod = 31 or'+
 ' prod = 44 or prod = 44 or prod = 47 (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.

DATASET DECLARE hh_file1.
AGGREGATE
  /OUTFILE='hh_file1'
  /BREAK=district vil hh
  /cprod_tt 'Calories produced in staple foods' = SUM(cprod_tt).

*verify variable is created and value is reasonable.

DATASET ACTIVATE hh_file1.
DESCRIPTIVES
  VARIABLES=cprod_tt
  /STATISTICS=MEAN STDDEV MIN MAX .

*save household level file.

SAVE OUTFILE='sample\hh_file1.sav'
  /COMPRESSED.

************************************************.

*Step 2 - generate a household level file containing the number of adult
 equivalents per household.

*roster (demography file).
GET
  FILE='sample\C-Q1A.SAV'.
DATASET NAME demog WINDOW=FRONT.

*close the datasets that are open that are no longer needed.
dataset close prod.
dataset close hh_file1.
```

```
*assigning adult equivalents to each member of the household based on
  gender and age.

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.

DATASET DECLARE hh_file2.
AGGREGATE
  /OUTFILE='hh_file2
  /BREAK=district vil hh
  /ae_tt 'Adult Equivalents' = SUM(ae).

DATASET ACTIVATE hh_file2.

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

*close the other dataset.
DATASET CLOSE demog.

SAVE OUTFILE='sample\hh_file2.sav'
  /COMPRESSED.

*step 3 join the total calories produced per household (hh_file1) with the total
adult equivalents per household (hh_file2).

MATCH FILES /FILE=*
 /FILE='sample\hh_file1.sav'
 /BY district vil hh.
EXECUTE.

* calculate calories produced per adult equivalent per day.

COMPUTE cprod_ae = cprod_tt / ae_tt / 365 .
VARIABLE LABELS cprod_ae 'Calories produced per adult equivalent per day' .
```

```
EXECUTE .

*ranking data by district using the calories produced per adult equivalent per
day into quartiles.

RANK
  VARIABLES=cprod_ae  (A) BY district  /NTILES (4) /PRINT=YES
  /TIES=MEAN .

MEANS
  TABLES=cprod_ae  BY ncprod_a  BY district
  /CELLS MEAN COUNT STDDEV  .

*just show the mean.
MEANS
  TABLES=cprod_ae  BY ncprod_a  BY district
  /CELLS MEAN .

SAVE OUTFILE='sample\hh_file3.sav'
  /COMPRESSED.
```

Exercise 2.1:

Produce similar output using calories retained (production minus sales) instead of calories produced. The table should show calories retained per adult equivalent per day using the same seven food crops. The output should be broken down by district and calorie retention 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. Computing the calories sold involves the same basic steps as computing the calories produced. (Step 1), refer to Step 1's instructions above.
f. Merge this newly created file, (the file containing calories sold), with the file containing calories produced, hh_file3.sav. (step 3 above)
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. Compute calories retained per adult equivalents per day (cret_ae).
j. Use the cret_ae to rank the data into quartiles.
k. Use the **Compare Means** command to show calories retained by **district** and **quartile**.
l. Save the data file.
m. Save the contents of the Syntax Editor.

Below is an example of the output you should produce:

Report

cret_ae Calories retained per adult equivalence

Ncret_ae NTILES of cret_ae by district	district DISTRICT	Mean	N	Std. Deviation
1	1 MONAPO	1148.0448	27	409.61445
	2 RIBAUE	1232.8030	29	350.22596
	3 ANGOCHE	912.7559	28	384.74681
	Total	1098.8770	84	401.03778
2	1 MONAPO	2211.3833	27	205.71992
	2 RIBAUE	2145.8446	30	202.81780
	3 ANGOCHE	1698.5099	29	168.49973
	Total	2017.5753	86	297.99128
3	1 MONAPO	3314.8568	28	477.12339
	2 RIBAUE	3126.3578	30	329.89358
	3 ANGOCHE	2405.0077	29	336.48560
	Total	2946.5741	87	547.14537
4	1 MONAPO	7619.1018	27	3557.13545
	2 RIBAUE	5759.0391	30	1649.58387
	3 ANGOCHE	4954.7625	29	2426.82446
	Total	6071.8027	86	2821.27091
Total	1 MONAPO	3570.9752	109	3032.69607
	2 RIBAUE	3081.4162	119	1902.73924
	3 ANGOCHE	2506.4982	117	1957.99071
	Total	3044.2336	343	2370.14648

TABLES

Using **Custom 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, **Custom tables** allows you to do the following:

1. 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.)

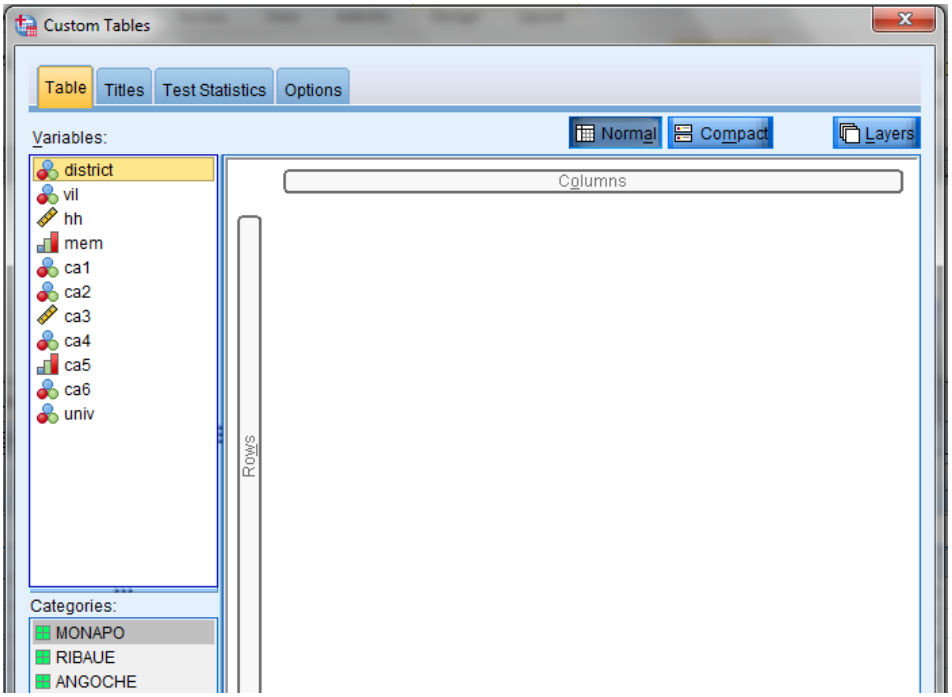
2. Manipulate table structure, content, and presentation format.

3. Include flexible percentages, specifying the base for the percentages (their denominator) so that they add to 100% across rows, columns, sub-tables, or whole tables.

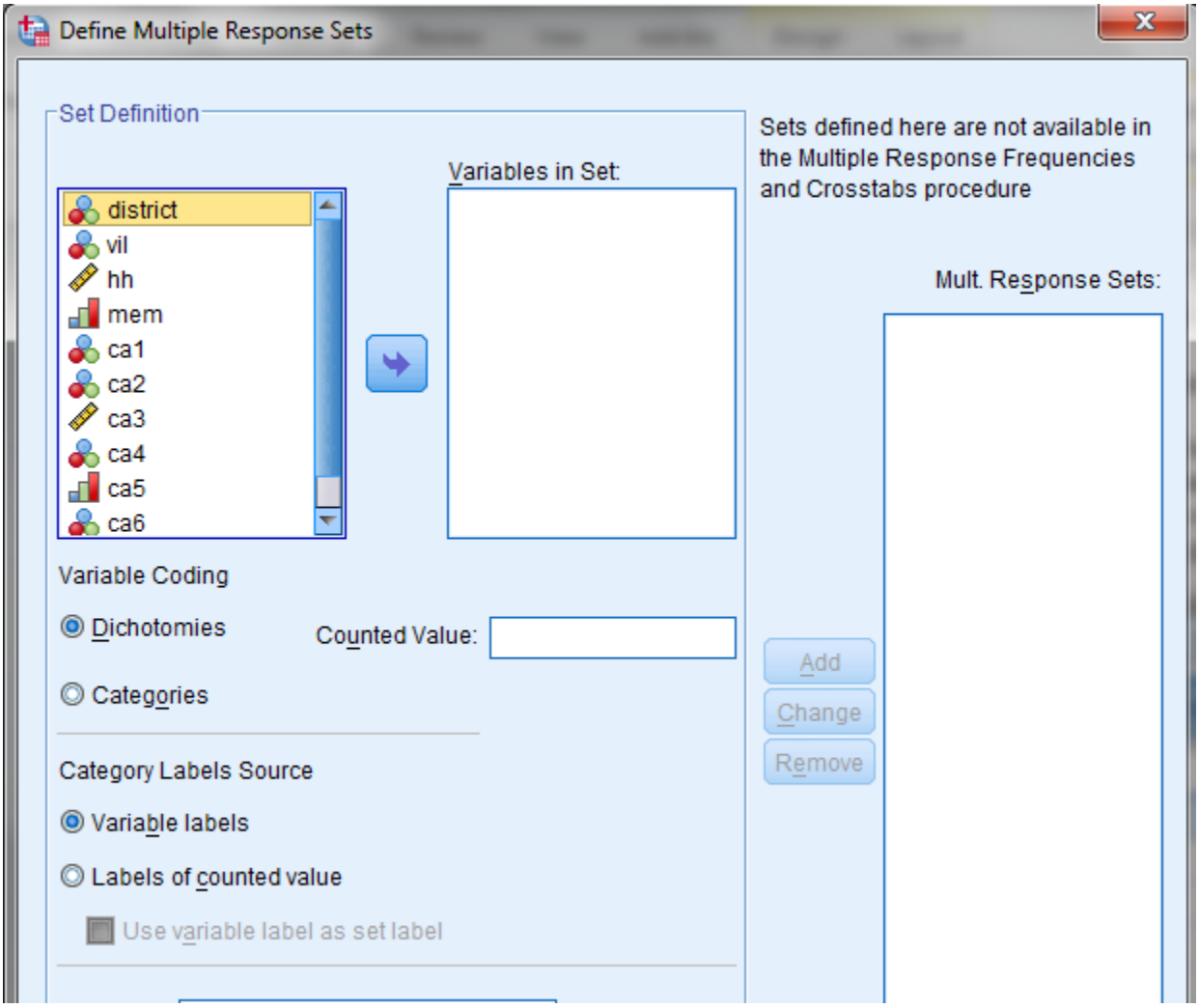
4. Display up to 60 characters for variable labels and value labels.

With this version of SPSS there are 2 choices under the **Tables** menu: Custom tables and multiple response sets.

Custom tables - A canvas pane will open. You build a table by dragging and dropping variables onto the rows and columns of the canvas pane. You can see a preview of the table that will be created. The pane does not show actual data values in the cells, but should show a fairly accurate view of the layout of the final table.



Multiple response sets – dialog box to define multiple response variables



CROSSTABS vs. TABLES

Let's compare the **Crosstabs** procedure with the **Custom tables** procedure for cross tabulation.

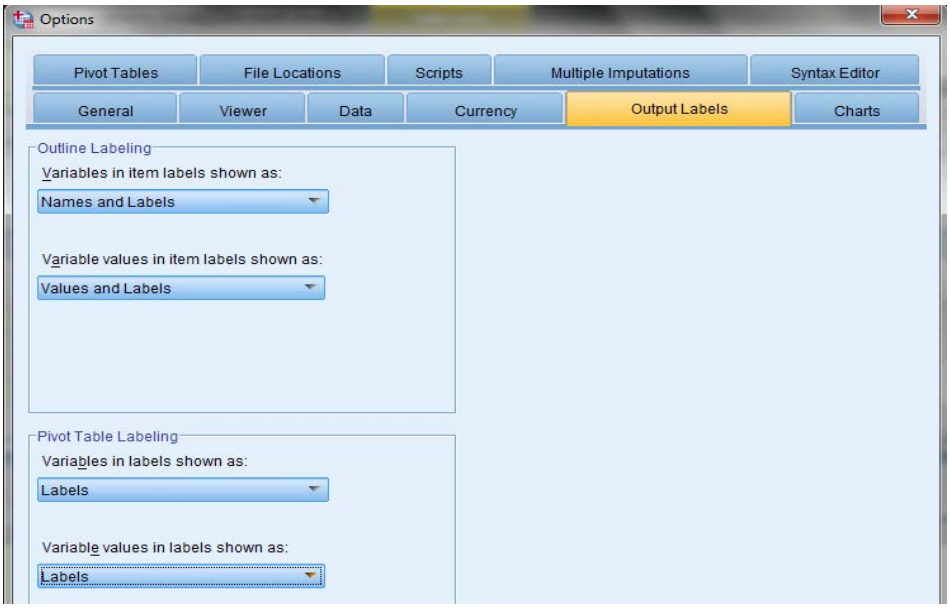
Open the member file we created that contains the age variable, Q1A-AGE.SAV.

1. **File / Open / Data...**
2. Select q1a-age.sav
3. Paste, change the dataset name to "d_age", change the directory structure to "sample", select both commands and run.

We are ready to produce tables that we want to use in our report. We do not want to see the actual values of the categorical variables in the output, only the value labels. We'll change the options for output to show only value labels.

1. **Edit / Options...**
2. Click on the **Output Labels** tab
3. Change the boxes under **Pivot Table Labeling** to show labels only

4. Click on **OK**



Now we will do a simple cross tabulation using the **Crosstabs** command.

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				Total
			0 to 10	11 to 19	20 to 60	61 and older	0 to 10
RELATION TO HEAD	HEAD	Count	0	6	296	41	343
		% within RELATION TO HEAD	.0%	1.7%	86.3%	12.0%	100.0%
	WIFE/HUSBAND	Count	0	25	280	5	310
		% within RELATION TO HEAD	.0%	8.1%	90.3%	1.6%	100.0%
	SON/DAUGHER	Count	503	184	31	0	718
		% within RELATION TO HEAD	70.1%	25.6%	4.3%	.0%	100.0%
	MOTHER/FATHER	Count	0	0	5	1	6
		% within RELATION TO HEAD	.0%	.0%	83.3%	16.7%	100.0%
	OTHER RELATIVE	Count	70	55	16	2	143
		% within RELATION TO HEAD	49.0%	38.5%	11.2%	1.4%	100.0%
Total		Count	573	270	628	49	1720

	% within RELATION TO HEAD	37.7%	17.8%	41.3%	3.2%	100.0%

CUSTOM TABLES (CTABLES command)

Let's use the **Custom Tables** command to produce the same table:

1. **Analyze / Tables / Custom Tables...**
   A small dialog box opens that asks if you want to define the variable properties. Click on **Ok**
2. Drag **ca2** to Rows and drop it onto the canvas pane
3. Right click on the variable **age_gp** and change the variable measure to "Ordinal", if it still has a measurement of scale. Drag the variable to Columns and drop it onto the canvas pane
4. Click on the variable **ca2** so that it is highlighted. Then under "Define" in the lower left side of the dialog box click on the button labeled **Summary Statistics.** (You could also right-click on that variable and choose the same option.)
5. The statistic **count** has already been placed in the Display: box. We also want the row percent. Select **Row N %** from the Statistics: box and click on the arrow to move it to the Display: box.
6. Click in the cell for the label on this row and change the label to "**%**" **.**
7. Click on **Apply to Selection**
8. In the lower center box labeled "Summary Statistics" next to Position: click on the dropdown arrow to change the position from columns to rows.
9. Paste and run.

The command is:

* Custom Tables.

```
CTABLES
  /VLABELS VARIABLES=ca2 age_gp DISPLAY=DEFAULT
  /TABLE ca2 [COUNT F40.0, ROWPCT.COUNT '%' PCT40.1] BY age_gp [C]
  /SLABELS POSITION=ROW
  /CATEGORIES VARIABLES=ca2 age_gp ORDER=A KEY=VALUE
EMPTY=INCLUDE.
```

Below is the output:

			Age group			
			0 to 10	11 to 19	20 to 60	61 and older
RELATION TO HEAD	HEAD	Count	0	6	296	41
		%	.0%	1.7%	86.3%	12.0%
	WIFE/HUSBAND	Count	0	25	280	5
		%	.0%	8.1%	90.3%	1.6%
	SON/DAUGHER	Count	503	184	31	0
		%	70.1%	25.6%	4.3%	.0%
	MOTHER/FATHER	Count	0	0	5	1
		%	.0%	.0%	83.3%	16.7%

	OTHER RELATIVE	Count	70	55	16	2
		%	49.0%	38.5%	11.2%	1.4%
	OTHER	Count	0	0	0	0
		%	.0%	.0%	.0%	.0%

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**.

You can add titles and remove the label for the Relationship to Head. Go back into the Custom Tables dialog box, either by clicking on the icon labeled 'Recall recently used dialogs' or selecting from the menus.

1. Highlight the Relationship to head variable in the Rows. Right click and remove the ✓ to the left of Show variable label

Add a title:

2. Click on the **Titles...** tab at the top of the dialog box.
3. In the Title box type:
   Table 1: SPSS for Windows Sample Session.
   Press <Enter>
   Type **Age Breakdown by Relation to Head**
4. In the Caption box type:
   Source: Nampula family sector household survey, 1991.
5. In the Corner box: type **Relationship to Head**

Add a total:

6. Highlight Age Group, right click and select "Categories and Totals". Place a ✓ (tick mark) in the box next to "Total". Click on **Apply.**
7. Change the Summary Statistics position from Rows to Columns.
8. Highlight Relation to Head, right click and select "Categories and Totals". Place a ✓ (tick mark) in the box next to "Total". Click on **Apply.**
9. Paste and run the command.

Below is the table produced from that command. Note: From the output window you may change the table properties, change the format of the values, use the pivot tray to rearrange the table and other various options that are available.

Table 1: SPSS for Windows Sample Session.

Age Breakdown by Relation to Head

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

Source: Nampula family sector household survey, 1991.

We want to save the output file.

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

To be able to use TABLES effectively, you will need to practice creating tables. Open the Help Menu and under the Topics tab, scroll down to the Custom Tables Option towards the end of the list. There are many examples of how to use Custom Tables. The benefits of having SPSS produce nice looking tables far outweigh the effort to create the table. For example: with periodic data, such as monthly prices, where each month the table should be updated, using syntax to produce the tables becomes very productive. Once you create a table format you like, you can use that code over and over, changing the variable names and titles as needed.

Compare Means vs. Custom Tables

Next we will compare computing averages using **Compare Means** and **Tables**, based on an example from section 2.

1. **File / Open / Data...**
2. Select hh-file3.sav
3. Paste, change the dataset name to "hh_file3", change the directory reference to "samp", select both commands and run

```
Note that you now have 2 datasets open,
this file is labeled "hh_file3".
```

4. **Analyze / Compare Means / Means...**
5. Move **cprod_ae** to Dependent List:
6. Move **ncprod_a** to Independent list: layer 1 of 1
7. Click on **Next**
8. Move **district** to Independent List: layer 2 of 2
9. Paste and run

Report

Calories produced per adult equivalent per day

NTILES of cprod_ae by district	DISTRICT	Mean	N	Std. Deviation
1	MONAPO	1221.7281	27	416.12856
	RIBAUE	1484.0298	29	422.11606
	ANGOCHE	1272.0519	28	486.25928
	Total	1329.0592	84	452.22243
2	MONAPO	2494.8048	27	377.12144
	RIBAUE	2517.4551	30	366.08053
	ANGOCHE	2431.9673	29	296.80050
	Total	2481.5167	86	345.82242
3	MONAPO	3968.1419	28	621.34028
	RIBAUE	4000.8905	30	549.83405
	ANGOCHE	3640.3535	29	453.28705
	Total	3870.1717	87	562.97704
4	MONAPO	9170.0222	27	4686.21141
	RIBAUE	7520.2527	30	2178.86354
	ANGOCHE	8364.3191	29	4054.90269
	Total	8316.5516	86	3764.16975
Total	MONAPO	4206.4675	109	3813.56406
	RIBAUE	3900.7967	119	2559.31057
	ANGOCHE	3950.2610	117	3390.51145
	Total	4014.5183	343	3271.40106

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 also add the Minimum and Maximum to the table for more information.

1. **Analyze / Tables / Custom Tables...**
2. Move **district** to Columns:
3. Move **Ncprod_a** to Rows:
4. Move **cprod_ae** to just under district in the Columns:
   Note that the statistics changed from count and % to Mean.
5. With Calories… highlighted, click on **Summary Statistics...** which is in the lower left corner of the dialog box.
6. Mean has already been specified. Click in the Format: cell and select the format **n,nnn**, Decimals will be 0
7. Select Minimum from the left side list of statistics, use the label **Min,** Format: **n,nnn**, click in Decimals to change to **0**
8. Select Maximum from the left side list of statistics, use the label **Max**, Format: **n,nnn**, click in Decimals to change to **0**
9. Click on **Apply to Selection**
10. In the lower center box labeled "Summary Statistics" next to Position: click on the dropdown arrow to change the position from columns to rows.
11. Highlight the District variable in the Columns. Right click and remove the ✓ to the left of Show variable label

12. Highlight the Percentiles…. variable in the Rows. Right click and remove the ✓ to the left of Show variable label
13. Highlight the Calories…. variable in the Columns. Right click and remove the ✓ to the left of Show variable label
14. Click on **Titles...** tab.
15. Type in the Title box: **Table 1: Food Production in Calories**. Press <Enter>, then type **per Adult Equivalent per Day**
16. Type in the Corner box: **Production Quartile**
17. Click on **Paste.**

Switch to the Syntax Editor. Your table should look similar to the one below.

Table 1: Food Production in Calories.

per Adult Equivalent per Day

Production Quartile		MONAPO	RIBAUE	ANGOCHE
1	Mean	1,222	1,484	1,272
	Min	294	429	354
	Max	1,956	1,938	1,952
2	Mean	2,495	2,517	2,432
	Min	1,973	2,030	2,024
	Max	3,169	3,120	2,961
3	Mean	3,968	4,001	3,640
	Min	3,176	3,141	2,996
	Max	5,067	4,834	4,563
4	Mean	9,150	7,520	8,364
	Min	5,107	4,984	4,692
	Max	28,466	13,124	20,485

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

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

You can also <right-click>, select Copy, then open up your word processor, <right-click> and Paste. The table comes in as a Word table that you can edit.

Exercise 3.1:

Produce a similarly formatted table using calories retained which you calculated in Exercise 2.1. Include totals by retention quartile. Your table should look similar to:

Food retention in calories per adult equivalent per day

Quartiles		MONAPO	RIBAUE	ANGOCHE	Total
1	Mean	1148	1233	913	1099
	Min	224	429	208	208
	Max	1806	1783	1391	1806
2	Mean	2211	2146	1699	2016
	Min	1807	1790	1396	1396
	Max	2544	2556	1936	2556
3	Mean	3317	3126	2405	2947
	Min	2555	2566	1984	1984
	Max	4303	3730	3055	4303
4	Mean	7619	5759	4955	6072
	Min	4360	3731	3064	3064
	Max	20874	9465	12675	20874
Total	Mean	3571	3081	2506	3044
	Min	224	429	208	208
	Max	20874	9465	12675	20874

Multiple Response Analysis

Occasionally questions are asked that require the respondent to select multiple answers. A single variable cannot record all the answers to this type of question adequately, since 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 together. SPSS provides two analysis options for this type of question.

To analyze groups of variables, they must be defined as a "*set*". A set is defined under Tables / Multiple Response Sets option and will be saved with the data file so the set can be used again.

SPSS allows two different grouping methods, to handle the two different ways to ask a multiple response question.

1. One type of multiple response is where there are several choices, but the respondent is asked to choose only the 3 "most important items". Only three categorical variables are defined to hold the values chosen. These are called "*category*" variables.
2. The other type of multiple response is the "check all that apply" where a value of 1 is assigned if the response is checked,, a value of 0 is assigned if the value is not checked. These are called *multiple dichotomy* variables.

Refer to the **Help / Contents** menu for more detail.

Question 35 in the household questionnaire is an example of a multiple response category question. It asks about crops grown principally to be sold. Each household is asked to specify up to three main crops. Three variables

were defined to hold the codes; these are: **h35a**, **h35b**, and **h35c**. The values allowed for these variables are the same. The question is left open-ended, however, since a code of 6 is allowed to specify another crop. The new crop is written down during data collection and the value for each the new crops are assigned after the survey is completed. The same set of value labels is applied to each of the variables. As you will see with the following commands, 12 different crops were coded for question 35.

Multiple Response sets (MRSETS command)

Category variables

You could run **Frequencies** on each of the variables individually, but you would then have to sum the results by hand to get the total number of households that choose that particular crop. Assuming that we only want to use Frequencies and/or Crosstabs, we will use **Analyze / Tables / Multiple Response Sets** to define the set and then use **Custom Tables** to produce the tables. Open the household level file.

1. **File / Open / Data...**
2. Select c-hh.sav
3. Paste, modify the directory specification to "sample", change the dataset name to "hh", block and run.

To create the table do the following:

1. **Analyze /Tables / Multiple Response Sets...**
2. Select **h35a**, **h35b**, **h35c** and move to Variables in Set:
3. Click on the radio button next to Categories in the Variable Coding section.
4. For the name of the variable - Name: **crops**
5. For the label - Label: **Crops grown principally to be sold**
6. Click on **Add** , and then **Paste**
7. Switch to the Syntax Editor and run the command **MRSETS**.

Now click on **Analyze / Tables / Custom Tables**

Note: If the variable level has not been defined for all variables, an information box will open prompting you to allow SPSS to scan the data and assign the proper level, choose Scan Data.

8. The set definition defined above can be found by scrolling to the end of the list of variables. Move **$crops**, into the Rows
9. Click on **Summary Statistics...** which is in the lower left corner of the dialog box.
10. Remove any statistics that appear in the Display: except Count.
11. Select Column Responses % from the Statistics box.
12. Select Table Responses % (Base: Count) from the Statistics box
    For an explanation of these statistics, click on the Help button and select Summary Statistics for Multiple Response Sets
13. Click on **Apply to Selection**
14. Click on "Categories and Totals". Place a ✓ (tick mark) in the box next to "Total". Click on **Apply.**
15. Click on **Paste**
16. Switch and run the command.

The Syntax editor should show this:

```
* Custom Tables.
CTABLES
  /VLABELS VARIABLES=$crops DISPLAY=DEFAULT
  /TABLE $crops [C][COUNT F40.0, COLPCT.RESPONSES PCT40.1,
TABLEPCT.RESPONSES.COUNT PCT40.1]
  /CATEGORIES VARIABLES=$crops ORDER=A KEY=VALUE
EMPTY=INCLUDE TOTAL=YES POSITION=AFTER.
```

Any variable that starts with a $ is a temporary variable. The output table is shown below.

The Column Responses % shows the percent of the total responses and adds to 100%. The Table Response % (Base:Count) gives the percent of households choosing the particular crop and can add to more than 100% since a household can choose up to three different crops.

		Count	Column Responses %	Table Response % (Base: Count)
Crops grown principally to be sold	COTTON	89	27.7%	42.6%
	PEANUTS	84	26.2%	40.2%
	SESAME	3	0.9%	1.4%
	SUNFLOWER	1	0.3%	0.5%
	RICE	85	26.5%	40.7%
	MAIZE, BEANS	41	12.8%	19.6%
	BANANA	4	1.2%	1.9%
	MANIOC	7	2.2%	3.3%
	SUGAR CANE	4	1.2%	1.9%
	TOBACCO	1	0.3%	0.5%
	SWEET POTATO	1	0.3%	0.5%
	CASHEW NUT	1	0.3%	0.5%
	Total	209	100.0%	153.6%

Using Sets and CTABLES to produce a crosstab table

We can look at the same information by district to determine which crops are most important in each district.

1. **Analyze /Tables / Custom tables.**
2. Move **$crops** into Rows: (it may already be there).
3. Move **district** to Columns.
4. With **$crops** highlighted, <right-click> and select **Summary Statistics**.
5. Remove any statistics in the Display box except the Count.
6. From the Statistics box, scroll down and select Column Responses % (Base: Count) and move it to the Display box.
7. From the Statistics box, scroll down and select Column Count %

(Base: Responses) and move it to the Display box.

8. Click on **Apply to Selection**
9. With **$crops** highlighted, <right-click> and select **Categories and Totals**.
10. Place a ✓ (tick mark) in the box next to "Total". Click on **Apply.**
11. Click on **Paste**
12. Switch and run the command.

The Syntax editor should show:

```
* Custom Tables.
CTABLES
  /VLABELS VARIABLES=$crops district DISPLAY=DEFAULT
  /TABLE $crops [C][COUNT F40.0, COLPCT.RESPONSES.COUNT PCT40.1,
COLPCT.COUNT.RESPONSES PCT40.1] BY
    district [C]
  /CATEGORIES VARIABLES=$crops ORDER=A KEY=VALUE
EMPTY=INCLUDE TOTAL=YES POSITION=AFTER
  /CATEGORIES VARIABLES=district ORDER=A KEY=VALUE
EMPTY=INCLUDE.
```

		DISTRICT								
		MONAPO			RIBAUE			ANGOCHE		
		Count	Column Response % (Base: Count)	Column Count % (Base: Responses)	Count	Column Response % (Base: Count)	Column Count % (Base: Responses)	Count	Column Response % (Base: Count)	Column Count % (Base: Responses)
Crops grown principally to be sold	COTTON	62	82.7%	68.1%	24	54.5%	45.3%	3	3.3%	1.7%
	PEANUTS	13	17.3%	14.3%	2	4.5%	3.8%	69	76.7%	39.0%
	SESAME	0	.0%	.0%	0	.0%	.0%	3	3.3%	1.7%
	SUNFLOWER	0	.0%	.0%	1	2.3%	1.9%	0	.0%	.0%
	RICE	5	6.7%	5.5%	2	4.5%	3.8%	78	86.7%	44.1%
	MAIZE, BEANS	7	9.3%	7.7%	18	40.9%	34.0%	16	17.8%	9.0%
	BANANA	0	.0%	.0%	2	4.5%	3.8%	2	2.2%	1.1%
	MANIOC	0	.0%	.0%	2	4.5%	3.8%	5	5.6%	2.8%
	SUGAR CANE	3	4.0%	3.3%	1	2.3%	1.9%	0	.0%	.0%
	TOBACCO	0	.0%	.0%	1	2.3%	1.9%	0	.0%	.0%
	SWEET POTATO	0	.0%	.0%	0	.0%	.0%	1	1.1%	.6%
	CASHEW NUT	1	1.3%	1.1%	0	.0%	.0%	0	.0%	.0%
	Total	75	121.3%	82.4%	44	120.5%	83.0%	90	196.7%	50.8%

Refer to the Help documentation to obtain an explanation of the difference between the Column Response % (Base: Count) and Column Count % (Base: Responses).

In the Output window you can <right-click> on the pivot table, select **Copy**, open your word processor, <right-click> and **paste.** The table pastes into the document as a table that you can edit. If the table is too wide for the page, <right-click> on the table and select **Autofit – Autofit to contents**. The table will be resized to fit the width of the document.

Looking at the table you can see that in Monapo cotton is the main cash crop; in Ribaue, it is cotton and maize/beans; in Angoche peanuts and rice are the main cash crops.

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

1. Make the Output window active.
2. Use **Save** from the **File** menu to automatically save under the name Session3.spv.

Let's now look at the *multiple dichotomy* type of mutiple response. Question 64 on the survey asks:

"Over the last five years, have you increased the quantities marketed of the following crops:"

The respondent is to answer yes if they have increased the quantities and no if they have not. There are 8 crops listed. There are 8 variables defined with values of 1 = yes and 2 = no. We will define a multiple response set for these variables and then use the **Custom Tables** command to analyze the set of variables.

Multiple Response sets (MDSETS command)

Multiple dichotomy variables

1. **Analyze / Tables / Multiple Response Sets...**
2. Select variables **h64a through h64h** and move them into the Variables in Set box.
3. The radio button next to Dichotomies is the default in the Variable Coding As section. In the Counted value, type **1**.
   which means we are looking at the values that are equal to yes
4. For the name of the variable - Name: **incprod**
5. For the label Label: **Increased quantities marketed in last 5 years**
6. Click on **Add** , and then **Paste**

You should see the command MRSETS with a subcommand "MDGROUP".

```
* Define Multiple Response Sets.
    MRSETS
      /MDGROUP NAME=$incprod LABEL='Increased quantities
         marketed in last 5 years'
        CATEGORYLABELS=VARLABELS VARIABLES=h64a h64b
         h64c h64d h64e h64f h64g h64h VALUE=1
      /DISPLAY NAME=[$incprod].
```

Go into the Custom Tables dialog box to now use this set variable.

7. Move **$incprod** to Rows:
8. Move **district** to Columns:
9. Highlight **$incprod ,** <right-click> and select Summary Statistics.
10. Select Row Responses % and Column Responses % and move them into the Display box. Click on **Apply to Selection** .
11. Add a total for both variables. Repeat instructions for each variable: (highlight variable, select **Categories and Totals**. Place a ✓ (tick mark) in the box next to "Total". Click on **Apply**.)
12. Click on **Paste**
13. Switch and run the command.

You can see how important a specific crop is to each district (row percent) and which crops are more important by district for increased sales (column percent).

If you want to produce another table using **Custom Tables**, the set variables are available to use. You could also save the data file to a new name. The Set definition will be saved. When you open that new file, the set definition will be available to use again.

Save the output file. We will use it in the next Section on graphs.

Hiding variables

If you have many variables in the data files and want to work with a subset of the variables without seeing the others, you can define a variable set that includes only those variables you want to use. This option is available under the Utilities menu.

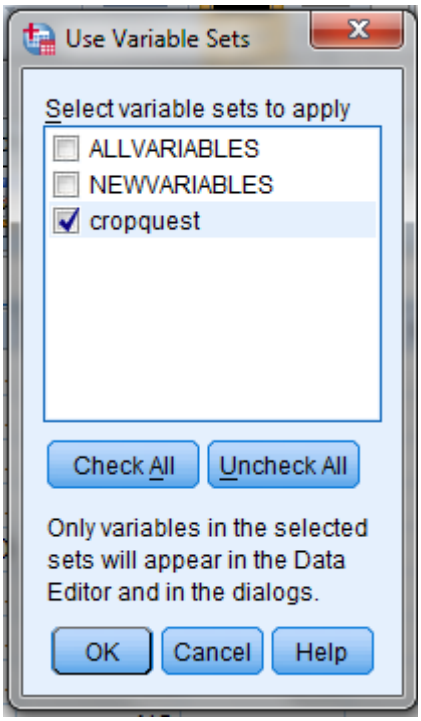
1. **Utilities / Define Variable Sets….**
2. Fill in the Set Name: that will identify the set of variables. Type: **cropquest**.
3. In the lower part of the dialog box, pick the variables you want to show in the data editor and in any dialog box and move them to the box labeled Variables in Set

```
Pick the key variables that identify the
 household and any other variables you want to
 see:
district vil hh h29, h31, h34, h36, h39, h52
```

4. At the top of the dialog box click on the button labeled **Add Set**
5. Click on **Close**

To see just the variables that you selected for the cropquest set go back into the Utilities menu:

6. Now we need to tell SPSS to use this set. Click on **Utilities / Use Variable Sets….**
7. Remove the tick marks on ALLVARIABLES and NEWVARIABLES and put a tick on "cropquest".
8. Click on **OK**



Now you will only see the variables that added to the set called cropquest. To see all the variables again go back to **Utilities / Show All Variable**

Creating Dummy Variables

Converting categorical variables to indicator variables

A dummy variable is a special case of a categorical variable. An dummy variable has two groups only, whereas other categorical variables can have more than two groups. Usually the values in indicator variables are 0 and 1 or no/yes.

SPSSS can convert categorical variables to dummy variables. Suppose that you want to do a regression analysis and control for effects of the different geographic regions. We have a variable called district which has 3 categories. We want to create dummy variables for the three districts. If the district is 1, the dummy variable will have a value of 1 if the case is in the district or a 0 if the case is not in the district.

The command to create these variables is under the Transform option in the menus.

1. **Transform / Create Dummy Variables**
2. The dialog box opens. Select the variable you want to use to create dummy variables and move it to the box on the right. Select **district**.
3. The next box down Dummy Variable Labels will use the value labels be default. That is fine. Value order will be ascending by default.
4. On the right side under Main Effect Dummy Variables place a tick mark ✓ by Create main-effect dummies.
5. For Root Names (One Per Selected Variable) type **dist**

If you are creating dummy variables for more than one variable

separate the root names with a space.

6. Click on **Paste.**

The syntax is

```
SPSSINC CREATE DUMMIES VARIABLE=district
ROOTNAME1=dist
/OPTIONS ORDER=A USEVALUELABELS=YES USEML=YES OMITFIRST=NO.
```

Run this command. In the Data Editor three variables will be created: dist_1, dist_2 and dist_3. The variable labels will be district=MONOPO, district=RIBAUE and district=ANGOCHE.

You can run a CROSSTABS of district by dist_1 dist_2 and dist_3 to look at the results.

```
CROSSTABS
  /TABLES=district BY dist_1 dist_2 dist_3
  /FORMAT=AVALUE TABLES
  /CELLS=COUNT
  /COUNT ROUND CELL.
```

Converting continuous variables to indicator variables

Suppose we want to create a new variable that indicates whether a person is 18 years old or older. You could have computed a new variable and assigned it a value of 1 if ca3 > =18. Then you would need a second step to recode the system missing to 0. There is another way to create this variable.

We will use the file c_q1a.dta. Open the file and then create a new variable using the **compute** command following the steps below:

1. Click on **File** then **Open .**
2. Select c_q1a.dta and **Paste**. Switch to the syntax editor and run the command, first deleting the reference to the folder.
3. Select **Compute Variable** from the **Transform** menu
4. Type the name of the new variable in the **Target Variable** box: **age18p**
5. Click on **Type & Label** to add a label for **age18p**. Type the label "**Age 18 and older**", then select **Continue**
6. In the **Numeric expression** box, type in
   ca3>=18
7. Click on the **Paste** button, switch to the syntax editor, and run the commands
8. Run a **frequency** to look at the results.

The command would be:

```
compute age18p = ca3>=18.
```

SECTION 4 - Graphs, tables, publications and presentations, Survey estimation to account for design effects ,

Copy table output to a word processor

The objective of this section is to give you the tools necessary to prepare reports, i.e. to learn how to move SPSS results into other applications. We will focus on a chart or table for examples. The methods used in this example would be quite similar for other SPSS output. In earlier sections instructions were also given on how to copy output to word processing applications.

The method is simple: once the SPSS results such as a chart or a table are produced, it can be printed or incorporated into reports prepared using word processors or publishing programs. It is always good to save the SPSS output file, in case you need to copy the output again. Incorporating tables and charts from SPSS can be done using a copy and paste procedure. We will use the output file from Section 3. Find the following table in the section3.spv file:

1. Click on **File / Open / Output...**
2. Select Session3.SPV in the folder where you saved your output from the sample session (*.SPV extension)
3. Locate one of the tables. Click once on the table to select it.
4. Select **Edit / Copy** from the menu. You could also use the mouse to <right-click> and select **Copy**.
5. Next open your word processor software.
6. With your word processor open, <right-click> and select **Paste**.
   The first option of the three paste options keeps the source formatting where the paragraph keeps 16 pt line spacing and right justifies the numbers and centers the column labels. (SPSS's format).
   The second option of the three paste options merges the formatting where it removes the formatting line spacing, changing it to single space and left justifies the text.
   The third paste option keeps the text only and does not put the output into a table format. Tabs separate the columns.
7. Back in the SPSS Output window, select the table and choose **Edit / Copy** from the menu. You could also use the mouse to <right-click> and select **Copy** .
8. Switch back to the word processes and <right-click> and select **Paste** .

You can do some editing to the table in the SPSS output window. If you want to remove some columns from the table, it is best to do it in SPSS. SPSS will keep the proper formatting for the column headings. Trying to delete columns in Word may not give you the results you are looking for and will require further editing and formatting in Word.

If you want to insert words or change words, that is best done in Word rather than SPSS. However, you can do it SPSS as well. If you want to change the format of the numbers, do the changes in SPSS using the formatting within the CTABLES command. It is much easier, either to specify the formatting in the CTABLES command or you can use can edit the table in the Output window to change the format.

You can also copy tables and paste the output from SPSS into a spreadsheet program, using the Copy/Paste procedure.

Other options available for the Output window are to export the *.SPV into an Excel, Powerpoint and pdf format. Click on **File / Export** and choose the type of output from the drop down box under Export Format. Quoting from the help menu in SPSS

• *Excel. Pivot table rows, columns, and cells are exported as Excel rows, columns, and cells, with all formatting attributes intact--for example, cell borders, font styles, and background colors. Text output is exported with all font attributes intact. Each line in the text output is a row in the Excel file, with the entire contents of the line contained in a single cell. Charts, tree diagrams, and model views are included in PNG format. Output can be exported as Excel 97-2004* or *Excel 2007 and higher*.

• *PowerPoint file (*.ppt). Pivot tables are exported as Word tables and are embedded on separate slides in the PowerPoint file, with one slide for each pivot table. All formatting attributes of the pivot table are retained--for example, cell borders, font styles, and background colors. Charts, tree diagrams, and model views are exported in TIFF format. Text output is not included.*

• *Portable Document Format (*.pdf). All output is exported as it appears in Print Preview, with all formatting attributes intact*. "

• *HTML and Web Reports are also available as formats for export.*

Copy graphics to a word processor

The process is basically the same for Graphics. As an example, we will look at the distribution of cashew tree ownership across households in the Mozambique data, using a histogram.

SPSS has enhanced the graphics menu. The option is called **Chart Builder.** The old legacy charts and interactive charts are still available. Syntax already developed using the legacy commands for charts should still work. Read the **Help / Contents / Chart Builder** to learn more about using Chart Builder.

Open the household level file, C-HH.SAV, which contains the tree ownership variable.

1. **File / Open / Data...**
2. Select C-HH.sav
3. Paste, edit the directory structure to replace it with "sample", select and run.

GRAPH command

Using the Legacy system, create the Histogram chart using the variable H57 (number of trees owned):

4. Select **Graphs / Legacy Dialogs / Histogram...**.
5. Find *H57* (Number of cashew trees) in the variable list and move it

into the Variables box.

6. Paste, select and run.

The command pasted is:

```
GRAPH
  /HISTOGRAM=h57 .
```

You should get a histogram chart like this:

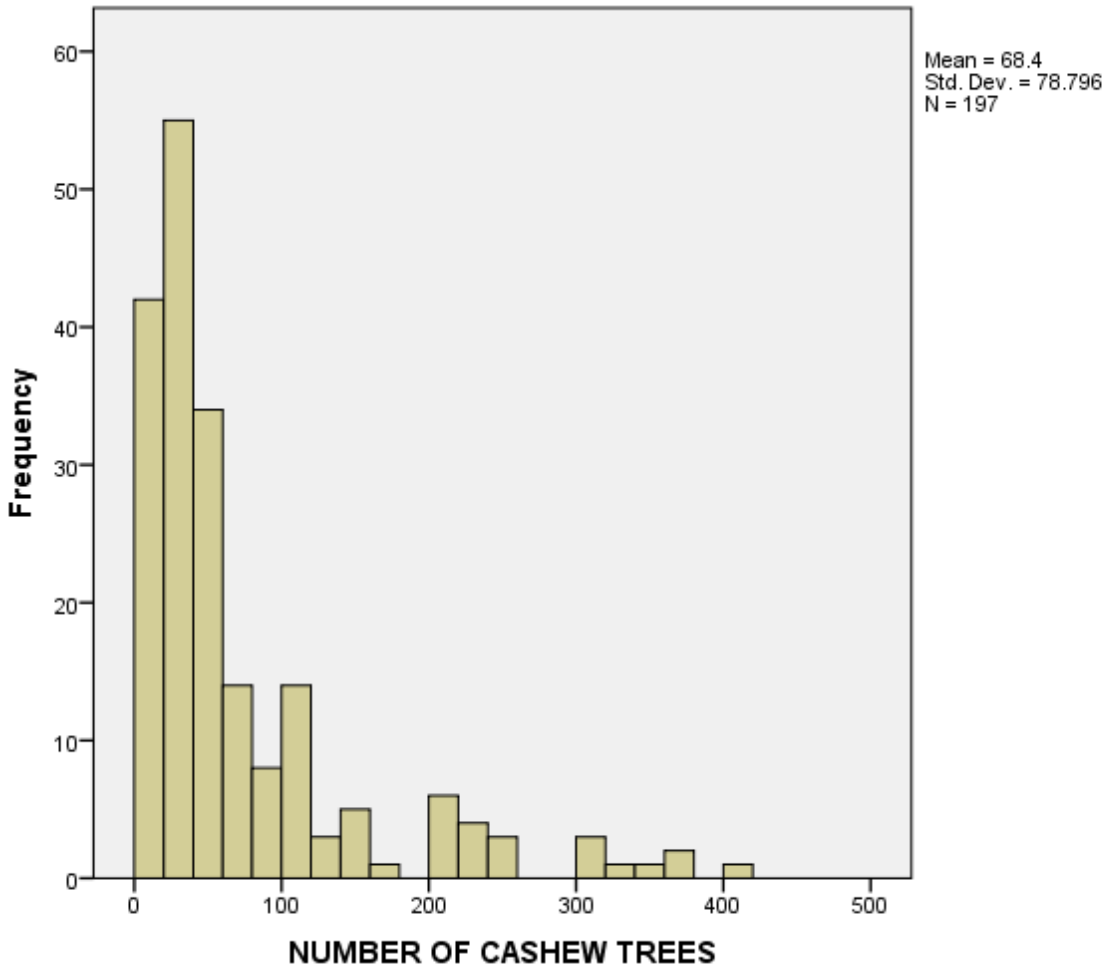


CHART BUILDER: GGRAPH command

Let's now use the new Chart Builder.

1. Select **Graphs / Chart Builder...**

   A dialog box opens giving you the opportunity to define your variables correctly with respect to measurement level and to add value labels. If you do not need to do any of these things, click on OK.

2. Select the type of chart you want to build from the **Gallery** tab. Click on **Histogram** and select the first example, labeled **Simple Histogram.** Move your mouse over the graph type, click and drag it up into the box above and drop it.
3. Another dialog box opens on the right with a Title bar labeled -

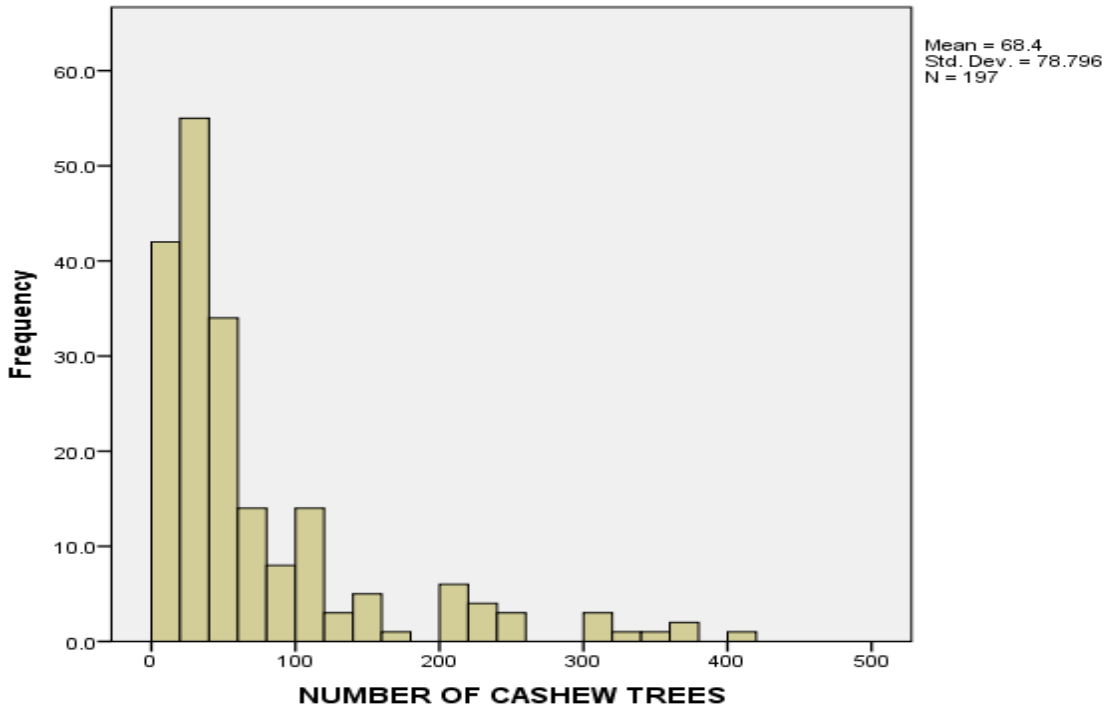
Element Properties. Selecting any of the items in the Edit Properties of: box displays different information below for you to choose how you want the items to display. You can explore this dialog box. We will use the defaults.

4. Find *H57* (Number of cashew trees) in the variable list and drag it to the box labeled X-axis?. Note that the Y-axis changes from Y-axis? to Histogram.
5. **Paste**, block starting at the GGRAPH through END GPL and run.

The command is:

```
* Chart Builder.
GGRAPH
  /GRAPHDATASET NAME="graphdataset" VARIABLES=h57
MISSING=LISTWISE REPORTMISSING=NO
  /GRAPHSPEC SOURCE=INLINE.
BEGIN GPL
  SOURCE: s=userSource(id("graphdataset"))
  DATA: h57=col(source(s), name("h57"))
  GUIDE: axis(dim(1), label("NUMBER OF CASHEW TREES"))
  GUIDE: axis(dim(2), label("Frequency"))
  ELEMENT: interval(position(summary.count(bin.rect(h57))),
shape.interior(shape.square))
END GPL.
```

The output looks very similar. Legacy GRAPH commands will eventually be phased out. It would be best to spend time learning the GGRAPH commands.



We can copy any one of the above charts to put it into a word processor.

1. Go to the Output window and click once on one of the graphs to select it. You can <right click> and choose Copy, or you can first edit the graph by clicking twice to open the Chart Editor. You can add Titles or change fonts or other actions, if you want.

2. From within the Chart Editor, select **Edit / Copy Chart**.

3. Close the Chart Editor and open your word processor.

4. In the word processor, click on **Edit / Paste Special**. Select **Device Independent Bitmap** and click on **OK** . Or you can just <right click> and choose **Paste**

You will not be able to edit this graph once it is in the word processor, other than the size, placement, wrapping of text and other basic aspects available within the word processor.

Exercise 4.1

Select another table from your Session3.SPV file. Repeat the steps above to copy the chart into your word processing document. Try making changes to the chart from within SPSS and then copy the new chart to your word processor document.

GRAPH BOARD TEMPLATE CHOOSER GGRAPH command

An option has been added to allow the user to visualize the graph based on the definition of the Variable Type.

1. Select **Graphs / Graph board template choose...**
   A dialog box opens showing the variable on the left. There are 4 tabs in this dialog box, "Basic","Detailed", Titles" and "Options".
2. Select a variable from the list on the left. (To select multiple variables use <ctrl click>.) SPSS proposes graphs and shows a visualization of what those graphs would look like.
3. To place the code in the syntax file, you can click on Paste. Switch to the syntax edit to select the code and run.
   Note that the GGRAPH command is used.
4. You could also just click on **OK** to see the results and return to select another chart to look at the results. Once you determine the chart you want, you can use the GGRAPH command to build the chart.

Click on the Help button in the dialog box to learn more about this option. You can also explore the command further by clicking on the Help and then Contents from the main menu.

Survey Estimation - Accounting for Design Effects

SPSS has provided an option to allow researchers to work with data that have not been selected purely on a simple random basis from a population but have a specific design for the sampling frame. "The Complex Samples option allows you to select a sample according to a complex design and incorporate the design specifications into the data analysis, thus ensuring that your results are valid." To read in detail about this option, go to the Help menus. The Help goes into detail about the properties of a complex sample, using complex samples procedures, and provides a list of references for further reading. The commands associated with the complex sample option start with "CS".

Survey data generally have three importance characteristics:

1. The weights applied to survey data are sampling weights - also called probability weights
2. The sample is clustered
3. Stratification is used in selecting the sample

If data meets any one of the above characteristics, the complex sample commands can be used for analysis. Briefly, sampling weights are used in analysis to give estimators that are approximately unbiased for whatever is being estimated for the whole population, i.e. one observation represents many elements in the population from which the sample is drawn.

Clustering by districts or villages is used in almost all survey sampling rather than selecting an independent sample. Further sub-sampling may occur within a district or a village as well. Units at the first level of sampling are called the "*primary sampling unit*" or "PSU" or cluster.

To summarize, weights are used to obtain the correct point estimates. Clustering and stratification are used to get the correct standard errors.

We will use a data set from Zambia from the Post harvest survey of the 2001/2002 agricultural season where the area planted for specific types of crops is tested.

1. Click on **File** then **Open**
2. Select Zambia_PHS0102_crop_area.sav and click on **Paste.**
3. Switch to the syntax editor and delete the reference to the folder, inserting the FILE HANDLE variable; change the dataset name to "phs" and then run the commands.

In Zambia for surveys conducted in the 1990s and early 2000, a stratified random sampling method was used. This method divided the districts into census supervisory areas (CSA). Within the CSA, Standard Enumerator Areas (SEA) were defined. The primary sampling unit (PSU) for this sample is the SEA. To identify each SEA as being unique the three variables - district, CSA and SEA, must be combined into one variable. District has 3 numbers, CSA has 3 numbers and SEA has 2 numbers. To create a new variable with these variables one must multiply the district variable by 100,000, add CSA multiplied by 100, and add SEA. The SPSS command is:

```
COMPUTE cluster1 = dist*100000 + csa*100 + sea.
EXECUTE.
```

Clusters may further be sampled in groups which are called strata. The Zambia example uses province - district as the strata. Strata are considered to be statistically independent and can be analyzed as such.

A weight has already been calculated for each household. The variable which contains this value is called **hhwgt**.

For the cluster we have computed the variable cluster1. We can use dist for the strata variable since it already contains the province value as part of the district code.

To be able to use the survey commands, we must first define the stratified random sampling method that was used to account for weighting, clustering and stratification. We will use the **svyset** command to specify the method.

1. Click on **Analyze** then **Complex samples**
2. Then click on **Prepare for analysis…**
3. The default option on the Analysis Preparation Wizard is Create a plan file. We can save the plan to a file. Click on **Browse**. Type the name PHS_sample_plan and click on **Save**.
4. The next screen is where we define the Stage 1: Design Variables. Move **dist** to the **Strata:** box
5. Move **cluster1 to** the **Clusters:** box
6. Move **hhwgt to** the **Sample Weight** box. Click on **Next**.
7. In this screen, there are three choices to describe the method selected for the sample. The default is WR (sampling with replacement) which is correct for this sample.
8. Click on **Next**
9. This screen summarizes the plan. Click on **Next**.
10. The last screen gives you the choice to Paste the syntax generated by the Wizard into a syntax window. Select is option. Click on **Finish**.
11. Switch to the syntax editor, edit the folder reference to replace it with the FILE HANDLE variable, then run the command.

The SPSS command is:

```
CSPLAN ANALYSIS
  /PLAN FILE='sample\PHS_sample_plan.csaplan'
  /PLANVARS ANALYSISWEIGHT=hhwgt
  /SRSESTIMATOR TYPE=WOR
  /PRINT PLAN
  /DESIGN STRATA=dist CLUSTER=cluster1
  /ESTIMATOR TYPE=WR.
```

After running the command we see a summary of the command in the Output window:

Summary

			Stage 1
Design Variables	Stratification	1	dist district
	Cluster	1	cluster1
Analysis Information	Estimator Assumption		Sampling with replacement

Plan File:

C:\Users\beaverm\Documents\sample\PHS_sample_plan.csaplan

Weight Variable: hhwgt weighting factor

SRS Estimator: Sampling without replacement

Once the survey design has been specified and saved to a file, it can be retrieved and used at any time for analysis.

We can use the **csdescriptives** command to look at the total estimates.

1. Click on **Analyze / Complex samples**
2. Then click on **Descriptives**
3. You need to specify the name of the plan. Click on the **Browse** button to select the file, which has an extension name of .csaplan. Then click on **Continue** .
4. In the **Variables** box select **maisea ricea milleta sunfa** and move them into the **Measures:** box.
5. Click on the **Statistics** button in the upper right corner of the dialog box.
6. In the **Summaries** place a tick in the box next to **Sum**
7. In the **Statistics** place a tick in the box next to **Confidence Interval**
8. Click on the **Continue** button, click on **Paste**.
9. In the syntax editor, replace the folder reference with the FILE HANDLE variable; then run the command.

The output is displayed below.

Univariate Statistics

		Estimate	Standard Error	95% Confidence Interval	
				Lower	Upper
Mean	maizea	.8041	.02651	.7519	.8562
	ricea	.0179	.00286	.0123	.0236
	milleta	.0765	.00842	.0599	.0931
	sunfa	.0301	.00420	.0219	.0384
Sum	maizea	649230.91	25105.88576	599840.35	698621.47
	ricea	14472.95	2360.00941	9830.13	19115.77
	milleta	61770.91	7346.12463	47318.95	76222.87
	sunfa	24319.15	3418.85814	17593.26	31045.04

Let's run the same analysis with only the weight specified to see the difference.

1. Click on **Analyze** then **Complex samples**
2. Then click on **Prepare for analysis…**
3. The default option on the Analysis Preparation Wizard is Create a plan file. We can save the plan to a file. Click on **Browse**. Type the name PHS_sample_plan_revised and click on **Save**.

4. The next screen is where we define the Stage 1. Move **hhwgt to** the **Sample Weight** box. Leave the other boxes blank. Click on **Next**.
5. In this screen, there are three choices to describe the method selected for the sample. The default is WR (sampling with replacement) which is correct for this sample.
6. Click on **Next**
7. This screen summarizes the plan. Click on **Next**.
8. The last screen gives you the choice to Paste the syntax generated by the Wizard into a syntax window. Select is option. Click on **Finish**.
9. Switch to the syntax editor, edit the folder reference to replace it with the FILE HANDLE variable, then run the command.
10 Click on **Analyze / Complex samples**
11. Then click on **Descriptives**
12. You need to specify the name of the plan. Click on the **Browse** button to select the file, which has an extension name of .csaplan. Then click on **Continue** .
13. In the **Variables** box select **maisea ricea milleta sunfa** and move them into the **Measures:** box.
14. Click on the **Statistics** button in the upper right corner of the dialog box.
15. In the **Summaries** place a tick in the box next to **Sum**
16. In the **Statistics** place a tick in the box next to **Confidence Interval**
17. Click on the **Continue** button, click on **Paste**.
18. In the syntax editor, replace the folder reference with the FILE HANDLE variable; then run the command.

Note, we have gotten the same point estimate as the design-based estimate, but the standard errors are much smaller and the 95% confidence interval is also different. The second table does not account for the sampling design but assumes the sample is random rather than stratified random.

Univariate Statistics

				95% Confidence Interval	
		Estimate	Standard Error	Lower	Upper
Mean	maizea	.8041	.01634	.7721	.8361
	ricea	.0179	.00164	.0147	.0211
	milleta	.0765	.00482	.0671	.0859
	sunfa	.0301	.00236	.0255	.0347
Sum	maizea	649230.91	14013.13427	621760.63	676701.18
	ricea	14472.95	1327.55916	11870.50	17075.39
	milleta	61770.91	3942.68357	54041.97	69499.84
	sunfa	24319.15	1907.91909	20579.01	28059.29

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 – Additional information on some commands
Annex 2 – Survey Instrument
Annex 3 – References

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. A new option in SPSS 22 allows you to copy the filtered cases to a new dataset window, where you can work with that dataset of cases.

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 Data 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 **Delete unselected cases** in the Output section of this dialog 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 sysmis.

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 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 MEMBER 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**	
3=cotton 5=peanuts 6=rice 21=cashew nut 30=beans 31=manteiga bean 41=dry manioc 47=corn 44=sorghum	Unit 1=sack 100 2=sack 50 3=kilo 4=liter 5=can 20	Qty	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	Qty	(enter the month)	Unit 1=sack 100 2=sack 50 3=kilo 4=liter 5=can 20	Qty	(enter the month or "all year", if appropriate)	Unit 1=sack 100 2=sack 50 3=kilo 4=liter 5=can 20 other	Qty
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
	3=cotton 5=peanuts 6=rice 21=cashew nut 30=beans 31=manteiga bean 41=dry manioc 47=corn 44=sorghum	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
VEN	**PROD**	**V2A**	**V2B**							**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 the file C-Q5.sav. Only variables VEN, V2A, V2B, V9A and V9B were kept for this exercise. The PROD variable replaces the V1 variable.