



INTERNATIONAL FOOD  
POLICY RESEARCH INSTITUTE  
*sustainable solutions for ending hunger and poverty*  
Supported by the CGIAR



# Training Manual

## Using SPSS for Poverty and Nutrition Policy Analysis

Compiled by

Wahidur Quabili  
IFPRI

Nigeria Strategy Support Program (NSSP)

October 2008

**IFPRI-ABUJA**  
International Food Policy Research Institute  
c/o International Center for Soil Fertility and Agriculture  
Development  
Plot 1413 Ogbagi Street  
Cadastral Zone 2, Garki, Abuja  
Nigeria  
Tel: +234 (9) 413-5960  
Fax: +234 (9) 4130873  
E-mail: [ifpri-nigeria@cgiar.org](mailto:ifpri-nigeria@cgiar.org)  
[www.ifpri.org](http://www.ifpri.org)

**IFPRI HEADQUARTERS**  
International Food Policy Research Institute  
2033 K Street NW  
Washington, DC 20006-1002 USA  
Tel. +1-202-862-5600  
Fax +1-202-467-4439  
E-mail [ifpri@cgiar.org](mailto:ifpri@cgiar.org)  
[www.ifpri.org](http://www.ifpri.org)

## **ABOUT NSSP/APSF**

The Agricultural Policy Support Facility (APSF) is an initiative to strengthen evidence-based policymaking in Nigeria in the areas of rural and agricultural development. Facilitated **by the Nigeria Strategy Support Program of the International Food Policy Research Institute (IFPRI)** in collaboration with the Federal Ministry of Agriculture and Water Resources and funded by the Canadian International Development Agency, APSF supports the implementation of Nigeria's national development plans by strengthening agricultural-sector policies and strategies through:

Enhanced knowledge, information, data, and tools for the analysis, design, and implementation of pro-poor, gender-sensitive, and environmentally sustainable agricultural and rural development policies and strategies in Nigeria;

Strengthened capacity for government agencies, research institutions, and other stakeholders to carry out and use applied research that directly informs agricultural and rural policies and strategies; and

Improved communication linkages and consultations between policymakers, policy analysts, and policy beneficiaries on agricultural and rural development policy issues.

**Training Manual**  
**Using SPSS for Poverty and Nutrition Policy Analysis**

Compiled by

Wahidur Quabili  
IFPRI

## Table of Contents

Background .....	1
Introduction to Survey Data Files .....	2
General Information about Using SPSS .....	3
Documentation and Transformation.....	4
Commands and Syntax for using SPSS for Poverty and Nutrition Policy Analysis.....	4
Opening and Saving Files .....	5
Documenting Data Sets .....	6
Processing a Raw Data File.....	9
Generating New Variables .....	11
Aggregating and Adding Files .....	16
Merging Files.....	20
Creating Composite Variables (e.g., Household Expenditures) .....	25
Developing a Tabulation Plan and Descriptive Analysis .....	27
Descriptive Analysis .....	28
T-test .....	30
Correlation Analysis .....	32
Crosstabs .....	33
Generating Plots and Graphs .....	35
Creating a Master File for Doing Regression Analysis .....	39
Poverty Analysis.....	40
Nutrition Analysis.....	43
Regression Analysis.....	44
References .....	49

## **Background**

Poverty and malnutrition in Nigeria is widespread and severe, and therefore, continues to be central to the development discourse. Sixty-four percent of rural Nigerians has a consumption level below the basic-needs poverty line, while forty-three percent of rural children, who are under five years of age, is stunted in growth. Thus, many rural Nigerians live under conditions of extreme vulnerability, both economically and physically, with limited abilities to cope with shocks that arise and safeguard their current level of well-being.

Government efforts at reducing poverty and malnutrition have culminated into numerous programmes. Prominent among them are:

- National Poverty Eradication Programme (NAPEP)
- National Economic Empowerment and Development Strategy (NEEDS)
- Millennium Development Goals (MDG)

While these programmes are desirable, their outcomes have been mixed. If Nigeria is to achieve the first Millennium Development Goal (MDG) of reductions in poverty, hunger, and malnutrition, and the fourth MDG of reducing child mortality, a clear understanding of the defining characteristics of the poor and malnourished, dimensions of poverty and malnutrition, and the determinants of such poverty and malnutrition are required for more sustainable policy actions. In addition, programmes that aim to reduce poverty and malnutrition can only be more effective if quantitative and qualitative poverty and nutrition analysis and statistics form the basis of their design.

During the planning of the Agricultural Policy Support Facility (APSF)<sup>1</sup> program in Nigeria, consultations with the Federal Ministry of Agriculture and Water Resources (FMAWR) and other stakeholders identified the need to build and strengthen capacities of ministries, departments and agencies in poverty and nutrition policy research and analysis. This need is premised on the fact that competent planners and policy analysis experts are essential for sound and effective policy analysis and planning.

To assist in ensuring that the poverty and nutrition policies and programs are supported by appropriate analysis, the APSF in collaboration with the National Bureau of Statistics organized a 4-day training course on Poverty and Nutrition Policy Analysis from October 6-9, 2008, at the National Bureau of Statistics. The content of the course was based on a needs-assessment carried out among the potential participants. The training hosted 24 participants from the Federal Ministry of Agriculture and Water Resources, Federal Ministry of Health, National Planning Commission, National Bureau of Statistics, research institutes and universities.

### **Objectives of the training course**

- Strengthen existing capacity across sectoral ministries and agencies in data analysis, specifically poverty and nutrition policy analysis
- Expand the knowledge base for developing agricultural, rural development , poverty, and nutrition-related policies and programs
- Enable the development of advanced computing skills
- Reinforce international best practice in poverty and nutrition analysis.

This training module is to introduce SPSS to the participants and to show them how to use SPSS to generate descriptive statistics and tables from survey data. In particular, the course aims to train the participants in the following methods:

- Working with survey data files
- File management such as opening, modifying, and saving files
- Documenting data files with variable labels and value labels
- Cleaning of data within and across the files

---

<sup>1</sup> APSF is a joint program implemented by the International Food Policy Research Institute in collaboration with the Federal Ministry of Agriculture and Water Resources, Nigeria

Data management such as merging, appending, and aggregating files  
 Generating new variables using various arithmetic and statistical functions and operands  
 Creating tables to describe the distribution of continuous and discrete variables  
 Creating tables to describe the relationships between two or more variables  
 Performing basic nutrition and poverty analysis  
 Using regression analysis to study the impact of various variables on a dependent variable.

## Introduction to Survey Data Files

### List of useful terms

The following are some key concepts that will be used throughout this training module. Most of you will be familiar with them, but it is worth reviewing the terms for those that may not know all of them.

**Records** (or cases or observations) are individual observations such as persons, farm plots, households, villages, or states. They are usually considered to be the .rows of the data file. For example, data set A (below) has five records and data set B has six records.

**Variables** are the characteristics, location, or dimensions of each record. They are considered the .columns of the data file.

In data set A (Table 1), there are four variables: the household identification number (HHID), the region where the household lives (Region), the size of the household (HHSIZE), and the distance from the house to the nearest source of water (DISTWAT).

In data set B (Table 2), there are six variables: the region (REGION), province (PROVINCE), household (HHNO), plot number (PLOT), whether or not it is irrigated (IRRIG), and the size of the plot (AREA).

The **level** of the dataset describes what each record represents. For example,

In data set A (Table 1), each record is a different household, so it is a *household-level data set*.

In data set B (Table 2), each record is a farm plot, it is a *plot-level data set*. Note that more than one record has the same household identification number.

**Table 1 Data set A**

HHID	REGION	HHSIZE	DISTWAT
101	1	4	15
102	1	5	30
103	1	6	25
104	2	3	50
105	3	7	20

**Table 2 Data set B**

REGION	PROVINCE	HHNO	PLOT	IRRIG	AREA
1	1	101	1	2	35.5
1	1	101	2	1	42
1	1	101	3	1	16.5
1	2	103	2	1	25
2	1	106	1	1	14
2	1	106	2	2	23
3	5	107	1	1	32.5
3	5	107	2	1	19

**Key variables** are the variables that are needed to identify a record in the data. In data set A, the variable HHID is enough to uniquely identify the record so HHID is the only key variable. In data set B, the key variables are REGION, PROVINCE, HHNO, and PLOT because all four variables are needed to uniquely identify the record. The first three records have the same region, province, and household, so these three variables are not enough to uniquely identify a record.

**Discrete variables** (or categorical variables) are variables that have only a limited number of different values. Examples include region, sex, income category, type of roof, and education level. Yes/no variables such as whether a household has electricity are also discrete variables.

**Binary variables** (or dummy variables) are a type of discrete variable that only takes two values. They may represent yes/no, male/female, have/don't have, or other variables with only two values.

**Continuous variables** are variables whose values are not limited. Examples include income, farm size, number of trees, rice consumption, coffee production, and distance to the road. Unlike discrete variables, continuous variables are usually expressed in some units such as Nigerian Naira, kilometers, hectares, or kilograms and may take fractional values

**Variable labels** are longer names associated with each variable to explain them in tables and graphs. For example, the variable label for HHSIZE might be .Household size and the label for DISTWAT could be Distance to water (km). Whenever possible, variable labels should include the unit (e.g., km).

**Value labels** are longer names attached to each value of a variable. For example, if the variable REG has eight values, each value is associated with a name. REG=1 could be Northeast Region.

## **General Information about Using SPSS**

### **Getting data into SPSS**

SPSS can read a variety of data formats, including data files saved in SPSS format, SAS datasets, database tables from many database sources, Excel and other spreadsheets, and text data files with both simple and complex structures.

**Table 3 SPSS data formats**

<b>Command</b>	<b>Descriptions</b>
SPSS data files	
Get	Reads SPSS-format data files
Import	Reads SPSS portable data files created with the Export
Add	Files combines multiple data files by adding cases
Match Files	Combines multiple data file by adding variables
Update	Replaces values in a master file with updated values
Data Files Created by Other Applications	
Get Translate	Reads spreadsheet and dBASE files
Get Data	Reads Excel files, text data files, and database tables
Database Tables	
Get Data	Reads Excel files, text data files, and database tables
Get Capture	Reads database tables
SAS and Stata Data Files	
Get SAS	Reads SAS dataset and SAS transport files
Get Stata	Reads Stata data files
Text Data Files	
Data List	Reads test data files
Get Data	Reads Excel files, text data files
Begin data –End data	Used with Data List to read inline text data

## **Documentation and Transformation**

### **Data definitions**

An SPSS data file can contain more than simply data values. The **SPSS dictionary** can contain a variety of metadata attributes, including measurement level, display format, descriptive variable, and value labels, and special codes for missing values.

**Table 4 SPSS metadata attributes**

<b>Command</b>	<b>Descriptions</b>
Variable Labels	Assigns descriptive labels to variables
Value Labels	Assigns descriptive labels to data values
Add Value Labels	Assigns descriptive labels to data values
Missing Values	Specifies values to be treated as missing
Rename	Changes variable names
Formats	Changes variable print and write formats
Variable Alignment	Specifies the alignment of data values in the Data Editor
Variable Width	Specifies the column width of data values in the Data Editor

### **Data Transformations**

You can perform data transformations ranging from simple tasks, such as collapsing categories for analysis, to more advanced tasks, such as creating new variables based on complex equations and conditional statements.

**Table 5 SPSS data transformations**

<b>Command</b>	<b>Descriptions</b>
Autorecode	Recodes the values of string and numeric variables to consecutive integers
Compute	Creates new numeric variables or modifies the values of existing string or numeric variables
Count	Counts occurrences of the same value across a list of variables
Create	Produces new series as a function of existing series
Date	Generates date identification variables
Numeric	Declares new numeric variables that can be referred to before they are assigned values
Rank	Produces new variables containing ranks, normal scores and Savage and related scores for numeric variables
Recode	Changes, rearranges, or consolidates the values of an existing variable
String	Declares new string variables
Temporary	Signals the beginning of temporary transformations that are in effect only for the next procedure

## **Commands and Syntax for using SPSS for Poverty and Nutrition Policy Analysis**

The following sections will provide knowledge and information on the commands and syntax that are used in SPSS for running and computing information in SPSS related to poverty and nutrition policy analysis. The initial sections focus on the basics of SPSS, while the later sections look specifically at computations and analysis that can undertaken through SPSS that support poverty and nutrition analysis. Within each

section, the step-by-step instruction for using the screen interface is given followed by the syntax for undertaken the same task.

## **Opening and Saving Files**

### **Opening an existing SPSS data file**

Click on **File**> Click **Open**> Click **Data**> Click **File Name**> Click **Open** or **Paste Syntax**

Syntax for opening the file "Roster.sav":

```
GET
FILE='U:\Nigeria\NLSS\Roster.sav'.
```

### **Opening a non-SPSS spreadsheet in SPSS data editor**

Non-SPSS files can be opened directly in SPSS data editor by

Click on **File**> Click **Open**> Click **Data**>Select the type of file> Click **File Name**> Click **Open** or **Paste Syntax**

Example: Syntax for opening a Stata file "roster.dta":

```
GET
STATA FILE='U:\Nigeria\NLSS\roster.dta'.
```

[Non-SPSS spread sheets can be converted into SPSS files by using DBMS/COPY.

DBMS/COPY is a software for transferring data between software programs such as SPSS, Stata, SAS etc.]

### **Saving a SPSS data file**

Click on **File**> Click **Save As** and give a file name> Click **Save** or **Paste Syntax**

Example: Syntax for saving a SPSS data file "Roster\_1.sav"

```
SAVE OUTFILE='U:\Nigeria\NLSS\Roster_1.sav'
/COMPRESSED.
```

### **Saving SPSS data file in a non-SPSS format**

Click on **File**> Click **Save as**>Select **Save as Type** and give a file name> Click **Save** or **Paste Syntax**

Example: Syntax for saving roster\_1.sav file in stata format

```
SAVE TRANSLATE OUTFILE='U:\Nigeria\NLSS\roster_1.dta'
/TYPE=STATA /VERSION=8
/EDITION=SE /MAP /REPLACE .
```

### **Opening and saving SPSS syntax and output files**

To open a new syntax files:

Click on **File**> Click **New**> Click **Syntax**

To open an existing syntax file:

Click on **File**> Click **Open**> Click **Syntax** and browse folder for file name

## Relational Operators

A relation is a logical expression that compares two values using a relational operator. In the command

IF (X EQ 0) Y=1

the variable X and 0 are expressions that yield the values to be compared by the EQ relational operator. The following are the relational operators:

<i>Symbol</i>	<i>Definitions</i>
EQ or =	Equal to
NE or ~= or ¬ = or <>	Not equal to
LT or <	Less than
LE or <=	Less than or equal to
GT or >	Greater than
GE or >=	Greater than or equal to

## Documenting Data Sets

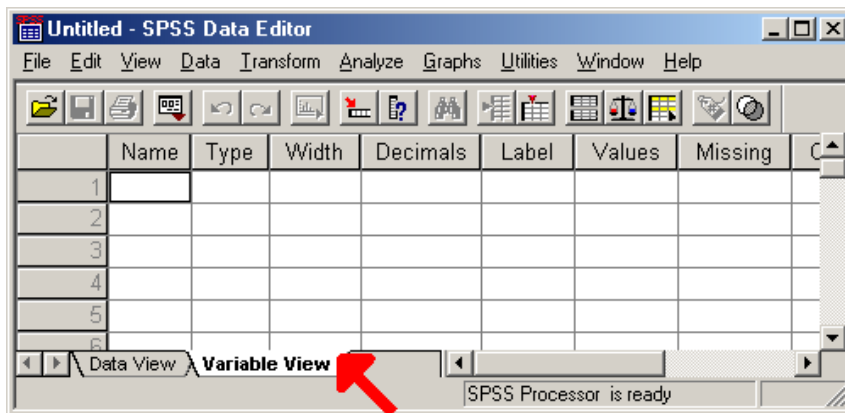
It's important to document data sets to make sure the data is used correctly. In versions 10 and later of SPSS, there are two tabs at the bottom of the SPSS screen

**Data View**

**Variable View**

Use the **Data View** tab to enter or view your data. Use the **Variable View** tab to add documentation to your data.

### Variable View



From the **Variable View** tab, you can

- Tell SPSS how to display the data in the SPSS data editor window (how many decimal places shown, how dates are displayed, and how wide the columns are);
- Provide SPSS with informational labels that will appear in your output window (labels for the variable itself and, if needed, labels for category levels); and
- Use the dialog box to specify any codes that represent missing data.

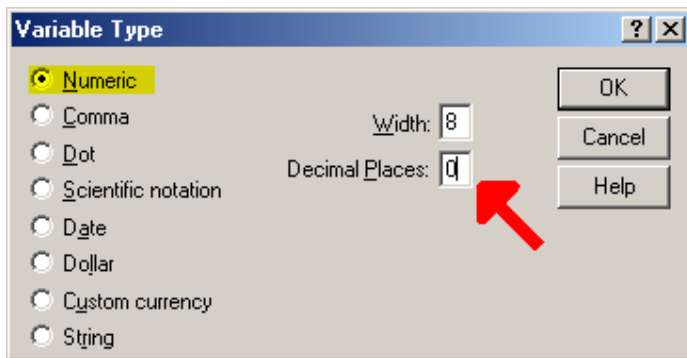
After you provide a variable name, take a look at the other columns in the **Variable View** tab. Of special interest are the following:

- Type – variable type
- Width – refers to the width of the column
- Decimals – refers to the number of decimal places
- Labels – description of the variable
- Values – description of the values
- Missing – if any value declared as missing
- Columns – column width
- Align – alignment of the variable (left, right, center)
- Measure – scale, ordinal or nominal

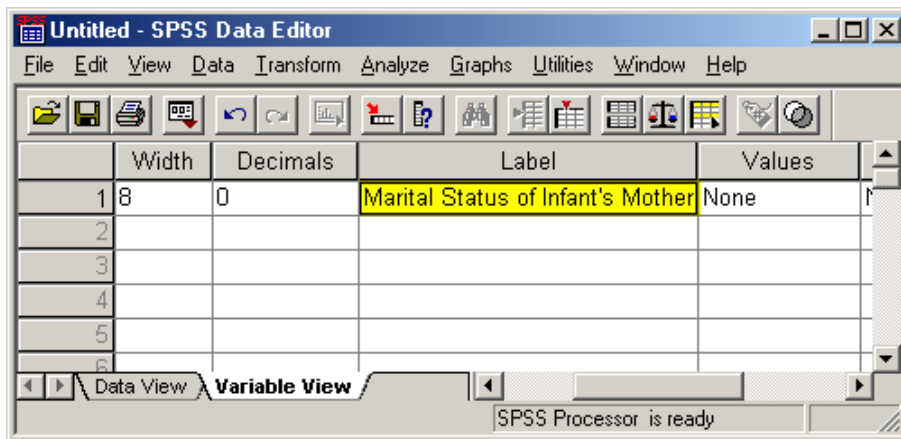
In the **Define Variables** dialog box, you can see the buttons: **Type**, **Labels**, **Missing Values**, and **Column Format**.

Example: Working with variable labels

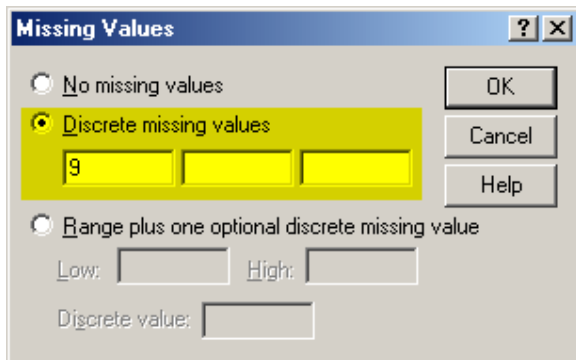
Marit\_st is a **categorical variable** with five codes (1=single, 2=married, 3=divorced, 4=widowed, 9=unknown). Marit\_st is the variable name for marital status of infant mothers



This variable uses numeric codes, so the **Numeric** option is selected. With no values larger than 9, its **Width** field can be changed slightly, but anything much smaller than 8 makes it difficult to see the **Variable Name** and the **Values** labels later. The number codes here do not require any decimals, so the **Decimal Places** field is changed from 2 to 0.



To add a variable label, click on the box under **Label**, type in "Marital Status of infant's Mother". If information about "Marital Status of infant's Mother" is missing, it will be designated with the value 9. See the box below.



To add more documentation, select **Data>Define Variable** from the SPSS menu or double click on the column header.

To write notes about a data file, click **Add Document**.

**Add Document** saves a block of text of any length in an SPSS-format data file. The documentation can be displayed with the **Display Document** command.

**Drop Documents** can be used to delete those documents from the working file.

### Writing variable and value labels (discussed above)

Variable labels, value labels, etc., can be written using SPSS syntax.

#### Example: Writing variable labels using SPSS syntax

Variable label `marit_st` "marital status of infant's mother".

Value label `marit_st` 1 "single" 2 "married" 3 "divorced" 4 "widowed" 9 "unknown".

Format `marit_st` F(8.0).

Missing value `marit_st` (9)

### Sorting a data file

To sort data in a matrix, click on **Data** and then click on **Sort Cases**. Click on the variable that you want sorted as the primary variable (i.e., HHID) and click on the black arrow that points to the right. This will move the variable title into the **Sort by** column. Choose **Sort Order** either ascending or descending. Click **OK** and SPSS will sort your data and automatically rearrange all the other variables.

Paste the syntax:

```
SORT CASES BY HHID (A).
```

### How to use SPSS help

SPSS help gives a tutorial, context-sensitive help topics (command syntax reference), pop-up definitions, statistical coach, and case studies.

#### Example: Working with help

For command syntax reference, Click on **Help> Click Command Syntax Reference>** Scroll down to command you are searching for (it's in alphabetical order)

### Practice

Open a ASSET.dbf file in SPSS  
Save a ASSET.sav file in stata format

Open ROSTER.sav file  
Display data document  
Extend AGE variable by 3 places after decimal  
Convert sex (numeric) variable in to string variable  
Add information  
Sort the file by per capita household expenditure

## Processing a Raw Data File

### Keeping, dropping, and renaming variables

DROP and KEEP are used to save a subset of variables. DROP specifies the variables that are not to be saved in the new data file; KEEP specifies the variables that are to be saved in the new data file.

Variables that are not named on KEEP are dropped.

RENAME changes the names of variables as they are copied into the new SPSS-format data file.

The syntax for keeping a variable is:

```
GET
```

```
FILE='U:\Nigeria\NLSS\Roster.sav' /KEEP HHID MID sex age relation marital_st.
```

The syntax for dropping a variable is:

```
GET
```

```
FILE='U:\Nigeria\NLSS\Roster.sav' /DROP occup o:
```

```
SAVE OUTFILE='U:\Nigeria\NLSS\Roster_n.sav' /RENAME (marital_st = marital).
```

### Display

DISPLAY exhibits information from the dictionary of the active dataset. It allows users to see a description of the variables. The information can be sorted and limited to selected variables.

Example:

```
GET FILE="/NLSS/HHEXP.sav".
```

```
DISPLAY DOCUMENTS.
```

```
DISPLAY DICTIONARY
```

Each DISPLAY command specifies only one keyword. The first command DISPLAY DOCUMENT requests documentary text and the second command DISPLAY DICTIONARY requests complete dictionary information for the HHEXP.SAV file.

DISPLAY SORTED DICTIONARY displays complete dictionary information for variables in the active dataset, sorted alphabetically by variable name.

## List Variables

LIST displays case values for variables in the active dataset. This allows users to see the data.

### **LIST with no subcommands:**

The command LIST by itself displays the values for all variables in the active dataset.

### **Controlling listed cases with CASES Subcommand:**

LIST VARIABLES=V1 V2 /CASES=FROM 10 TO 100 BY 2.

This command produces a list of every second case for variables V1 and V2, starting with case 10 and stopping at case 100.

## Temporary

This command reads a subset of the dataset. The results of the request are stored temporarily, but at the next command the program will return to reading the entire dataset. With TEMPORARY, you can perform separate analyses for subgroups in the data and then repeat the analysis for the file as a whole.

TEMPORARY can also be used to transform data for one analysis but not for other subsequent analyses.

### Example:

TEMPORARY.

Select if (pcexp>10000).

List HHID PID PCEXP

HHID PID and PCEXP will be listed only for those who have PCEXP greater than 10000

### **Practice**

Open ROSTER.SAV

Write variable label for OCCUP

Variable label OCCUP 'Primary occupation'.

Add value labels for OCCUP (66=unemployed, 77=retired)

Syntax:

Add value label OCCUP 66 "Unemployed" 77 "Retired"

Drop Flag and Order variables from the file

Rename PCEXP into PERCAPX

Syntax:

Save outfile 'ROSTER\_N.sav' /drop flag order /rename  
(PCEXP=PERCAPX).

Frequency of occupation of household head if per capita expenditure  
<1000 (use TEMPORARY command)

Syntax:

TEMPORARY.

Select if PCEXP<1000.

FREQUENCY OCCUP.

## Generating New Variables

### Compute

COMPUTE creates new numeric variables or modifies the values of existing string or numeric variables. The variable named on the left of the equals sign is the *target variable*. The variables, constants, and functions on the right side of the equals sign form an *assignment expression*.

Example: Multiply PCEXP by HHSIZE:

Click on **Transform**>Click **Compute**>Write the name of target variable (HHEXP) and the variable you want to work with (PCEXP)> blue arrow (it will paste the variable in the numeric expression box> numeric expression (\* to multiply)>2<sup>nd</sup> variable (HHSIZE)>blue arrow>OK or paste Syntax:

```
COMPUTE HHEXP = PCEXP * HHSIZE.
```

### Common functions

#### **Statistical functions:**

**SUM(arg list):** Sum of the non missing values across the argument list  
**MEAN(arg list):** Mean of the non missing values across the argument list  
**SD(arg list):** Standard deviation of the non missing values across the argument list  
**VARIANCE(arg list):** Variance of the non missing values across the argument list  
**CFVAR(arg list):** Coefficient of variation of the non missing values across the argument list (the coefficient of variation is the standard deviation divided by the mean)  
**MIN(arg list):** Minimum non missing value across the argument list  
**MAX(arg list):** Maximum non missing value across the argument list

#### **Arithmetic functions:**

**RND(arg):** Round the absolute value to an integer and reaffix the sign. RND(SCALE) is -5 when SCALE equals -4.7.  
**TRUNC(arg):** Truncate to an integer. TRUNC(SCALE) is -4 when SCALE equals -4.7.  
**MOD(arg,arg):** Remainder (modulo) of the first argument divided by the second. When YEAR equals 1983, MOD(YEAR,100) is 83.  
**SQRT(arg):** Square root. SQRT(SIBS) is 1.41 when SIBS equals 2.  
**EXP(arg):** Exponential. e is raised to the power of the argument. EXP(VARA) is 7.39 when VARA equals 2.  
**LG10(arg):** Base 10 logarithm. LG10(VARB) is 0.48 when VARB equals 3  
**LN(arg):** Natural or Napierian logarithm (base e). LN(VARC) is 2.30 when VARC equals 10.  
**SIN(arg):** Sine. The argument must be specified in radians. SIN(VARD) is 0.84 when VARD equals 1.  
**COS(arg):** Cosine. The argument must be specified in radians. COS(VARE) is 0.54 when VARE equals 1.

#### **Using statistical functions**

```
COMPUTE NEWSAL = SUM(SALARY,RAISE).
```

```
COMPUTE MINVAL = MIN(V1,V2,V3,V4).
```

```
COMPUTE MEANVAL = MEAN(V1,V2,V3,V4).
```

#### **Using arithmetic functions:**

```
COMPUTE NEWVAR=RND((V1/V2)*100).
```

```
COMPUTE INCOME=TRUNC(INCOME).
```

```
COMPUTE MINSQRT=SQRT(MIN(V1,V2,V3,V4)).
```

### **Creating a string variable:**

A string variable is an alphabetical variable. When you have data that is not numerical, such as the name of a state, you would create a string variable.

Click on **Transform**>Click **Compute**>Click **Go to Function Group** and select **String**>Click **Type & Label**>Select **Type String**>Write the width of variable>Write the label of the variable>Click **Continue**>Click **OK** or **Paste**.

```
STRING newvar (A22).
```

```
COMPUTE newvar='done'.
```

```
VARIABLE LABELS newvar 'it is done or not'.
```

```
EXECUTE.This syntax will create a new string variable "newvar" in the data set.
```

### **Generating age in months from date of birth and date of interview:**

```
Compute age_mo=( YRMODA (int_yr, int_mo, int_da) – YRMODA (birth_yr, birth_mo, birth_da))/(365/12).
```

Yr = year, mo = month, da=day, int-interview date, birth = birth day

### **Generating new variables by using IF command**

Example:

```
If (HAZ<-2) stunt=1. Where HAZ is Height for age Z-score
```

```
If (HAZ>=-2) stunt=0.
```

```
Variable label stunt 'Stunted growth (HAZ<-2)'.
```

```
Value label stunt 1 'Stunt' 0 'Not stunt'.
```

#### **Practice**

Generate square of household size

Syntax:

```
Compute hhsize_sq= HHSIZE*2.
```

Round per capita expenditure

Syntax:

```
COMPUTE PCEXP_r = RND (PCEXP).
```

Generate log (natural) of per capita expenditure

Syntax:

```
COMPUTE ln_PCEXP= LN (PCEXP).
```

Open ANTHROPOMETRY.SAV file and generate age of the children using date of birth and date of interview (use appropriate variable names)

Syntax:

```
Compute age_mo=( YRMODA (int_yr, int_mo, int_da) – YRMODA (birth_yr, birth_mo, birth_da))/(365/12).
```

Variable label age\_mo "age of child in months".

## Count

This is a way to count occurrences of a single value:

```
COUNT TARGET=V1 V2 V3 (2).
```

The value of TARGET for each case will be either 0, 1, 2, or 3, depending on the number of times the value 2 occurs across the three variables for each case.

### Example: counting occurrences of a range of values and system-missing values:

Say we have 10 variables, Q1 to Q10. If we want to count in how many variables values are 0 or less and in how many variables values are missing, we can write the syntax as below

```
COUNT QLOW=Q1 TO Q10 (LO THRU 0)
/QSYSMIS=Q1 TO Q10 (SYSMIS).
```

Two variables -- QLOW and QSYSMIS -- would be created in the data set. QLOW is the number of variables with values 0 or less and QSYSMIS is the number of variables with missing data.

## Recode

RECODE changes, rearranges, or consolidates the values of an existing variable. RECODE can be executed on a value-by-value basis or for a range of values. Where it can be used, RECODE is much more efficient than the series of IF commands that produce the same transformation.

### ***Recode into same variable***

Click on **Transform>Recode> Select Into same variable>Select Variable to be recoded>Select Old and new value>Write old and new value>Click Add>Click Continue>Click OK or Paste Syntax**

Syntax example:

```
RECODE educ (99=0)
```

### ***Recode into a new variable***

Click on **Transform>Recode> Click into different variable>Select input variable>Give name and write label of output variable> Click Old and New Value>Write old and new value>Click Add> Click Continue> Click Change> Click OK or Paste Syntax**

Syntax example:

For example, if we have a variable for years of schooling "educ" where 99 means never attended school, we can recode this 99 into 0 years of schooling by writing the following syntax:

```
RECODE educ (99=0) (else=copy) INTO educ1 .
VARIABLE LABELS educ1 'education (new)'.
EXECUTE .
```

It will recode 99 into 0. The other values will remain the same.

Syntax example 2:

```
RECODE V1 TO V3 (0=1) (1=0) (2,3=-1) (9=9) (ELSE=SYSMIS).
```

In this example, for variables V1 to V3 0 will be recoded to 1, 1 will be recoded to 0, 2 and 3 will be recoded to -1, 9 will remain as 9 and other values/missings will be recoded to missings (sysmis)

### Syntax example 3:

For recoding a string variable, the syntax would be as follows:

```
RECODE STRNGVAR ('A','B','C'='A')('D','E','F'='B')(ELSE=' ').
```

In this example, we are recoding strings A, B and C as “A,” strings D, E, and F as “B” and all other strings as blank.

### ***Recoding a string variable into a numeric target***

```
RECODE SEX ('M'=1) ('F'=2) INTO NSEX.
```

### **Categorizing variables**

#### Example: Per capita expenditure quintile

Click on **Transform**> Click **Rank Cases**> Select variable to be ranked (e.g., pccexp)> Select ‘a’ by variable (e.g., by state)> Select **Rank Type** (e.g., NTILES=5)>Click **Continue**> Click **OK** or **Paste Syntax**

#### Syntax examples:

\* Without using any location.

```
RANK
```

```
VARIABLES=pccexp (A) /RANK /NTILES (5)
```

This will create a new variable Npccexp as the overall expenditure quintile.

\* By location

```
RANK
```

```
VARIABLES= pccexp (A) BY state /RANK /NTILES (5)
```

This will create a new variable Npccexp as the expenditure quintile by location (here it is by state).

\* We can give the name of the new categorized variable (say quintile).

```
RANK
```

```
VARIABLES= pccexp (A) /RANK /NTILES (5) into quintile
```

```
Variable label quintile “Per capita expenditure quintile.”
```

### **Cleaning and checking the files (ensuring consistency within the file)**

#### Example:

Check duplicate cases in a household roster file.

Click on **Data**> Click **Identify duplicate cases**> Select variables to be checked (HHID and PID)> Click **OK** or **Paste Syntax**

The lag function can also be used to identify duplicate cases.

#### Example:

Identify duplicate member IDs (MID) in each household (HHID).

Duplicate cases can be identified, or “flagged” by writing the following syntax:

```
If ((HHID=lag(HHID) and PID=lag(PID)) flag=1.
```

Temporary.

Select if flag=1.

List HHID PID AGE SEX

This will give us a list of all duplicate cases as described above.

Marital status and age can also be checked.

Example:

It is very unlikely that an individual will be married before 15 years of age. Those cases can be identified by writing following syntax:

Temporary.

Select if (age<15 & marit\_st>1).

List HHID PID AGE SEX MARIT\_ST

### Practice

Open ROSTER.SAV and recode s1q2 (variable for sex of person)  
1=male and 2=female into a new variable

Syntax:

```
STRING SEXN (A8).
```

```
RECODE s1q2 (1='Male') (2='Female') INTO SEXN.
```

```
VARIABLE LABELS SEXN 'sex of the person'.
```

Check if age of the household head is less than 18 years

Syntax:

```
Compute age = sum (s1q5y, s1q5m/12).
```

```
Variable label age 'age in years'.
```

Temporary.

```
Select if (s1q3=1 & age<18) .
```

```
List HHID PID s1q2 s1q3 age
```

Check if there are more than one household heads in any household

Syntax:

```
Select if s1q3=1.
```

```
If (HHID=lag (HHID) & s1q3=lag(s1q3)) flag=1.
```

Temporary.

```
Select if flag=1.
```

```
List HHID PID s1q2 s1q3.
```

Open HHEXP.SAV and generate per capita expenditure quartile by state

Syntax:

```
RANK
```

```
VARIABLES= pcexp (A) BY state /RANK /NTILES (5) INTO
```

quartile

## Aggregating and Adding Files

### Aggregating files

The AGGREGATE command aggregates groups of cases in the active dataset into single cases and creates a new aggregated file, or creates new variables in the active dataset that contain aggregated data. The values of one or more variables in the active dataset define the case groups. These variables are called **break variables**. A set of cases with identical values for each break variable is called a **break group**. Aggregate functions are applied to **source variables** in the active dataset to create new aggregated variables that have one value for each break group.

### Options

You can create new variables in the active dataset that contain aggregated data, replace the active dataset with aggregated results, or create a new SPSS-format data file that contains the aggregated results.

You can copy documentary text from the original file into the aggregated file using the DOCUMENT subcommand. (By default, documentary text is dropped.)

You can create aggregated variables using any of 19 aggregate functions. The functions SUM, MEAN, and SD can aggregate only numeric variables. All other functions can use both numeric and string variables.

You can specify variable labels for the aggregated variables. Variables created with the functions MAX, MIN, FIRST, and LAST assume the formats and value labels of their source variables. All other variables assume the default formats described under Aggregate Functions.

Click on **Data> Click aggregate> Select break variables** (gender and marital)> blue arrow (it will paste the variable in the break variable box)>select the variables to be aggregated (age)> blue arrow (it will paste the variable in the summa>**select aggregate function** (mean, median, sum etc.)>**write variable name** and label>go to **save box and select** one of the three options (e.g. create a new data set containing only the aggregated variables)> give a name of the new file> Click **OK** or **Paste Syntax**

### Syntax example:

AGGREGATE

    /OUTFILE='c:\temp\temp.sav'

    /BREAK=gender marital

    /age\_mean=MEAN(age)

    /age\_median=MEDIAN(age).

AGGREGATE creates a new SPSS-format data file, temp.sav, that contains two break variables (gender and marital) and all of the new aggregate variables.

BREAK specifies gender and marital as the break variables. In the aggregated file, cases are sorted in ascending order of gender and in ascending order of marital within gender. The active dataset remains unsorted.

Three aggregated variables are created: age\_mean contains the mean age for each group defined by the two break variables; age\_median contains the median age; and income\_median contains the median income.

**Aggregate functions:**

**SUM(varlist)** Sum across cases.

**MEAN(varlist)** Mean across cases

**MEDIAN(varlist)** Median across cases

**SD(varlist)** Standard deviation across cases

**MAX(varlist)** Maximum value across cases

**MIN(varlist)** Minimum value across cases

**PGT(varlist,value)** Percentage of cases greater than the specified value

**PLT(varlist,value)** Percentage of cases less than the specified value

**PIN(varlist,value1,value2)** Percentage of cases between value1 and value2, inclusive

**POUT(varlist,value1,value2)** Percentage of cases not between value1 and value2

**FGT(varlist,value)** Fraction of cases greater than the specified value

**FLT(varlist,value)** Fraction of cases less than the specified value

**FIN(varlist,value1,value2)** Fraction of cases between value1 and value2, inclusive

**FOUT(varlist,value1,value2)** Fraction of cases not between value1 and value2

**N(varlist)** Weighted number of cases in break group

**NU(varlist)** Unweighted number of cases in break group

**NMISS(varlist)** Weighted number of missing cases

**NUMISS(varlist)** Unweighted number of missing cases

**FIRST(varlist)** First nonmissing observed value in break group

**LAST(varlist)** Last nonmissing observed value in break group

**New aggregated data file (containing break and aggregated variables only)**

	gender	age_mean	groupSize	var	var	va
1	1	30.00	2			
2	2	36.67	3			
3						
4						
5						
6						

**Aggregate variables appended to active dataset**

	age	gender	age_mean	groupSize	var	va
1	25	1	30.00	2		
2	35	1	30.00	2		
3	20	2	36.67	3		
4	30	2	36.67	3		
5	60	2	36.67	3		
6						

**Practice**

Generate household size from ROSTER.SAV in a new aggregated file  
Syntax:

```
*select only current household members from the roster  
Select if s1q20=1.  
DATASET DECLARE HHSIZE.SAV.  
SORT CASES BY state sector ric hhno.  
AGGREGATE  
  /OUTFILE='U:\Nigeria\HHSIZE.SAV'  
  /PRESORTED  
  /BREAK=state sector ric hhno  
  /HHSIZE 'household size'=NU(pid).
```

Generate household size from ROSTER.SAV in the active file

Syntax:

```
    SORT CASES BY state sector ric hhno.  
    AGGREGATE  
      /OUTFILE=* MODE=ADDVARIABLES  
      /PRESORTED  
      /BREAK=state sector ric hhno  
      /HHSIZE 'household size'=NU(pid).
```

Generate number of population by state (from ROSTER.SAV)

Syntax:

```
    SORT CASES BY state.  
    AGGREGATE  
      /OUTFILE=* MODE=ADDVARIABLES  
      /PRESORTED  
      /BREAK=state  
      /statepop 'state population'=NU(pid).
```

### **Adding files (appending cases)**

The ADD FILES command combines cases from 2 up to 50 SPSS-format data files by concatenating cases. When cases are concatenated, all cases from one file are added to the end of all cases from another file.

#### Example: Adding two files

School1.sav

SCHOOLID	STUD	TEACHER
101	650	12
102	724	14

School2.sav

SCHOOLID	STUD	TEACHER
103	578	10
104	819	15

Syntax:

```
ADD FILES FILE="U:\Nigeria\school1.sav"  
/FILE="U:\Nigeria\school2.sav".  
EXECUTE.
```

The resulting file will be as below:

SCHOOLID	STUD	TEACHER
----------	------	---------

101	650	12
102	724	14
103	578	10
104	819	15

The ADD FILES command concatenates cases from the SPSS-format data files school1.sav and school2.sav. All cases from school1.sav precede all cases from school2.sav in the resulting file.

### Practice

Add roster\_part1 and roster\_part2 files  
Sort appended files by household and member ID and save the files as ROSTERN.SAV

Syntax:

```
GET
FILE='U:\Nigeria\ROSTER_part1.sav'.
ADD FILES /FILE=*
/FILE='U:\Nigeria\ROSTER_part2.sav'.
SORT CASES BY state(A) sector(A) ric(A) hhno(A) pid(A).
SAVE OUTFILE='U:\Nigeria\ROSTERN.sav'
/COMPRESSED.
```

## Merging Files

MATCH FILES combines variables from 2 up to 50 SPSS-format data files. MATCH FILES can make parallel or nonparallel matches between different files or perform table lookups. **Parallel matches** combine files sequentially by case (they are sometimes referred to as **sequential matches**). **Nonparallel matches** combine files according to the values of one or more key variables. In a table lookup, MATCH FILES looks up variables in one file and transfers those variables to a case file.

The files specified on MATCH FILES can be SPSS-format data files or open datasets in the current session. The combined file becomes the new active dataset. Statistical procedures following MATCH FILES use this combined file. You must use the SAVE or XSAVE commands if you want to save the combined file as an SPSS-format data file.

In general, MATCH FILES is used to combine files containing the same cases but different variables. To combine files containing the same variables but different cases, use ADD FILES. To update existing SPSS-format data files, use UPDATE.

### **Parallel matches (one-to-one merging)**

Two data files can be combined by merging them side by side, matching up observations based on an identifying variable (key variable).

#### Example:

For example, we have two data files A and B. File A contains household food and non-food expenditures and file B contains household size. These two data files can be merged together to have the expenditure

observations on the same line with the household size observation based on the key variable HHID. Both files must be sorted in same order.

File A

HHID	FOODX	NONFOODX
1001	2300	560
1002	3200	1500
1003	2500	1250
1004	1800	900

File B

HHID	HHSIZE
1001	4
1002	3
1003	5
1004	3

Syntax:

```
GET
FILE='U:\Nigeria\NLSS\File A.sav'.
MATCH FILES /FILE=*
/FILE='U:\Nigeria\ NLSS\File B.sav'
/BY HHID.
EXECUTE.
```

Note: For merging one-to-one cases, use the subcommand **File** to open the file to be merged with the active file. After merging the Files A and B, the data would appear as follows:

File C

HHID	FOODX	NONFOODX	HHSIZE
1001	2300	560	4
1002	3200	1500	3
1003	2500	1250	5
1004	1800	900	3

### Nonparallel matches (One-to-many merging)

**One-to-many merge** means one observation in one file may have multiple matching records in another file. For example, one can merge household expenditure (File C) with a household roster file (File D). File D has multiple observations for each household (HHID).

File C

HHID	FOODX	NONFOODX	HHSIZE
1001	2300	560	4
1002	3200	1500	3
1003	2500	1250	5

File D

HHID	MEMID	AGE	SEX
1001	1	45	M
1001	2	36	F
1001	3	9	F
1001	4	5	M
1002	1	31	M
1002	2	24	F
1002	3	2	F
1003	1	55	M
1003	2	45	F
1003	3	19	M
1003	4	14	F
1003	5	10	M

Syntax:

```
GET
FILE='U:\Nigeria\NLSS\File D.sav'.
MATCH FILES /FILE=*
/TABLE='U:\Nigeria\ NLSS\File C.sav'
/BY HHID.
EXECUTE.
```

Note: For merging one-to-many cases, use the subcommand **Table** to open the file to be merged with the active file.

After merging the Files D and C, the data would appear as follows:

HHID	MEMID	AGE	SEX	FOODX	NONFOODX	HHSIZE
1001	1	45	M	2300	560	4
1001	2	36	F	2300	560	4
1001	3	9	F	2300	560	4
1001	4	5	M	2300	560	4
1002	1	31	M	3200	1500	3
1002	2	24	F	3200	1500	3
1002	3	2	F	3200	1500	3
1003	1	55	M	2500	1250	5
1003	2	45	F	2500	1250	5
1003	3	19	M	2500	1250	5
1003	4	14	F	2500	1250	5
1003	5	10	M	2500	1250	5

Example:

HHID	FOOD1	FOOD2	FOOD3	FOOD4
1	1	4	6	9
2	2	4	1	8
3	7	9	2	6

```
VARSTOCASES /ID = id
/MAKE FCODE FROM FOOD1 FOOD2 FOOD3 FOOD4
/KEEP = HHID
/NULL = KEEP.
```

HHID	FOOD
1	1
1	4
1	6
1	9
2	2
2	4

2 1  
2 8  
3 7  
3 9  
3 2  
3 6

### Practice

Restructure RAWFOOD.SAV

GET

FILE='U:\Nigeria\RAWFOOD.sav'.

Click on **Data**> Click **Restructure**> Click **What do you want to do** (Select **Restructure selected variables into cases**)> Click **Next**> Click **How many variable groups you want to restructure** (Select one)> Click **Next**> Click **Use selected variables** (Select HHID)> Click **Name the target variable** (INGRED)>Select the variable group you want to restructure (INGRED1 to INGRED5)>Select **Fixed variables** (FOOD)> Click **Next**> Click **How many index variables do you want to create** (one)> Click **Next**> Click **Sequential numbers**>edit name and label of index variable (say the name of the variable is NINGRED and the label is 'ingredient number') Click **Finish**> Click **OK**

An index variable is a new variable that identifies the group of new cases that was created from the original cases. Here we will get an index variable NINGRED which would have the values of 1 to 5.

Syntax:

```
VARSTOCASES
```

```
/MAKE INGRED FROM INGRED1 INGRED2 INGRED3  
INGRED4 INGRED5
```

```
/INDEX=NINGRED "ingredient number" (5)
```

```
/KEEP=HHID FOOD
```

```
/NULL=KEEP.
```

## **Creating Composite Variables (e.g., Household Expenditures)**

### **Generating household expenditures**

We will illustrate how to create a composite variable by constructing a nonfood expenditures variable.

#### Example:

Generate monthly household nonfood expenditures from Block 10-A1 of the file, which refers to less frequently purchased items.

#### Syntax

```
GET
```

```
FILE='U:\Nigeria\Nigeria_NLSS_03_04\data\Sec10A1_NFHouseExp_Infrequent.sav'.
```

( We are using 12 months expenditure here to estimate monthly expenditure.)

The variable name of yearly nonfood expenditure is s10aq3. We are generating a variable for monthly non food expenditure:

```
compute nonfx_lf= s10aq3/12.
```

Here we are dividing the variable name for yearly nonfood expenditure by 12 to get the monthly nonfood expenditure.

Aggregating the monthly nonfood expenditure to household level:

```
SORT CASES BY state sector ric hhno.
```

```
AGGREGATE
```

```
/OUTFILE='U:\Nigeria\nonfoodx_less_freq.sav'
```

```
/PRESORTED
```

```
/BREAK=state sector ric hhno
```

```
/nonfx_lf 'hh non food expenditure per month (less frequently purchased)'=SUM(nonfx_lf).
```

### **Generating household size from the roster file**

(We will use a readymade household expenditure file (HHEXP.sav) for doing this practice.)

#### Example syntax:

```
GET FILE='U:\Nigeria\Nigeria_NLSS_03_04\data\Roster.sav'.
```

```
SORT CASES BY state sector ric hhno.
```

```
AGGREGATE
```

```
/OUTFILE='F:\Nigeria_trip\Training\HHSIZE.SAV.sav'
```

```
/PRESORTED
```

```
/BREAK=state sector ric hhno
```

```
/pid_nu=NU(pid).
```

### **Generating the total value of assets**

Use [SEC12B\_asset.sav].

Check unusual value of any assets (set a cut-off). Add the values of all three values in column 4 (For how much could you sell it now). Aggregate total value into household level.

#### Example:

```
GET FILE SEC12B_asset.sav
compute asst_val=sum (12bq4a, 12bq4b, 12bq4c)
*Aggregating total value of asset into household level
Aggregate outfile= 'd:\NLSS\asset_value.sav'
/break state sector ric hhno
/asst_val 'Total value of assets'=sum (asst_val).
```

### **Creating Z-score (Nutrition)**

A Z-score is computed by comparing each child's weight and height to the distribution of weight and height values of children referenced in the National Child Health Survey/World Health Organization (NCHS/WHO).

#### ***Calculating the weight for height z-score:***

$WHZ = \frac{\text{actual weight of child} - \text{median weight of reference child (same height, age and sex)}}{\text{absolute value of standard deviation (SD)}}$

Use ngkr4zrt.sav (children file of DHS, Nigeria 2003) to check unusual weight or height of children (i.e., set cut-offs).

In this case, the z-scores have already been calculated and are in the file. So the next step is to check the range of z-scores and set cut-offs.

### **Generating landholding size**

Use Sec9A1\_agri\_land.sav to check unusual size of any land parcel (set a cut-off).

Convert all land parcels into acres or hectares.

Select owned land only and aggregate land size into household level.

#### Examples:

\* Converting all land into acres

```
if (s9aq3=1) land_ac=s9aq4.
```

```
if (s9aq3=2) land_ac=s9aq4*2.471.
```

\*Selecting owned land only

```
Select if (s9aq1=1).
```

\*Aggregating land size into household level

```
Aggregate outfile= 'U:\Nigeria\landsize.sav'
```

```
/break state sector ric hhno
```

/land\_ac 'Landholding size in acre'=sum (land\_ac).

## Developing a Tabulation Plan and Descriptive Analysis

A tabulation plan outlines the presentation of information obtained from analyzing data. Here we are developing a tabulation plan in response to the following research question: What is the per capita household expenditure and other household demographics by region (urban or rural)?

For the purposes of this practice, use the following dummy (sample) Tables 6 and 7.

**Table 6 Monthly per capita expenditure, household size and percent of female headed household by region**

Descriptions	Statistics	Urban	Rural	All
Monthly per capita household expenditure (Naira)	Mean			
	Median			
	Std. Dev.			
	N			
Percent expenditure on Food	Percent			
Household size	Mean			
	Median			
	Std. Dev.			
	N			
Percent female headed household				

Syntax example:

MEANS TABLES=pcexp BY sector

/CELLS MEAN MEDIAN STDDEV COUNT.

**Table 7 Monthly per capita expenditure, household size and percent of female headed household by per capita expenditure quartile**

Descriptions	Statistics	Quartile 1 (lowest)	Quartile2	Quartile 3	Quartile 4 (highest)	All
Monthly per capita household expenditure (Naira)	Mean					
	Median					
	Std. Dev.					
	N					
Percent expenditure on Food	Percent					
Household size	Mean					
	Median					
	Std. Dev.					
	N					
Percent female headed household						

Note: Check 'ALL' columns of table 1 and 2 (values should be identical)

Syntax example:

rank pcexp /ntiles (4) into quartile.

variable label quartile 'Per capita expenditure quartile'.

```
MEANS TABLES=pcexp BY quartile
/CELLS MEAN MEDIAN STDDEV COUNT.
```

Example for creating female headed household variable from Roster.sav file:

```
*Selecting household heads only.
Select if (relation=1).
If (relation=1 & sex=2) femhh=1.
If (relation=1 & sex=2) femhh=0.
Variable label femhh 'female headed household'.
Value label femhh 1 'Female headed housheold' 0 'not female headed'.
Save outfile 'd:\NLSS\femhh.sav'.
```

## Descriptive Analysis

### Frequency

The FREQUENCY command produces tables showing frequency counts and percentages of the values of individual variables. You can also use the FREQUENCY command to obtain statistics tables for categorical variables and to obtain statistics tables and graphical displays for continuous variables.

Click on **Analyze**> descriptive statistics>Click **Frequency**> Select variable> Click **OK** or **Paste Syntax**

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

Output:

#### Statistics

sector

N	Valid	19158
	Missing	0

#### sector

	Frequency	Percent	Valid Percent	Cumulative Percent
Valid 1 Urban	4646	24.3	24.3	24.3
2 Rural	14512	75.7	75.7	100.0
Total	19158	100.0	100.0	

This frequency table shows distribution of urban and rural households

## Generating tables

Another descriptive analysis is to generate tables, for example, by expenditure quintile, by location, or by gender.

Click **Analyze**> Click **Tables**> Click **General Table**> Select the variables for rows and columns> Click **Edit statistics**> Click **Continue**> Click **OK** or **Paste Syntax**

### Syntax example:

The following syntax is asking the program to generate a table of percent of households in each expenditure quintile by urban/rural location:

```
rank pcexp /ntiles (5) into quintile.  
variable label quintile 'Per capita expenditure quintile'.  
value label quintile 1 'lowest' 5 'highest'.  
TABLES  
/FORMAT ZERO MISSING('.')  
/GBASE=CASES  
/FTOTAL= $t000001 "Total"  
/TABLE=quintile + $t000001 BY sector > (STATISTICS)  
/STATISTICS  
cpct( quintile( F5.1 ) 'percent':sector ) /TITLE 'Village by Regions'.
```

This creates a table showing the distribution of households in each expenditure quintile by urban and rural, below:

**distribution of expenditure quintile by urban and rural**

		1 Urban	2 Rural
		percent	percent
Per capita expenditure quintile	1 lowest	12.7	22.3
	2	14.5	21.8
	3	18.3	20.5
	4	24.6	18.5
	5 highest	29.9	16.8
Total		100.0	100.0

**Mean tables**

Mean tables give an average value for a given variable. These can be helpful in poverty and nutrition analysis by providing a general idea of the poverty and nutrition status of a population.

Click **Analyze**> Click **Compare Means**> Click **Means**> Select dependent (PCEXP) and independent (quintile) list of variables> Click **Option**> Select **Statistics**> Click **Continue**> Click **OK** or **Paste Syntax**.

MEANS

TABLES=pcexp BY quintile

/CELLS MEAN COUNT STDDEV MIN MAX .

**Table 8 Report pcexp per capita expenditure in local current prices**

quintile Percentile Group of pcexp	Mean	N	Std. Deviation	Minimum	Maximum
1	8198.1845	3831	2621.50	792.46	12146.49
2	15702.07	3832	2077.56	12146.83	19310.52
3	23827.489	3832	2745.35	19310.59	28870.61
4	36774.584	3832	5141.47	28871.18	46929.36
5	83799.347	3831	74389.85	46935.50	2286241.58
Total	33659.047	19158	42820.19	792.46	2286241.58

The output tables can be pivoted:

Double click on **Output Table**> Click **Pivot**> Click **Transpose Rows and Columns**

Report						
pcexp Per capita expenditure in local current prices						
	quintile Per capita expenditure quintile					
	1	2	3	4	5	Total
Mean	8198.1845	15702.0697	23827.4893	36774.5841	83799.3472	33659.0469
N	3831	3832	3832	3832	3831	19158
Std. Deviation	2621.49714	2077.55982	2745.34882	5141.47239	74389.85083	42820.18884
Minimum	792.46	12146.83	19310.59	28871.18	46935.50	792.46
Maximum	12146.49	19310.52	28870.61	46929.36	2286241.58	2286241.58

**Practice**

Complete the dummy tables 6 and 7 above.

**T-test**

The t-test assesses whether the means of two groups are statistically different from each other. This analysis is appropriate whenever you want to compare the means of two groups.

T-TEST compares sample means by calculating Student's *t* and displaying the two-tailed probability of the difference between the means. Statistics are available for one sample (tested against a specified value), independent samples (different groups of cases), or paired samples (different variables). Other procedures that compare group means are ANOVA, ONEWAY, UNIANOVA, GLM, and MANOVA. (GLM and MANOVA are available in the SPSS Advanced Models option.)

**One-sample test**

This type of T-test helps the analyst decide if a sample mean is different from a hypothesized population mean or a standard value.

Example:

T-TEST TESTVAL 28000 /VARIABLES=ASSET.

This one-sample t test compares the means of ASSET with the standard value (28000).

**Independent-samples test**

The two sample t-test simply tests whether or not two independent populations have different mean values on some measure, for example in poverty or nutrition.

To request an independent-samples test, use the GROUPS and VARIABLES subcommands. The output includes a Group Statistics table showing summary statistics by group for each dependent variable and an Independent-Samples Test table showing both pooled- and separate-variance estimates, along with the *F* value used to test homogeneity of variance and its probability. The two-tailed probability is displayed for the *t* value.

T-TEST

```
GROUPS = quintile(1 5)
/MISSING = ANALYSIS
/VARIABLES = pcexp
/CRITERIA = CI(.95) .
```

**Group statistics**

	quintile per capita expenditure quintile	N	Mean	Std. Deviation	Std. Error Mean
Pcexp monthly per capita expenditure	1	3831	8198.1845	2621.49714	42.35389
	5	3831	83799.3472	74389.85083	1201.87032

**Independent samples test**

		Levene's Test for Equality of Variances		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
pcexp monthly per capita expenditure	Equal variances assumed	733.603	.000	-62.864	7660	.000	-75601.163	1202.616	-77958.62	-73243.71
	Equal variances not assumed			-62.864	3839.513	.000	-75601.163	1202.616	-77958.62	-73243.71

	not assumed						6264		99	34
--	-------------	--	--	--	--	--	------	--	----	----

Note: This independent-samples t-test compares the means of the two groups defined by values 1 and 5 of QUINTILE for variable PCEXP. Probability levels are two-tailed. To obtain the one-tailed probability, divide the two-tailed probability by 2.

### Paired-samples test

The paired samples t-test compares the means of two variables. It computes the difference between the two variables for each case, and tests to see if the average difference is significantly different from zero.

To request a paired-samples test, use the PAIRS subcommand. The output includes: a Paired Statistics table showing univariate statistics by pairs, a Paired Samples Correlations table showing correlation coefficients and the two-tailed probability level for a test of the coefficient for each pair, and a Paired Samples Test table showing the paired differences between the means and two-tailed probability levels for a test of the differences.

#### Example:

If we have per capita calorie intake of same population for two different times (say for 2003 and 2004), we can do a paired sample t-test to see if the difference of calorie intakes are statistically significant or not.

```
T-TEST PAIRS=pccal_03 WITH pccal_04 (PAIRED)
/CRITERIA=CI(.9500)
/MISSING=ANALYSIS.
```

### Correlation Analysis

Correlation analysis determines the extent to which changes in the value of an attribute (such as landholding size) are associated with changes in another attribute (such as per capita expenditure).

CORRELATIONS (alias PEARSON CORR) produce Pearson product-moment correlations with significance levels and, optionally, univariate statistics, covariances, and cross-product deviations. Other procedures that produce correlation matrices are PARTIAL CORR, REGRESSION, DISCRIMINANT, and FACTOR.

A simple variable list on the VARIABLES subcommand produces a square matrix. We can also request a rectangular matrix of correlations between specific pairs of variables or between variable lists using the keyword WITH on VARIABLES.

#### Example:

```
CORRELATIONS
/VARIABLES=tland vasset hhsizes.
```

### Correlations

		tland07 Total owned land in 2007 (homestead +cult+other)	vasset07 Value of durbale assets in 2007 (consdur+ag durab+nona gdur jewelry)	hhsiz Household size
tland07 Total owned land in 2007 (homestead+cult+other)	Pearson Correlation Sig. (2-tailed) N	1  425	.592  425	.298  425
vasset07 Value of durbale assets in 2007 (consdur+agd durab+nona gdur jewelry)	Pearson Correlation Sig. (2-tailed) N	.592  425	1  425	.231  425
hhsiz Household size	Pearson Correlation Sig. (2-tailed) N	.298  425	.231  425	1  425

The correlations matrix gives the values of the specified correlation tests. Each row of the table corresponds to one of the variables. Within each row, the figures refer to the correlation, the two-tailed significance, and the number of observations. Each column also corresponds to one of the variables. In this example, the cell at the 3rd column of the 2<sup>nd</sup> row is the correlation of owned land with household asset. Here total owned land is positively and linearly correlated, with a coefficient of correlation=0.592. This means that as total land increases, the value of assets also increases linearly. This correlation coefficient is significantly different from zero, since the two-tailed significance level is <.01.

### Crosstabs

A cross tabulation (cross tab) displays the joint distribution of two or more variables.

The CROSSTABS command produces contingency tables showing the joint distribution of two or more variables that have a limited number of distinct values. The frequency distribution of one variable is subdivided according to the values of one or more variables. The unique combination of values for two or more variables defines a cell. CROSSTABS generates a Case Processing Summary table, a Cross Tabulation table, and a Chi-Square Tests table.

#### Example:

To see the age group distribution of population, we can do a cross tabulation of age group by gender.

```
CROSSTABS
  /TABLES=agegroup BY sex/FORMAT= AVALUE TABLES
  /STATISTIC=CHISQ
  /CELLS= COUNT
  /COUNT ROUND CELL .
```

**Case Processing Summary**

	Cases					
	Valid		Missing		Total	
	N	Percent	N	Percent	N	Percent
agegroup age group * sex Sex	92610	100.0%	0	.0%	92610	100.0%

**agegroup age group \* sex Sex Crosstabulation**

Count		sex Sex		
		1 Male	2 Female	Total
		agegroup age group	1.00 age<5 years	6242
	2.00 age 5-11 years	10470	9055	19525
	3.00 age 12-17 years	7183	5740	12923
	4.00 age 18-34 years	9849	12904	22753
	5.00 age 35-54 years	8715	8773	17488
	6.00 age 55 years and above	4775	3321	8096
	Total	47234	45376	92610

**Chi-Square Tests**

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	935.01	5	.000
Likelihood Ratio	937.74	5	.000
Linear-by-Linear Association	24.94	1	.000
N of Valid Cases	92610		

a. 0 cells (.0%) have expected count less than 5. The minimum expected count is 3966.79.

The variable AGEGROUP defines the rows and the variable SEX defines the columns of the Cross Tabulation table. CELLS show the row and column counts. STATISTICS requests the chi-square statistics displayed in the Chi-Square Tests table.

The result is significant (asymp. Sig. <.01), meaning that the distribution of age group differs by gender.

### **Chi-square**

This tests the statistical significance of the cross tabulations. Chi-squared should not be calculated for percentages. The cross tabs must be converted back to absolute counts (numbers) before calculating chi-squared. Chi-squared is also problematic when any cell has a joint frequency of less than five.

#### **Practice**

T-test per capita expenditure for urban versus rural (use HHEXPENDITURE.SAV)

Correlation of variables FOODX NONFOODX and ASSET

Do a crosstab of Female headed household by urban rural

### **Generating Plots and Graphs**

Plots and graphs are useful for explaining the analysis and data, and lead to better understanding.

The command GRAPH generates a high-resolution chart by computing statistics from variables in the active dataset and constructing the chart according to your specification. The chart can be a bar chart, pie chart, line chart, error bar chart, high-low-close histogram, scatter plot, or Pareto chart. The chart is displayed where high-resolution display is available and can be edited with a chart editor and saved as a chart file.

Example: Create a bar graph of mean per capita expenditure by expenditure quintile.

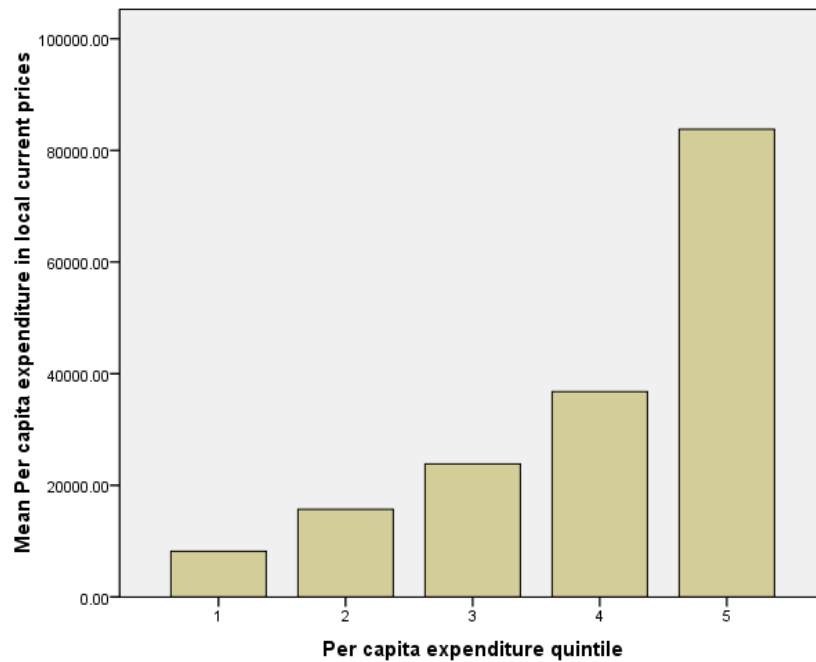
Syntax:

RANK pcexp /Ntiles (5) into quintile.

Variable label quintile "per capita expenditure quintile".

GRAPH /BAR (SIMPLE) = MEAN (pcexp) BY quintile.

Figure 1



GRAPH computes aggregated functions to obtain the values needed for the requested chart and calculates an optimal scale for charting. The chart title, subtitle, and footnote are assigned as they are specified on the TITLE, SUBTITLE, and FOOTNOTE subcommands. If you do not use these subcommands, the chart title, subtitle, and footnote are null.

The syntax for title, subtitle, and footnote would be as follows:

```
GRAPH Title "Mean per capita expenditure by expenditure quintile"
```

```
/subtitle "All households"
```

```
/footnote "source: NLSS data, 2003"
```

```
/BAR (SIMPLE) = MEAN (pcexp) BY quintile.
```

Example: Create a histogram of mean per capita expenditure.

For this example, we will drop some extreme outliers from this analysis

Syntax:

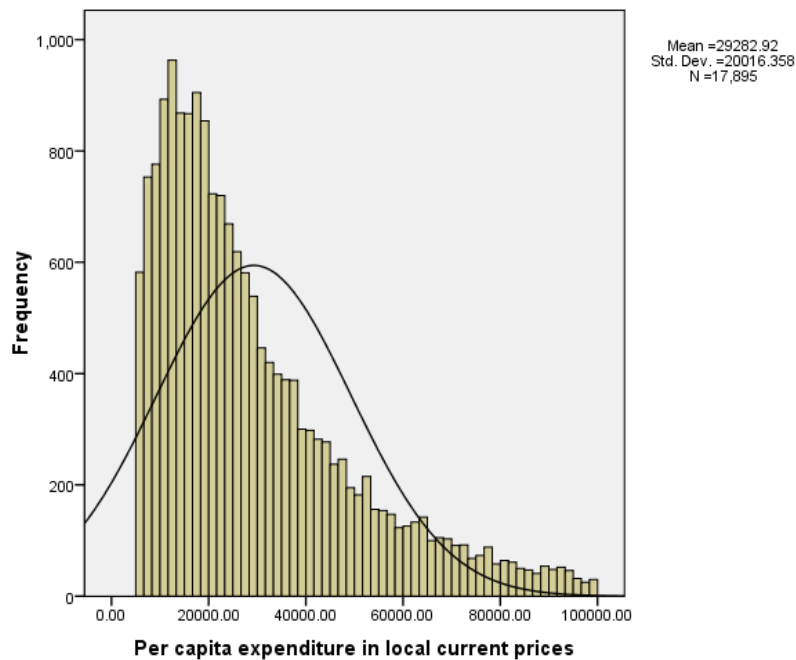
```
Temporary.
```

```
select if (pcexp>5000 & pcexp<100000).
```

```
GRAPH
```

```
/HISTOGRAM(NORMAL)=pcexp.
```

Figure 2



**Summary functions for graphs:**

**MINIMUM:** Minimum value of the variable

**MAXIMUM:** Maximum value of the variable

**N:** Number of cases for which the variable has a non missing value

**SUM:** Sum of the values of the variable

**CUSUM:** Sum of the summary variable accumulated across values of the category variable

**MEAN:** Mean

**STDDEV:** Standard deviation

**VARIANCE:** Variance

**MEDIAN:** Median

**GMEDIAN:** Group median

**MODE:** Mode

**PTILE(x):** Xth percentile value of the variable X must be greater than 0 and less than 100

**PLT(x):** Percentage of cases for which the value of the variable is less than x

**PGT(x):** Percentage of cases for which the value of the variable is greater than x

**NLT(x):** Number of cases for which the value of the variable is less than x

**NGT(x):** Number of cases for which the value of the variable is greater than x

**PIN(x1,x2):** Percentage of cases for which the value of the variable is greater than or equal to x1 and less than or equal to x2. x1 cannot exceed x2

**NIN(x1,x2):** Number of cases for which the value of the variable is greater than or equal to x1 and less than or equal to x2. x1 cannot exceed x2. Summary functions are usually used with summary variables (variables that record continuous values, such as age or expenses).

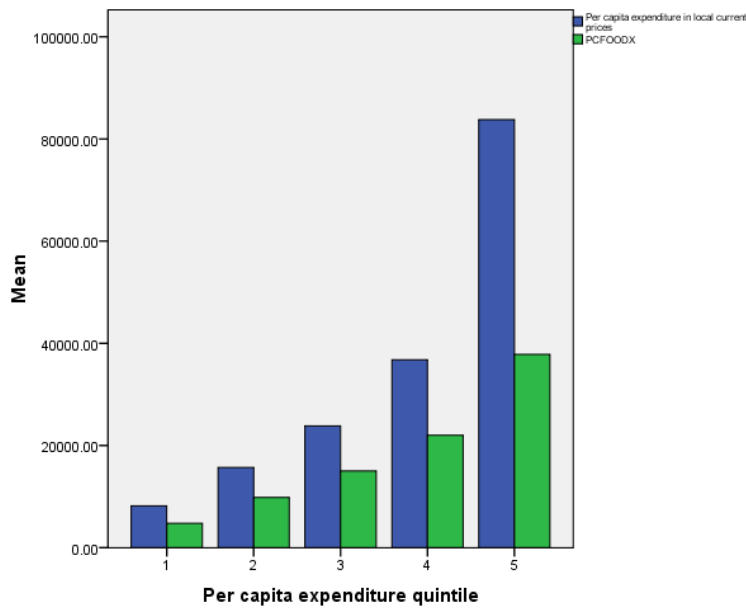
To use a summary function, specify the name of one or more variables in parentheses after the name of the function, as in:

GRAPH /BAR = SUM(PCEXP) BY QUINTILE.

Multiple summary functions can be specified for more chart types. For example, the same function can be applied to a list of variables, as in:

GRAPH /BAR = SUM(PCEXP SHAREFD) BY QUINTILE.

**Figure 3**



This syntax is equivalent to:

GRAPH /BAR = SUM(PCX) SUM(PCFOODX) BY QUINTILE.

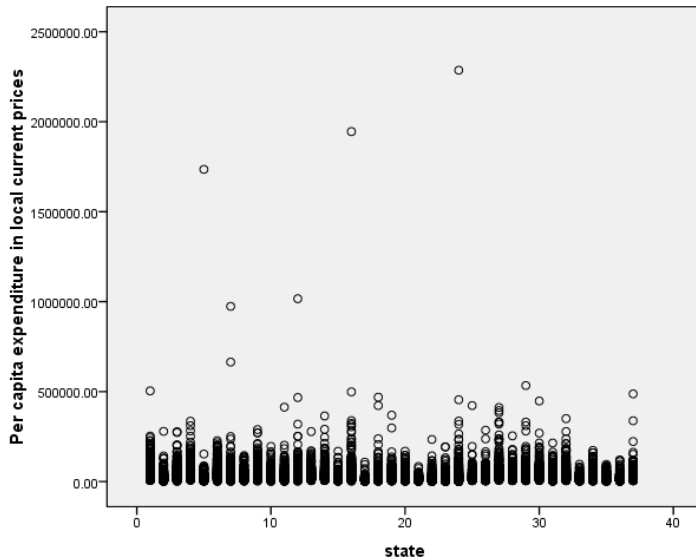
Different functions can be applied to the same variable, as in:

GRAPH /BAR = MEAN(PCEXP) MEDIAN(PCFOODX) BY QUINTILE.

Example: Two way scatter plot graph (state with per capita expenditure):

```
GRAPH
/SCATTERPLOT(BIVAR)=state WITH pcexp
/MISSING=LISTWISE.
```

Figure 4



#### Practice

Generate a bar graph for mean per capita expenditure by urban rural (use HHEXP.SAV)

Syntax:

Generate a scatter plot for household size with per capita expenditure (use HHEXP.SAV)

Syntax:

Generate a pie graph for total population by urban rural (Use ROSTER.SAV)

Syntax:

### Creating a Master File for Doing Regression Analysis

The following variables can be included in the file:

- Household Identification number
- ID of individuals within Households
- Per capita expenditure
- Household size
- Age of household head (from household roster)

Education of household head (from household roster)  
 Female headed household dummy (from household roster)  
 Total value of household asset  
 Number of dependents members of less than 15 or greater than 60 (from household roster)  
 Land holding size  
 Any member of the household not eaten anything for a whole day due to hardship at home (dummy)  
 Had to borrow food from friends or neighbors due to hardship at home (dummy)  
 Region dummy (RECODE command can be used for generating dummy variables)

## **Poverty Analysis**

In this manual, we don't teach how to derive the poverty line, but illustrate how to calculate poverty measures using existing poverty line. The poverty line can be used to estimate:

Head count ratio (p0)  
 Poverty gap ratio (p1)  
 Poverty gap squared (p2)

**Headcount Ratio (P0):** The proportion of total population who are below the poverty line.

**Poverty Gap Ratio:** Expresses the total amount of money which would be needed to raise the poor from their present incomes (y) to the poverty line (z), as a proportion of the poverty line, and averaged over the total population. This measures the *depth* of poverty. The poverty gap ratio is the mean distance separating the population from the poverty line (with the non-poor being given a distance of zero), expressed as a percentage of the poverty line.

Estimate poverty gap

$$\text{Poverty Gap (P1)} = \frac{1}{n} \sum_{i=1}^q [(Z - Y_i)/Z]$$

= mean proportionate poverty gap across the whole population (zero gap for the non-poor)

n=number of total population

Z=poverty line

Y<sub>i</sub>=Income of poor

**Poverty Gap Squared:** One problem with the poverty gap ratio is that it ignores variations in expenditure (the distribution of income and spending) amongst the poor, since the gap is an average. This can be rectified by squaring the poverty gap, which provides the statistical second moment of the distribution. Squaring the individual poverty gaps means that the larger gaps count for more than the smaller gaps, and hence the measure captures the *severity* of poverty in a population.

$$\text{Severity of poverty (P2)} = \frac{1}{n} \sum_{i=1}^q [(Z - Y_i)/Z]^2$$

### Practice

Example for estimating poverty gap & severity of poverty:

Income of four individuals in a sample: 1,2,3,4

Poverty line =  $Z = 3$ ; number of population  $N=4$ ,  $q$ =number of population below or on the poverty line

Head count ratio ( $P_0$ ) =  $q/N = 3/4 = 0.75$  or 75%

$$\begin{aligned} \text{Poverty gap } P_1 &= [(3-1)/3 + (3-2)/3]/4 \\ &= [(2/3) + (1/3)]/4 \\ &= [(3/3)/4] \\ &= 1/4 \text{ or } 0.25 \end{aligned}$$

\*\* Poverty analysis by using NLSS data.

\*\* Preparing the expenditure file to merge with roster file.

GET

FILE='U:\Nigeria\Nigeria\_NLSS\_03\_04\data\HHEXP.sav'.

sort cases by state sector ric hhno pid.

save outfile 'U:\Nigeria\Nigeria\_NLSS\_03\_04\data\HHEXP\_N.sav'

/keep state sector ric hhno weightea fpindex nfpindex fdtotby fdtotpr  
fdtot fdtotdr edtexp hltexp renthh nfdftot nfdiqtot nfdtot nfdtotdr hhexp  
hhexpdr pcexpdr pcexp sharefd sharefd2 aeexpdr hhsz\_1 quinttr popwt  
corep modp corepr modpr dpdpr agegrp5 pov age1560 sizehh othp  
s2cq12 agehlt agegrp10 occgrp edgrp englit nglit lit edage enroll1.

\*\* merging roster and household expenditure files.

GET

FILE='U:\Nigeria\Nigeria\_NLSS\_03\_04\data\Sec1\_Roster.sav'.

sort cases by state sector ric hhno pid.

\*\*Checking if there is any duplicate cases.

compute flag=0.

if (state=lag(state) & sector=lag(sector) & ric=lag(ric) & hhno=lag(hhno) &  
pid=lag(pid)) flag=1.

freq flag.

\*\* no duplicate cases in the roster file.

match files /file=\*

/table ='U:\Nigeria\Nigeria\_NLSS\_03\_04\data\HHEXP\_N.sav'

/by state sector ric hhno.

execute.

\*\* We are using an arbitrary poverty line to do the poverty estimates.

```

** Say poverty line is 20,000 naira.

compute POVLINE=20000.
if (PCEXP<POVLINE) POOR=1.
if (PCEXP>=POVLINE) POOR=0.

variable label POOR 'poor or non-poor'.
value label poor 1 'poor (below poverty line)' 0 'Non poor (above poverty
line)'.
freq poor.

* estimating poverty gap.

if (poor=1) gap=(povline-pcexp)/povline.
if (poor=0) gap=0.

** generating a break variable for aggregating.
compute total=1.

AGGREGATE
/OUTFILE=* MODE=ADDVARIABLES
/BREAK=total
/gap_sum 'summation of gaps'=SUM(gap)
/N 'number of population'=NU(pid).

* poverty gap.
COMPUTE P1=gap_sum/N
variable label P1 'poverty gap'.

** Estimating severity of poverty (squared poverty gap).

compute gapsq=((povline-pcexp)/povline)**2.
if (poor=0) gapsq=0.

AGGREGATE
/OUTFILE=* MODE=ADDVARIABLES
/BREAK=total
/gapsq_sum 'summation of squared gaps'=SUM(gapsq).

compute P2=gapsq_sum/N.
variable label P2 'severity of poverty'.
execute.

```

**Practice:** Complete the following table

	Urban	Rural	All
Poverty headcount (percent)			
Poverty gap			
Squared poverty gap			

## **Nutrition Analysis**

Nutrition analysis is measuring the prevalence of stunting, wasting and underweight. Stunting is low height for age. Wasting is low weight for height, and underweight is low weight for age.

In this case, let us define:

Stunting = Height for age Z-score  $< -2$

Wasting = weight for height Z-score  $< -2$

Underweight = weight for age Z-score  $< -2$

To generate a stunting variable using the 2003 Nigeria DHS data set for children, perform the following commands:

```
GET
```

```
FILE='D:\Training\ngkr4zrt.SAV'.
```

```
VALUE LABELS v025
```

```
1 "Urban"
```

```
2 "Rural".
```

```
RECODE FOR ANTHROPOMETRY.
```

```
COMPUTE HAZ = hw5 / 100.
```

HAZ=height for age Z-score. You must divide the hw5 variable in the DHS dataset by 100 to get the Z-score.

(We can exclude extreme outliers (HAZ  $< -6.0$  or  $> 6.0$ ) from this analysis.)

```
RECODE
```

```
HAZ (Lowest thru -6.0001=SYSMIS) (6.0001 thru Highest=SYSMIS) .
```

Generating a stunting variable:

```
if (HAZ $< -2$ ) stunt=1.
```

```
if (HAZ $\geq -2$ ) stunt=0.
```

```
VARIABLE LABELS stunt 'Stunting (HAZ $< -2.0$ )'.
```

```
VALUE LABELS stunt
```

```
0 '(HAZ $< -2$ )'
```

1 '(HAZ>=-2)'.

**Practice:** Complete the following table

**Table 9 Prevalence of malnutrition among preschool children aged 6 to 60 months by urban rural**

Location of household	Sex of Children	Number of children	Percent Stunting ( HAZ <-2)	Percent Underweight (WAZ <-2)	Percent Wasting (WHZ<-2)
Urban	Boys				
	Girls				
	All				
Rural	Boys				
	Girls				
	All				
All	Boys				
	Girls				
	All				

Note: HAZ = height-for-age Z-score; WAZ = weight-for-age Z-score; WHZ = weight-for-height Z-score.

**Practice**

Do a crosstab: Stunted children by underweight mother  
 Generating underweight mother variable:

MISSING VALUES v445 ("9999").  
 COMPUTE bmi = v445 / 100 .

If (BMI <18.5) undwt\_mom=1.  
 If (BMI >=18.5) undwt\_mom=0.  
 Variable label undwt\_mom 'Underweight mother'.  
 Value label undwt\_mom 0 'not under weight' 1 'Under weight (BMI<18.5)'

CROSSTABS  
 /TABLES=undwt\_mom BY stunt1  
 /FORMAT= AVALUE TABLES  
 /STATISTIC=CHISQ CORR  
 /CELLS= COUNT ROW COLUMN TOTAL .

**Regression Analysis**

Regression analysis look for the causal effect of one variable upon another, for example the effect of household expenditure on calorie intake.

The REGRESSION command calculates multiple regression equations and associated statistics and plots. REGRESSION also calculates colinearity diagnostics, predicted values, residuals, measures of fit and influence, and several statistics based on these measures.

Example:

Click on **Analyze**>Click **Regression**>Click **Linear**> Select **Dependent variable**> Select **Independent variables**> Select **Method**>Click **OK** or **Paste Syntax**

The output would be as follows::

REGRESSION

/MISSING LISTWISE

/STATISTICS COEFF OUTS R ANOVA

/CRITERIA=PIN(.05) POUT(.10)

/NOORIGIN

/DEPENDENT pcexp

/METHOD=ENTER hhsz1 radio tv nrooms sanitary cultland pcatta1 pprice1.

The following tables are generated by these commands:

**Variables Entered/Removed<sup>b</sup>**

Model	Variables Entered	Variables Removed	Method
1	pprice1 hh rice /hhsz1, sanitary Sanitary latrine=1, pcatta1 hh att /hhsz1, radio Household has a radio=1, hhsz1 Household size, tv Household has a television=1, cultland 1 cultivable land owned (acres), nrooms Number of rooms <sup>a</sup>		. Enter

a. All requested variables entered.

b. Dependent Variable: pcexp Monthly per capita hhold csm

*The Variables Entered/Removed table shows the variables entered in the regression. Sometimes variables can be removed from the regression for co-linearity or missing data points. In this case, no variables were removed.*

**Model Summary**

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.650 <sup>a</sup>	.422	.422	365.21035

**Model Summary**

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.650 <sup>a</sup>	.422	.422	365.21035

a. Predictors: (Constant), pprice1 hh rice /hhsiz, sanitary Sanitary latrine=1, pcatta1 hh atta /hhsiz, radio Household has a radio=1, hhsiz1 Household size, tv Household has a television=1, cultland 1 cultivable land owned (acres), nrooms Number of rooms

*The Model Summary table shows the r-squared and standard error of the estimate.*

**ANOVA<sup>b</sup>**

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	4.909E8	8	6.136E7	460.043	.000 <sup>a</sup>
	Residual	6.710E8	5031	133378.601		
	Total	1.162E9	5039			

a. Predictors: (Constant), pprice1 hh rice /hhsiz, sanitary Sanitary latrine=1, pcatta1 hh atta /hhsiz, radio Household has a radio=1, hhsiz1 Household size, tv Household has a television=1, cultland 1 cultivable land owned (acres), nrooms Number of rooms

b. Dependent Variable: pexp Monthly per capita hhold csm

*The analysis of variance (ANOVA) table analyses the variance of the regression model. The F statistic shows whether the model variables are jointly significant or not. If significance is < .01 (or .05) then the model is significant.*

**Coefficients<sup>a</sup>**

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	379.712	22.475		16.895	.000
	hhsiz1 Household size	-59.480	2.626	-.278	-22.654	.000
	radio Household has a radio=1	63.617	15.253	.045	4.171	.000
	tv Household has a television=1	456.807	21.143	.246	21.605	.000
	nrooms Number of rooms	122.110	5.072	.310	24.077	.000
	sanitary Sanitary latrine=1	273.718	16.409	.188	16.681	.000
	cultland 1 cultivable land owned (acres)	22.559	2.182	.118	10.338	.000
	pcatta1 hh atta /hhsiz	20.621	1.194	.186	17.265	.000
	pprice1 hh rice /hhsiz	.672	.035	.211	19.282	.000

Coefficients<sup>a</sup>

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	379.712	22.475		16.895	.000
	hhsz1 Household size	-59.480	2.626	-.278	-22.654	.000
	radio Household has a radio=1	63.617	15.253	.045	4.171	.000
	tv Household has a television=1	456.807	21.143	.246	21.605	.000
	nrooms Number of rooms	122.110	5.072	.310	24.077	.000
	sanitary Sanitary latrine=1	273.718	16.409	.188	16.681	.000
	cultland 1 cultivable land owned (acres)	22.559	2.182	.118	10.338	.000
	pcatta1 hh atta /hhsz	20.621	1.194	.186	17.265	.000
	pprice1 hh rice /hhsz	.672	.035	.211	19.282	.000

a. Dependent Variable: pexp Monthly per capita hhold csm

The coefficients table provides the effect of each of the independent variable on the dependent variables. It also provides the t-values and level of significance of the regression coefficient (given on column B). For example: household size has a negative relation with per capita expenditure that means if household size increases per capita expenditure drops by 59.48 naira. This is statistically significant as the sig is <.01.

## Probit

PROBIT can be used to estimate the effects of one or more independent variables on a dichotomous dependent variable (such as dead or alive, employed or unemployed, product purchased or not).

### Practice

Do a linear regression as follows:

Dependent variable: Per capita household expenditure

Independent variables: Household size, Education of household head, Age of household head, Land holding size, Urban-rural dummy, Female household head dummy [Dummy for if a member not eaten whole day, Dummy for borrowing food from friends, Location dummy]

Merge all the required variables in a master file:

```
GET
```

```
FILE='D:\Training\HHEXP.SAV'.
```

```
match files/file=*
```

```
/table 'D:\Training\asset.SAV'
```

```
/by state sector ric hhno.
```

```
match files/file=*
```

```
/table 'D:\Training\landsize.SAV'
```

```
/by state sector ric hhno.
```

Click on **Regression**> Click **Linear**> Select **Dependent variable** (PCEXP)> Select **Independent variables** (HHISIZE, ASSET, LAND etc)> Select **Method** (ENTER)> Click **OK** or **Paste Syntax**

You can change the option before clicking **OK**

Default options are: probability entry (0.5) and removal (0.1), includes constant in the equation and exclude missing listwise,

Syntax:

```
REGRESSION
```

```
  /MISSING LISTWISE
```

```
  /STATISTICS COEFF OUTS R ANOVA
```

```
  /CRITERIA=PIN(.05) POUT(.10)
```

```
  /NOORIGIN
```

```
  /DEPENDENT pcexp
```

```
  /METHOD=ENTER hhsiz asset landacre.
```

This manual is a very short introduction of SPSS for beginners. It has been developed with an emphasis for its potential use to analyze household data related to poverty and nutrition. We only discussed a few basic features of SPSS in this training. There are hundreds of other features in SPSS that a user can explore for his/her specific discipline. Users are encouraged to use the SPSS “help” menu. We also advise students to make use of the tutorial that comes with the SPSS software.

## **References**

Babu, Suresh and Rhoe, Valerie. "Poverty Measurement and Analysis" presentation. International Food Policy Research Institute (IFPRI).

"Documenting your SPSS Data Sets." Children's Mercy Hospitals and Clinics.  
<http://www.childrensmercy.org/stats/data/document.asp> (accessed October 2008).

Fink, Gayle. "SPSS Basics for Institutional Researchers" presentation. Association for Institutional Research, University System of Maryland, University of Maryland.

Mignot, Nicholas. Stata Manual used for IFPRI Vietnam Poverty Mapping and Market Access project.

"Poverty and Income Distribution." David Harvey's Research Web Site.  
<http://www.staff.ncl.ac.uk/david.harvey/AEF806/Poverty.html> (accessed October 2008).

SPSS Base Manual.

SPSS Learning Module. UCLA: Academic Technology Services, Statistical Consulting Group.  
<http://www.ats.ucla.edu/stat/SPSS/modules/merge.htm> (accessed October 2008).