March 2013

Technical Brief

How to approach a dataset

Part 2: Data preparation

Aldo Benini

A note for ACAPS

How to approach a dataset -

Part 2: Data preparation

Version 15 March 2013

Table of Contents

ACKNOWLEDGEMENT 5
ACRONYMS AND EXPRESSIONS 5
1. SUMMARY 6
2. INTRODUCTION 8
2.1 WHAT THIS IS ABOUT 8
[Sidebar:] The scenario assumed for this note *8*
2.2 TYPICAL CHALLENGES WITH NEW DATASETS 9
2.3 EXCEL TECHNIQUES 10
2.4 RELATED ACAPS RESOURCES 10
3. DATA AND WORKBOOK 11
[Sidebar:] The source of the demonstration data - Bangladesh assessment *11*
4. STEPS IN DATA PREPARATION 14
4.1 BACKUP AND FILE NAME CONVENTIONS 14
4.2 RECORDS AND VARIABLES 15
4.2.1 RECORD IDENTIFIERS 15
4.2.2 MARKING AND DOCUMENTING VARIABLES 16
[Sidebar:] Long-form to wide-form *19*
4.2.3 DESCRIPTIVE STATISTICS 21
4.2.4 CLARIFYING MISSING VALUES 23
4.3 VALUE INSPECTION AND RECODING 24
[Sidebar:] Listing the unique values of all variables *26*
4.4 FINAL SHAPE OF THE DATA TABLE 28
5. AUXILIARY INFORMATION 30
[Sidebar:] Alternating between Excel and STATA *30*
6. CONCLUSION 32

Tables and figures

Table 1: Bangladesh assessment timeline 12
Table 2: Record identifiers after appending regional data tables 16
Table 3: Color codes for types of variables 17
Table 4: Recoding of verbose entries 26
Table 5: STATA sample output - list of nominal duplicate sites 31

Figure 1: Flowchart of major data preparation steps 7
Figure 2: Create names from selection 17
Figure 3: Setting up the Variables sheet 18
Figure 4: The function INDIRECT referencing a named range 21
Figure 5: Example of inspecting descriptive statistics 22
Figure 6: Selecting all blank cells within a range 23
Figure 7: Example of a recoding table with prefixes 25
Figure 8: The user-written function UNIQUE 27
Figure 9: Defining a recoding table 27

Acknowledgement

This note was written with the very active support of Sandie Walton-Ellery, ACAPS, Benoit Munsch, Solidarités International, both in Dhaka, Bangladesh as well as Emese Csete, ACAPS Nairobi, and Patrice Chataigner, ACAPS Geneva. Walton-Ellery and Munsch supplied the raw dataset, interpretation of select findings, and the description of the 2011 assessment context that the reader may find in the main body of this note. Csete and Chataigner helped with the alignment between the text of this note and the flow of the demonstration workbook.

Acronyms and expressions

ACAPS	Assessment Capacities Project
ECB	Emergency Capacity Building
GoB	Government of Bangladesh
R	A statistical program
SPSS	A statistical program
STATA	A statistical program
VBA	Visual Basic of Applications

1. Summary

This is the second of three notes on "how to approach a dataset" that emerges from a needs assessment following a disaster. It addresses typical challenges of data management or, as some might say, data preparation. A companion note looks at the design of the database and another at analysis.

The note speaks to typical data editing and formatting challenges that assessment analysts face when data collection and entry were designed and done rapidly, without the benefit of pre-tested templates and built-in validation. It demonstrates how some of them can be handled with data from a real assessment in Bangladesh.

The demonstration workbook, in a sequence of sheets, takes the data from its initial close association with the questionnaire to a shape that is better suited to the analysis. Some of the sheets are purely didactic; in real life, the analyst, already pressed for time, will perform several or all of the preparation steps in the same sheet.

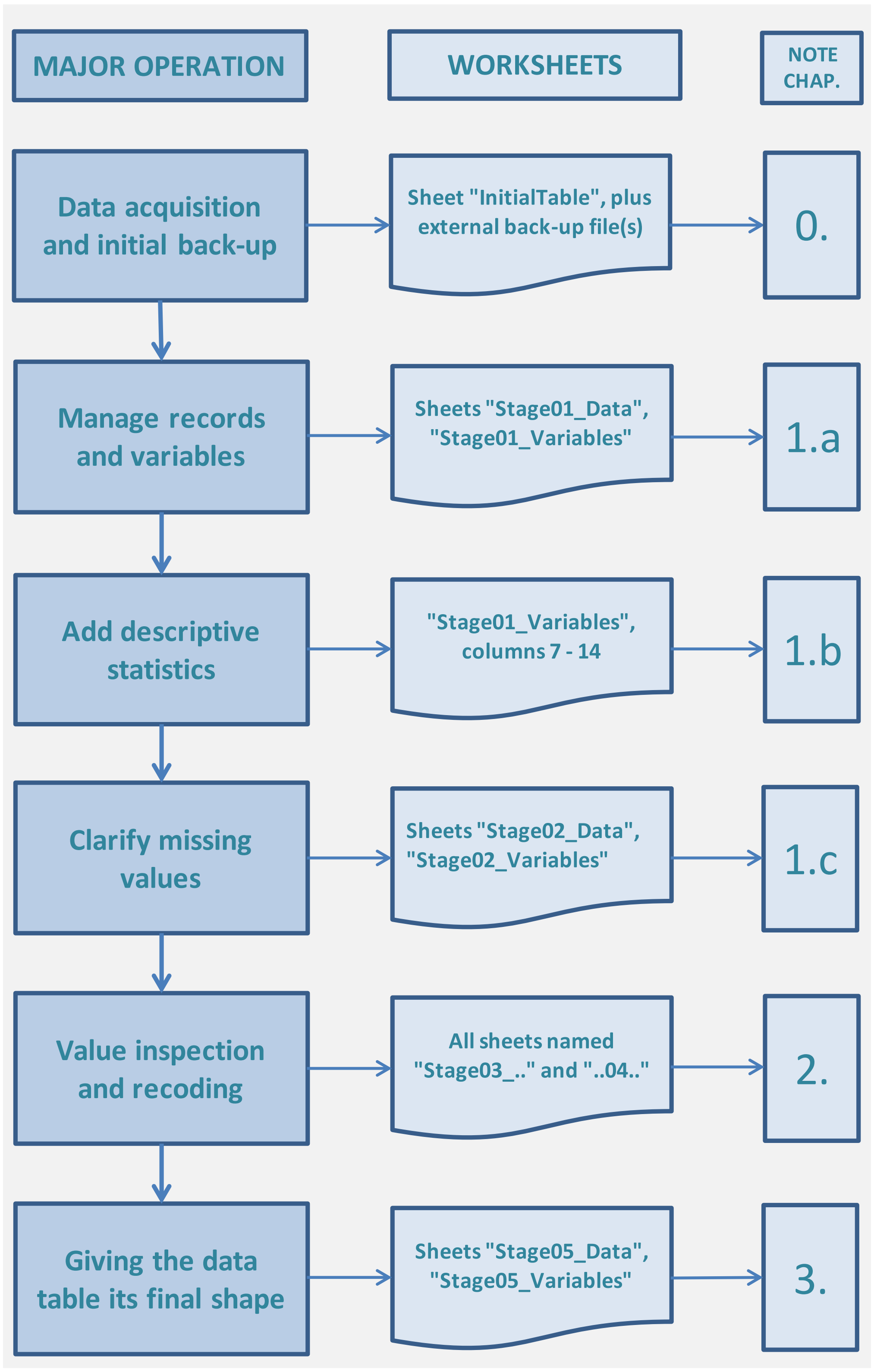
The note emphasizes the value of proper documentation. The workbook demonstrates how the sheet documenting the variables can instantly, with the help of an Excel function, be turned into a table of descriptive statistics. These facilitate both data inspection at this point and the answers to a potentially large number of questions that may crop up during the report writing.

How thoroughly the analyst can in fact inspect (and then edit) the data depends on time, style and work standards; we offer graded advice from superficial make-over to comprehensive listings of all values present in all variables and their recoding as desired.

Our assumption is that careful data preparation will pay dividends in analysis and reporting. This may not always succeed; the pressure to release preliminary findings early can interfere with the initial due diligence. By and large, however, the data preparation should be able to remove most of the obstacles on the way to smoother analysis.

The flowchart on the next page enumerates the major operations, the affiliated sheets of the demo workbook, as well as the corresponding sections of this note.

Figure 1: Flowchart of major data preparation steps


MAJOR OPERATION
WORKSHEETS
NOTE CHAP.
Data acquisition and initial back-up
Sheet "InitialTable", plus external back-up file(s)
0.
Manage records and variables
Sheets "Stage01_Data", "Stage01_Variables"
1.a
Add descriptive statistics
"Stage01_Variables", columns 7 - 14
1.b
Clarify missing values
Sheets "Stage02_Data", "Stage02_Variables"
1.c
Value inspection and recoding
All sheets named "Stage03_.." and "..04.."
2.
Giving the data table its final shape
Sheets "Stage05_Data", "Stage05_Variables"
3.

2. Introduction

2.1 What this is about

This note speaks to data analysts in needs assessment teams in post-disaster situations. It provides guidance for situations in which a new assessment dataset requires preparation before it is suitable for analysis.

Assessment formats vary; prescribing a uniform process is not feasible. The steps of preparing for analysis are exemplified with a dataset from Bangladesh. They are demonstrated in the accompanying Excel workbook.

Our goal is to provide a rational and time-saving approach to confronting a new dataset as it reaches the analyst after data collection and entry. Our objective, in dealing with the Bangladesh data, is to document the data and transform the table from an appearance close to the questionnaire to a shape conducive to analysis and reporting.

This note details the what, how, and why for each step. More detailed rationales are offered in sidebars, which the hurried reader may initially skip, to return there at leisure.

[Sidebar:] The scenario assumed for this note

Assessment context, team quality and data processing capacity vary widely. For what follows we assume a scenario that in some points is not too atypical of needs assessments conducted by NGOs. In other respects it simplifies field realities for the benefit of exposition. We assume:

- The assessment was designed with a common-sense view to what would likely prove important to know. *An analysis plan was not drawn up prior to data collection. Similarly, for the data management side, complete codebooks and data validation features were not in place.*
- The team creates a variety of document types: Formatted datasets, textual notes, images and maps. *This note addresses the processing of formatted datasets only.*
- The data have been minimally cleaned (some spelling differences removed in recurring names; numbers formatted as numbers, dates as Excel-compliant dates; the distinction between zero (0) and missing (blank) respected)). However, the cleaning is not entirely complete, and some variables likely need recoding, thus *warranting a complete inspection of all variables*.
- The data arrives in an integrated flat table. If the observed sites were reported in separate records (notably from team observations, male and female community groups), these tables have already been merged on common identifiers. *In other words, the analyst received the table in wide shape. Briefly we outline how to deal with long-shaped data, in which different records regarding the same site are stacked one upon the other.*
- There may be other datasets from secondary sources, from or about entities other than the observed localities, or from other assessments. *They are important, but are not considered in this note.*

Our overarching assumption is that assessment teams work under great time pressure, during data preparation as well as analysis. Pressures are strong to release preliminary findings, derived from data that may not yet have been properly processed. This may cause later inefficiencies or even erroneous findings. This note is to help expedite the process, in the belief that the effective and efficient data preparation will pay dividends during analysis and reporting.

This note is less relevant for assessments that work with database templates equipped with exhaustive codebooks, entry screens with in-built data validation, and pre-defined analysis plans. But we also believe that such privileged conditions occur in a minority of assessment situations, and that the concerns that this note addresses remain typical and well founded for many.

2.2 Typical challenges with new datasets

Commonly, analysts who receive incompletely edited datasets notice three kinds of challenges:

- Detectably incorrect and doubtful **values** (which call for data editing)
- Poorly formatted **variables** and **tables** (which call for better formatting)
- Pack of documentation of **records** and variables (which call for identifiers and metadata)

Within each of those broad groups, certain subtypes of errors and bad practice occur more frequently than others.

Challenges with values

- Missing values ("don't know", "not applicable", etc.) and negative observations (absence of a trait, zero of a quantity, "no" and other negated question forms) are not clearly distinguished. Blank cells in the data table may mean either.
- Entries in categorical variables come in multiple spelling variants; respondents chose from inconsistent category sets (e.g., of administrative units - "my city is your district").
- Numerical variables are laced with actual counts as well as broadly estimated magnitudes, particularly at the extremes.

Challenges with variables

- Supposedly numeric variables arrive in text format (often with dates), or have values written as text ("2 camps" in "Number of camps"), or blanks that Excel does not recognize as such (a systemic problem in Excel).
- The meaning, let alone the measurement, of a variable is not clear, or is no longer clear the way it is recorded in the data table.
- The effort needed to recode verbose or poorly coded variables is difficult to estimate; perfectly clean solutions may take too much time to produce.

Challenges with records

- There is no unique identifier; the unique identifier is not correct; none of the unique identifier(s) permits recreating the original sort order.
- Remarks qualifying the underlying unit (the assessed community) are hidden in Excel cell comments, which promptly go unnoticed in the analysis, instead of in a variable.

Challenges with tables

- The final data table resulted from a faulty appending of several partial tables (data entry was decentralized, and the order of variables was changed in one place).
- The table closely follows the questionnaire language and appearance. It is not formatted for analysis and reporting needs.

It is not possible to foresee all editing and formatting challenges that may obstruct the effective and efficient use of a needs assessment data set. Our demonstration dataset had been cleaned to some extent; the remaining problems on which we focus in this note have to do chiefly with missing vs. negative, selective recoding, and documenting variables.

2.3 Excel techniques

We manage and analyze data in the spreadsheet application MS Excel. We assume that readers have mid-level skills. They are familiar, or can familiarize using Excel's Help function, with the following features:

- R1C1 reference style
- Absolute, relative and mixed-typed cell references
- Named ranges
- The general workings of Excel functions.

The data preparation that we describe makes use of these specific functions: COUNT, COUNTA, MIN, MAX, SUM, MEDIAN, AVERAGE, IF, VLOOKUP, INDIRECT, CONCATENATE, TEXT, LEFT.

All operations can be achieved with those features. However, to obtain a complete listing of the unique values of all variables, we describe a time-saving procedure that relies on a small number of macros and user-written functions. These features are optional. They assume that users can access such functions (in the formula bar) and macros (through Alt+F8). Users do not have to write code, but if these features are to work in their own data files, the code stored in the demo workbook should be copied to a VBA module of their own.

To offer these user-defined functions and macros, the demo workbook has been saved in macro-enabled mode (.xlsm).

2.4 Related ACAPS resources

This note overlaps with these related notes published in the ACAPS resource repository:

- *How to approach a dataset - Part 1: Database Design*

  This is the companion note that concerns recommendations for database Design using Excel.

- *How to approach a dataset – Part 3: Analysis*

This is a companion note to this note, using the same Bangladesh data and taking them from the point where the data management part finishes.

- *Documenting methods and data in rapid needs assessments*[1]

This document shares with this note points regarding the documentation of variables, the limitations imposed by purposive sampling, and the use of the Excel function INDIRECT as a convenient tool to produce some of the descriptive statistics.

- *Severity rating: Analysis note*[2]

Exemplified with data from an assessment in Yemen, this document offers guidance on formal devices (tagging variables, Pivot table templates, semi-automated recoding). The major difference is that the Yemen data requires multiple records per assessed site (priority sector, issue severity rating) whereas our approach here deals with "one site = one record" situations.

- *Heat maps as tools to summarize priorities expressed in needs assessments*[3]

A demonstration, using a segment of the Bangladesh assessment data, of how priority data (with unique first, second .. nth choices from a menu of more than n options) can be handled for group comparisons and visualized in heat maps.

These documents are not necessary for the proper understanding of what follows here. They do elaborate on some of the topics. For example, *"Severity rating: Analysis note"* includes an extensive discussion of recoding issues.

3. Data and workbook

The demonstration data for this note is from the *"Joint Humanitarian Needs Assessment of Water-logged Areas of South West Bangladesh 2011"* conducted by a group of NGOs and assisted by ACAPS. The sidebar below gives background.

The demo workbook is called *HowToApproachADataset_Datamanagement_130315AB.xlsm* and can be found at *http://www.acaps.org/resources/advanced-material* .

[Sidebar:] The source of the demonstration data - Bangladesh assessment
By Sandie Walton-Ellery, ACAPS, Dhaka

[1] May 2012; http://www.acaps.org/resourcescats/download/documenting_methods_and_data_in_rapid_needs_assessments/110.

[2] January 2012 [This is the working title on the ACAPS Web page; the formal title on the note is "Data analysis in needs assessments"]; http://www.acaps.org/resourcescats/download/severity_rating_analysis_note/88

[3] October 2011; http://www.acaps.org/resourcescats/download/heat_maps_as_tools_to_summarise_priorities/69

The "Joint Humanitarian Needs Assessment of Water-logged Areas of South West Bangladesh" of summer 2011 was the upshot of efforts at assessment preparedness in Bangladesh that had been initiated earlier that year. What began as interest in improving assessments for the Emergency Capacity Building (ECB) consortium of seven agencies has grown, over a period of approximately eighteen months, into an assessment approach embedded in national coordination mechanisms with the active buy-in of a broad range of stakeholders including the Government of Bangladesh (GoB). The assessment approach has been used in the field several times and findings have informed response planning as well as refinements to the assessment methodology.

Following an initial phase of scoping and awareness raising, the first opportunity to test a common assessment approach came in August 2011, when severe water-logging affected the south-west of the country. Heavier than usual rains during the end of July and early August 2011 resulted in flooding in many areas of Bangladesh. In the southwestern districts of Shatkira, Khulna and Jessore, flood waters did not recede and potentially significant humanitarian needs in Southwest Bangladesh were identified. On 25th August, the Disaster Management Bureau, Ministry of Food and Disaster Management, estimated that over 800,000 people in Satkhira district were affected.

While multiple assessment reports were generated in the early stages of the flooding and water-logging, these reports were not sufficiently compatible in methodology to provide a consolidated overview of the situation. In the face of apparent growing humanitarian needs and fragmented information on these needs, ECB initiated a joint needs assessment, in which sixteen organizations collaborated.

The objectives of the joint assessment were:

- To provide a shared overview of the situation in all affected areas of the south west
- To identify immediate humanitarian needs that were not being addressed
- To understand recovery needs of affected people

Table 1: Bangladesh assessment timeline

26th August – 6th September	Deciding on the need for an assessment Planning for the assessment Travelling to field location
7th September	Training of teams in Khulna (field coordination hub)
8th – 12th September	Community level data collection
13th – 15th September	Data entry
16th-22nd September	Data processing, joint analysis, interpretation and report preparation

The assessment aimed to understand both the quantitative impact of the water-logging as well as the qualitative impact; i.e., how many people were affected and how they were affected.

Quantitative information was obtained through the collation of local government figures on the number of affected people. Assessment teams collected this information from the Union Councils (lowest tier of local government) directly and also from sub-district officials.

Qualitative information on the impacts of the water-logging was collected through male and female community group discussions and direct observations of assessment teams in 63 sites throughout the affected area.

The data set used in this document is taken from these community group discussions and direct observations.

The data segment used here

The original data were stored in a workbook with three sheets, one holding the assessment teams' direct observations (55 columns, with almost as many variables), one for the male community group information (115 columns), and one for the female groups (203 columns). Initially there were no matchable common identifiers. Once these were in place, the data for the 63 sites was merged in one sheet, filling 375 columns.

For the purposes of this demonstration, a digest of 119 variables was created, exemplifying the data types of interest. For reproducibility, the question numbering (which was used in the column headers of the original data tables) was retained as an element of the variable names.

In the original data set, the response to yes/no questions had, to a certain extent, already been coded using 1 for "yes" and 0 for "no", which is a recommended practice.

A peculiar feature needs to be pointed out: To a large extent, the absence of an item in the response to multiple-choice questions (and in some others as well) resulted in a blank, rather than a zero, during data entry. We did not have the time to check across all response sets whether the respondent (the community group) had not answered the question (blank as missing) or had selected some item(s) (with blanks for the non-selected ones, to be treated as zeros). In order to obtain valid item means over respondents, and for purely pedagogical purposes, we assume that there were no questions unanswered, and thus no missing values. On this premise, we flushed all blanks with zeros. It is obvious that from a data management viewpoint this is not a legal operation in all circumstances, but appears defensible in ours.

In real-life situations, the data entry must maintain the distinction between item non-response (all blanks or some missing-value text code) and options not selected (zero, or "no" eventually converted to zero). To illustrate the treatment of missing values, the companion workbook on analysis demonstrates the handling of multiple-choice data with missing, with a small simulated dataset.

The demonstration data is in sheet "InitialTable", which is a segment of the original variables. The full dataset is reproduced for interested readers in sheet "CompleteBangladeshData", at the rightmost sheets tab in the workbook.

4. Steps in data preparation

We proceed by first introducing precautions at the workbook level. Then we address challenges with records and variables, including the rapid production of minimal descriptive statistics on all variables. We deal with the problem of blanks that can mean either missing values or negative response. Editing the non-missing values is the next step. Finally we take the data table into a shape that is suited for analysis and reporting, rather than tied to the language of the questionnaire. Auxiliary tables are given for export of the data to statistical programs (in this case STATA) and to document all named ranges used in the demo workbook.

This flowchart attached to the summary shows the correspondence between the successive steps of the editing and formatting process and the sheets of the demo workbook. However, as we underline at several points of this note, there are more sheets than the process in real life would require. The analyst would typically perform several consecutive steps in one sheet and continue with others using copies only if he considered the next step particularly risky.

4.1 Backup and file name conventions

Before any changes are made to any worksheets, workbook-level precautions are in order.

What

Acquire the data, secure it, make its current stage easily understood. Set rules and obtain commitment from team members.

How

- Save the original dataset in one or several backup locations. Save subsequent work versions with file names staring in the yymmdd-date and 24-hour time stamp, followed by the initials of the creator and a substantive caption, as in

  [network drive]:\DataOriginal\130105_0915AB_Floods_RawData_3Regions
  [network drive]:\DataProcessed\130105_1000AB_Floods_3Regions_Combined
  c:\DataProcessed\130105_1001AB_Floods_3Regions_Cleaning

[network drive]:\DataProcessed\130105_1100AB_Floods_3Regions_Cleaned
etc.

- Back up important working files daily to at least one location outside the main working place.

Why

Backups of the data files should be made, not only of the raw data, but at successive points of collating, cleaning and analyzing. Appropriate procedures will vary with the local context.

The risk of inadvertent overwrites of newer versions (that hold more records, or have newer formatting, or more calculated variables) with older ones is ever present. It looms in individuals, offices and teams with several members successively or concurrently working with the same data. Saving documents with file names that combine date and time of creation, the initials of the creator, and a short substantive description reduces the risk of confusion and loss of work. With yy.mm.dd-time prefixed, files will sort in the order of creation, as in

130105_1100AB_Floods_3Regions_Cleaned
130105_1145GIS_Floods_pcodes_added
130106_0830AB_Floods_with_Codebook
etc.

4.2 Records and variables

4.2.1 Record identifiers

What

Reviewing record identifiers and, if still needed, creating them should be the first of the operations that make any changes inside the sheet with the data table.

The records of the data table need unique identifiers. Depending on context and purpose, several identifying variables may be needed, including one that allows for the initial sort order to be re-created if necessary.

How

- If a consecutive numeric record numbering as the initial sort order does not yet exist, create one in an inserted column to the left.

- Leave other existing identifiers as they are. If additional identifiers - e.g. "p-codes", corresponding record identifiers from secondary data tables - are likely to become available soon, insert columns to host them.

[In the demo dataset, we have only one, "New number", in column 1.]

Why

Unique record identifiers are helpful as link fields, controls of duplicates, and easy reference in communicating about specific data points. Identifiers with an (alpha-) numeric order permit recreating the initial sort order any time later, which is particularly helpful if errors of data entry or processing turn up later and need to be traced back.

If the data are entered in several places (participating agency field offices; computers in the central office), a record identifier needs to be included for each entry station to create unique identifiers locally. The identifier may be identical with the one written on the hardcopy data carrier (normally the questionnaire). In addition, a running number, prefixed with a symbol for the entry station, should be created in the sequence of record entry so that in each of the local data tables the original sort order can always be recreated.

The central office then appends the several contributing data tables. Appending in Excel is error-prone, particularly if some entry operators changed the order of fields. Therefore, at the time of appending tables, a unique identifier for all records of the combined table should be added. Naturally, a running number 1, 2 … N serves the purpose of keeping the original sort order reproducible.

Table 2: Record identifiers after appending regional data tables

Unique record number AFTER APPENDING	Running number BY REGIONAL DATA ENTRY OPERATOR	PLACEHOLDER for p-Codes	Questionnaire number ON HARDCOPY	Team number	District	Upazila
1	1		T10-Jess-Abhy-001	10	Jessore	Abhynagar
2	2		T10-Jess-Abhy-002	10	Jessore	Abhynagar
3	3		T10-Jess-Abhy-003	10	Jessore	Abhynagar
4	4		T10-Jess-Abhy-004	10	Jessore	Abhynagar
5	5		T10-Jess-Abhy-005	10	Jessore	Abhynagar
etc. with records for Jessore District						
25	1		T07-Kuln-Paik-001	7	Kulna	Paikgacha
26	2		T07-Kuln-Paik-002	7	Kulna	Paikgacha
27	3		T07-Kuln-Paik-003	7	Kulna	Paikgacha
28	4		T07-Kuln-Paik-004	7	Kulna	Paikgacha
etc. with records for Kulna District						

Note: Fictitious example

4.2.2 Marking and documenting variables

What

Make data elements conveniently accessible for rapid documentation and descriptive statistics of all variables. Highlight variables that belong together, using a suitable cell background color scheme, and document them in a separate sheet.

How

If you are in A1 reference style, switch (forever!) to R1C1 style (Excel Options - Formulas - Working with formulas - check R1C1 reference style. (Note: "R" stands for "row", "C" for "column" in the English edition. French language users will find "L1C1", where "L" stands for "ligne").

Below row 1, color the columns of variables that substantively belong together, by a color scheme of your choice.
[We have chosen to color by type of variable, but this is a matter of purpose and efficiency]

Table 3: Color codes for types of variables

⬜	Identification variable
🟫	"Before - after" pair of categorical variables
🟦	Binary indicator
🟧	Grouping variable
🟩	Multiple-choice data
🟦	Ordered categorical indicator
🟥	Priority data

To make it easier for readers, we inserted a row below row 1 to note these variable types again in words. This is for purely didactic reasons; in real life one will not bother.]

Create short names for all variables:

In sheet "Stage01_Data", insert a row below row 4.

Create a name that Excel (as well as any statistical program) can read, starting with a letter and underscore, followed by a three-digit number equal to the current column number, as in "a_001", "a_002", etc.

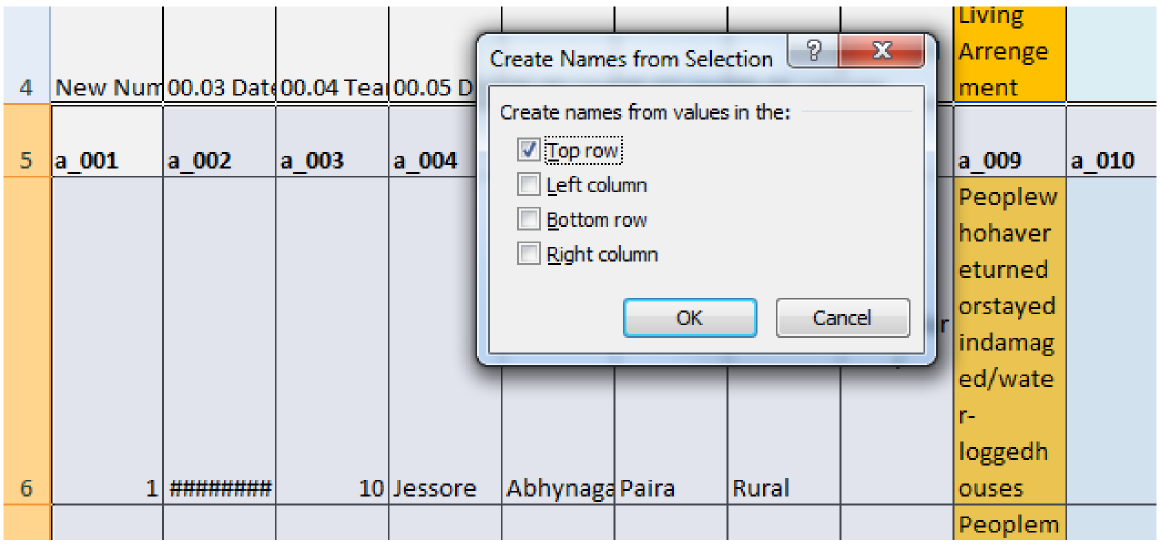
Write the first two, "a_001" and "a_002", then copy them rightward, by selecting them and dragging the bottom right cross-shaped handle, up to "a_119"[4].

Name each of the data columns after its short variable name, i.e. R5C1:R68C1 as "a_001", etc.:

The naming operation is best done for all data columns at the same time, by selecting the row of short variable names together with the data range, i.e. by selecting R5C1:R68C119, then from the menu: *Formulas - Defined names - Create from selection - check Top row only.*

Figure 2: Create names from selection

[4] Alternatively, we can create these names semi-automatically, by writing in cell R5C1 the formula *=CONCATENATE("a_", TEXT(COLUMN(RC), "000"))*, then copying it rightward and replacing the formulas with their values. This is less efficient here, but in other contexts of semi-automatic string creation the function CONCATENATE is superior.



4	New Num	00.03 Date	00.04 Tea	00.05 D					Living Arrengement	
5	a_001	a_002	a_003	a_004					a_009	a_010
6	1	#########	10	Jessore	Abhynaga	Paira	Rural		Peoplewhohaveeturnedorstayedindamaged/water-loggedhouses	
									Peoplem	

Create Names from Selection

Create names from values in the:

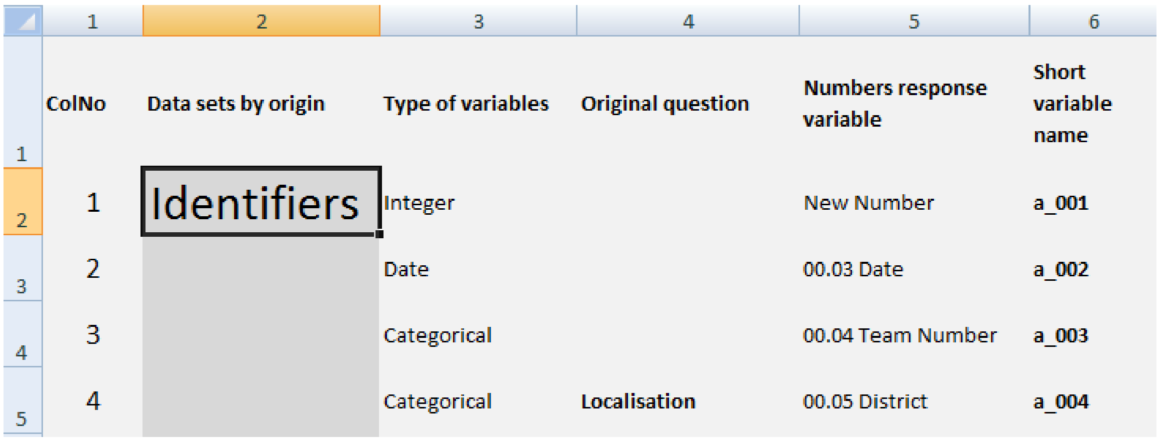
- [✓ Top row]
- Left column
- Bottom row
- Right column

[to see the effect, click the drop-down arrow in the cell address box and click, e.g., a_001.]

Create the Variables sheet:

Insert a new sheet, name it "Stage01_Variables" and name the column headers in R1C1:R1C6 as shown in the screenshot below.
In "Stage01_Data", select the range R1C1:R5C119, copy and paste (transpose) into "Stage01_Variables", starting in R2C2 (the framed cell in the screenshot).
In column 1, pull down a consecutive numbering 1, 2... 119, corresponding to the column numbers in the data sheet.

Figure 3: Setting up the Variables sheet



	1	2	3	4	5	6
1	ColNo	Data sets by origin	Type of variables	Original question	Numbers response variable	Short variable name
2	1	Identifiers	Integer		New Number	a_001
3	2		Date		00.03 Date	a_002
4	3		Categorical		00.04 Team Number	a_003
5	4		Categorical	Localisation	00.05 District	a_004

Why

R1C1 style makes working with formulas and with mixed-style cell references easier.

Response to certain types of questions, e.g. multiple-choice, is typically recorded in several fields; marking these columns by the same color facilitates orientation and navigating within the sheet. Our choice, to mark by variable type, is didactic; in the real life of assessments, one may prefer a coloring scheme by sector.

Proper variable names are needed for several reasons: The questions are unsuitable as category header texts in later table output; short names are easy to see in Pivot table field name boxes; and short names, numbered consecutively, are intuitively suited as names of the underlying data column ranges.

In Excel data tables, referencing data columns by named ranges, ideally by their very bold-faced column headers, is highly advantageous. Excel then knows, for example, that

Name	**Refers to**
a_001	=Stage01_Data!R6C1:R68C1

These ranges can subsequently be referenced in formulas (for statistics, for re-coding or as criteria in IF-type functions of other variables), in macros, and in Pivot tables. Once the ranges are named, formulas that reference them do not have to be adjusted when they are copied or moved. Naming all column ranges in the data range by the header names is done with just one menu command.

Similarly, we create a Variables sheet for several connected reasons: to document the variables in an easy overview; to exploit Excel's defined-name feature to compute descriptive statistics; to note any special observations, special transformations or measurement units, etc. in a "Comment" column; to eventually replace the original question text with variable labels suitable in tables and reports.

In real life, do not bother with written text in "types of variables" - this is for demonstration only -; instead, include a column "Comment" in the Variables sheet.

[Sidebar:] Long-form to wide-form

As the reader may have noticed by now, the demo dataset holds information from three sources - team's assessments, men's groups and women's groups. Originally, these were in separate tables. They were merged on the record identifier "New Number". As a result, all the information on a given assessed community now resides in one record, i.e. in one Excel row.

This begs the question of what to do if at data entry, instead of creating separate tables, the three records for each site had been entered in the same table, on top of each other. Or, similarly inconveniently, if they had been entered in the same table, each as a compact set, then seamlessly stacked one upon the other. This would have been unlikely here because the questionnaires (and data entry forms) for the three groups were different.

It has happened in other contexts. This is known as *long-form data*. The situation is aggravated if no common identifiers are used for the records of the different groups pertaining to the same site, and even more so if the respondents identified themselves by different geographic references (e.g. men and women in the same site being recorded by the names of different villages). Obviously, a common identifier is required for each site, and in addition an indicator variable to specify the type of respondent / group for each record. This can be done on the strength of local knowledge only. In other words, it cannot be fully automated.

Assuming the identifier issues have been resolved, we sort on the group indicator, then the common identifier. We copy and paste the set of records, for one group after the other, into a new sheet, *side by side*. We inspect whether they correctly match, visually at least for a small sample, with a suitable IF-based temporary indicator for large samples (e.g., = IF([identifierA= identifierB,0,1). If the sum of the indicator values down the column > 0, then there are mismatches that need to be addressed. Once satisfied with the matching, we delete redundant multiples of the same ID and area variables. The data now is in *wide form*.

If the reconciliation by sites is incomplete, the merging cannot proceed by simple cut-and-paste. We separate the record sets into separate sheets, name the data ranges, create an ID column in the common destination sheet for all ever used site IDs, and import the data, using the function VLOOKUP, into the horizontally adjacent areas. Unmatchable records are to be treated as additional sites unless one has reason to exclude certain records from the analysis.

4.2.3 Descriptive statistics

What

Inspect the data rapidly for key differences of interest:

Numeric or text type variable - as planned?
Text entries - where only numeric expected?
Blanks anywhere? - in the context of missing vs. genuine zero, "no", "not present", etc.
Range and averages (mean, median), outliers

so that "last chances" to cross-check with paper questionnaires and/or with colleagues knowing the context of an apparent anomaly can be seized before advancing further in processing and analysis.

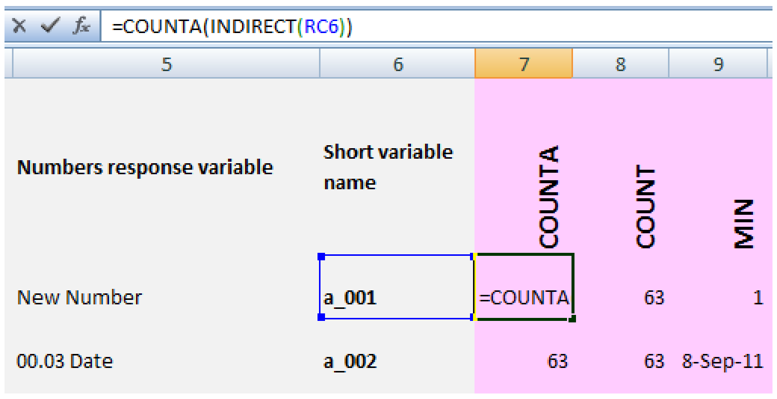
How

Create a space in the Variables sheet, like the purple area in sheet Stage01_Variables.
Fill column headers with the names of function or statistics to be computed.
In row 2, create corresponding formulas. The demo workbook uses:

Address	**Formula**
R2C7	=COUNTA(INDIRECT(RC6))
R2C8	=COUNT(INDIRECT(RC6))
R2C9	=MIN(INDIRECT(RC6))
R2C10	=MEDIAN(INDIRECT(RC6))
R2C11	=AVERAGE(INDIRECT(RC6))
R2C12	=IF(RC[1]=0,"",SUM(INDIRECT(RC6)) / 63)
R2C13	=MAX(INDIRECT(RC6))
R2C14	=SUM(INDIRECT(RC6))

where the mixed-type reference RC6 points to the intersection of the same row with column 6, which is where the variable names are listed. INDIRECT tells Excel to understand the content of that cell as the name of a defined range, as in this example.

Figure 4: The function INDIRECT referencing a named range



=COUNTA(INDIRECT(RC6))

5	6	7	8	9
Numbers response variable	Short variable name	COUNTA	COUNT	MIN
New Number	a_001	=COUNTA	63	1
00.03 Date	a_002	63	63	8-Sep-11

- Copy the formulas down to the bottom of the table.
- Excel reports errors in cells for which the statistics could not be computed (e.g., because they are text), or if the data themselves had Excel-defined errors. For tidiness, select any cells with errors (F5 - Special - Formulas - Errors) and delete their contents.

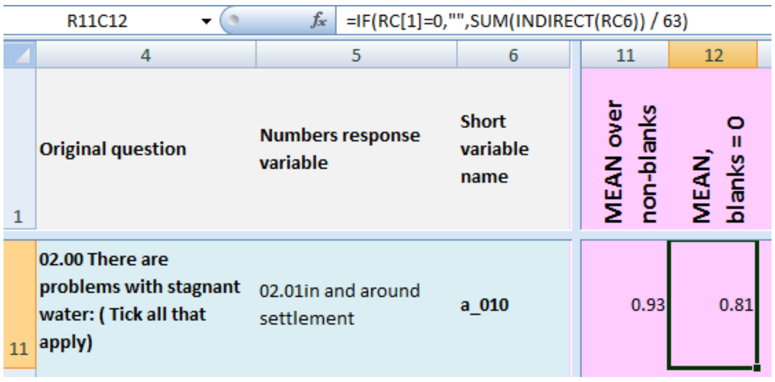
Why

We need a procedure that produces the statistics rapidly. The table exploits the power of defined names to look up data and of the function INDIRECT to reference the named ranges for functions like COUNTA (count of all non-blank cells), COUNT (count of all numeric cells), etc[5].

The table gives a quick overview of the missing value situation (there are missing if COUNTA < 63 in the demo sheet), numeric or non-numeric variable type (if the latter: COUNT = 0), the range (MIN to MAX), central tendency (MEDIAN and mean) and the difference in means depending on whether blanks are considered missing or zero. SUM is routinely included, but has no immediate meaning for this dataset. If there are many continuous variables, one will want to give also the standard deviation (STDEV in Excel), which we have not done for this dataset.

Note, for example, the difference in the fraction of sites observed with stagnant water in and around settlements, depending on whether blanks are taken to be missing values (0.93), or blanks mean "not the case" and should be replaced by zeros (0.81).

Figure 5: Example of inspecting descriptive statistics



```
R11C12    =IF(RC[1]=0,"",SUM(INDIRECT(RC6)) / 63)
```

	4	5	6	11	12
1	Original question	Numbers response variable	Short variable name	MEAN over non-blanks	MEAN, blanks = 0
11	02.00 There are problems with stagnant water: (Tick all that apply)	02.01in and around settlement	a_010	0.93	0.81

If blanks mean "not the case", then 0.93 is biased upward, and 0.81 is the true sample value. If the blanks represent genuine missing observations, then the pattern of sites with missing may be of interest.

Conveniently, if after inspection some of the data are changed, the statistics will update automatically.

Later, during the analysis and report-writing phases, the descriptive statistics readily answers a number of specific questions, for which otherwise Pivot tables or formulas would have to be run ad-hoc.

[5] See the Excel Help for the various uses of INDIRECT.

4.2.4 Clarifying missing values

What

The data are inspected for missing values (blanks, explicit missing-value codes). Blanks cells that represent genuine negative observations ("no", "not present", "option not taken", etc.), are replaced with zeros.

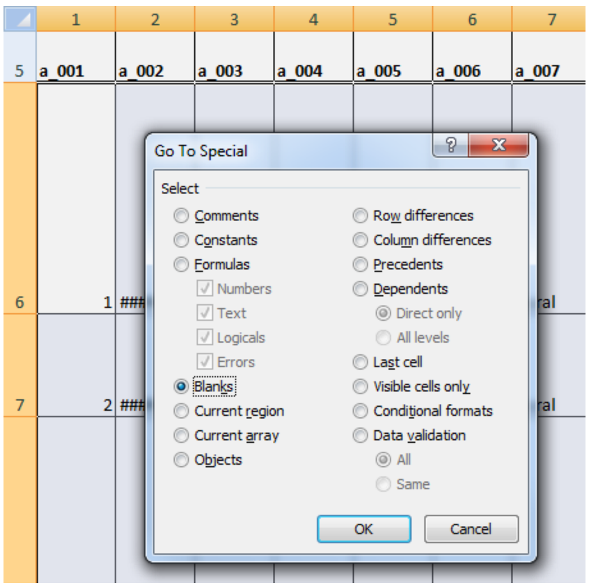
[In the Bangladesh data, we treat all blanks as genuine negative observations, to be replaced by zeros.]

How

Select the segment of the data range in which replacements are to take place (= in our case, the entire data range R6C1: R68C119).

- Use F5 - Special - Blanks to select all blanks in the range.
- With the usual Crtl+H replacement command, replace blanks with 0.

Figure 6: Selecting all blank cells within a range



Select: [✓ Blanks]

[We present the result, for didactic reasons, in new worksheets, in Stage02_Data and Stage02_Variables. To avoid confusion in the definitions of named ranges, the prefix of the short variable names was changed from "a_" to "b_". In "real life" there is no need for new sheets, or new prefixes.

Note that the descriptive statistics have updated. The values in columns 12 and 13 are now all equal (as a result of flushing blanks with zeros), except in a few rows, e.g. row 33 (because in variable b_032, two records are "don't know" - genuine missing!), 71 and 120.]

Why

The handling of missing values as well as the treatment of negative observations ("no", "absent", "not the case", etc.) and of genuine numeric zeros are frequent sources of error; cells for which the missing vs. negative-value status is doubtful should be inspected. In binary variables (yes / no), the positive value should be coded as 1, the negative as 0. Frequently, entry operators leave all or some of the negative observations in response to multiple-choice and preference data blank.

Under time pressure, the inspection can be split for greater efficiency:

- At this step, we inspect only the pattern of blanks and zeros in variables for which "MEAN over non-blanks: and "MEAN, blanks = 0" differ.

- Inspection for explicit missing value codes used (na for "not applicable", dk or dnk for "don't know", etc.) and for instances of "yes" or "no" not yet coded properly can wait until we have listed all values in all variables. We cover this further below.

At this step, a decision is taken which blank cells need to be filled with zeros (because they represent genuine negative observations), which to leave blank (if our convention is to use blanks for missing or not applicable), and which to replace with some explicit missing value code (if we want all missing to be explicitly coded).

In our Bangladesh data, however, we have no possibility to establish which blanks stand for genuine missing, and which for negative observations. We have reason to assume that the data are virtually complete, and that blanks do represent negative observations - chiefly options not chosen under multiple-choice and priority-type questions.

With these data and under these assumptions, it is correct to replace all blanks with zeros. In datasets for which these assumptions do not hold, replacement has to be selective.

4.3 Value inspection and recoding

What

We inspect the values in the variables with an intensity and thoroughness that may vary by purpose, work standard, and time available. We replace unhelpful entries (e.g. mis-spellings, verbose descriptions, etc.) with more suitable variants, in consistent manner ("recoding variables"). We create variants suitable for tables and graphs, with pre-fixes that force meaningful sort orders in Pivot tables for later editing.

How

Decide on one from among three procedures:

If quick visual inspection suffices to identify all variants to be replaced, replace them using the standard Ctrl+H replacement command. If the default (alpha-) numeric sort order is unsuited for later analysis and reporting, use prefixes to force a better order, as exemplified by this recoding table.

Figure 7: Example of a recoding table with prefixes

14.00 Assessment team overall ranking of the situation at this site: (tick one only)	
Alphabetical: **c_019_original**	From lowest to highest: **c_019_recoded**
Extremely vulnerable	4-Extremely vulnerable
Relatively Normal	1-Relatively normal
Seriously vulnerable	3-Seriously vulnerable
Seriouslyvulnerable	3-Seriously vulnerable
Vulnerable	2-Vulnerable

1. If the variants to be replaced, or their replacements, are not immediately obvious, but matter only in a small number of variables, make for each considered variable a Pivot table in order to list all variants, together with their frequencies. Decide which variants to replace, with what (perhaps combining some categories), and use Ctrl+H in the data table.

2. If we want to list the unique values for all variables and set up machinery for recoding that will automatically update if we decide to modify replacement categories, opt for the procedure described in the sidebar below.

The demo workbook presents the results of the third procedure, in Stage03_Data and Stage03_Variables. Two auxiliary sheets needed in this case, Stage03_UniqueValues and Stage03_RecodingArea, are also shown. In the variable names, we again change the prefix, to "c_", for didactic reasons and to avoid confusion in named ranges. In real life, this change would not be necessary.

The same results could have been obtained by the first or second approach, although with higher demands on the visual inspection of the entire data sheet, thus in slower and more error-prone ways.

Why

There are many reasons why we want to inspect the data in greater detail. Often we need to replace some or all of the recorded values in a variable with other values that we find more appropriate for our purpose, an operation that is a form of recoding. Reasons for recoding are: spelling corrections, date (day, month, year) formatting, translation, language style and simplification, pre-fixes to create better sorting in tables, combination (in categorical variables), rounding (in continuous variables), and possibly others. Several are obvious in this example.

Table 4: Recoding of verbose entries

00.10 Living Arrangement	
Alphabetically sorted, verbose **c_009_original**	New sort order, succinct **c_009_recoded**
People displaced and staying in collective centre	1-Collective centers
People displaced and staying on road sides, embankments and other high ground	2-Roadside / embankments
People marooned in their homes	3-Marooned
People not marooned, displaced or whose houses have been damaged (visually unaffected)	5-Home undamaged
People who have returned or stayed in damaged/water-logged houses	4-Damaged or water-logged
Peoplemaroonedintheirhomes	3-Marooned
Peoplewhohavereturnedorstayedindamaged/water-loggedhouses	4-Damaged or water-logged

[Sidebar:] Listing the unique values of all variables

In small data tables, the inspection of values within variables can proceed visually, with values replaced by the simple replace (Crtl+H) command, in the same variable, or in a copy named differently if the original variable is to be preserved as is. However, the analyst will soon be overwhelmed if the sample or the number of variables is large. At this point, a somewhat strategic decision is needed as to whether a simple, selective inspection and recoding mode is sufficient, or whether all variables need to be processed.

If we decide to list the unique values of all variables, a semi-automated procedure is needed[6]. It relies on the use of user-written functions and macros that are included with the demonstration workbook. This approach is suitable for Excel users with minimal knowledge of how to place code in a VBA module and how to insert functions into cells and to run macros. The user is not required to write any code, just to copy (from the demo workbook), paste (into a VBA module of his macro-enabled workbook), and use the code.

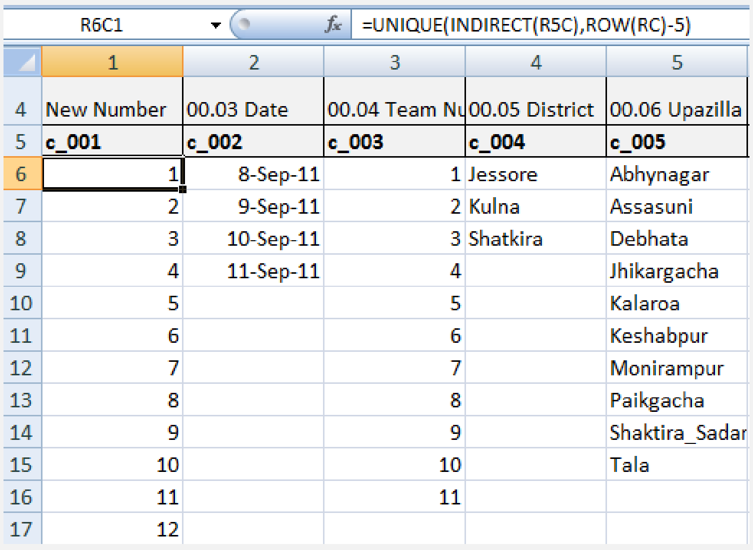
In the demo workbook we execute the procedure as follows:

Sheets Stage03_Variables, Stage03_Data
We create sheets "Stage03_Variables" and "Stage03_Data" as copies of their Stage02 namesakes. To avoid confusion in named ranges, in row 5 of the data sheet and column 6 of the variables sheet we replace the "b_" prefix with "c_". In the data sheet, we create named ranges for all data columns, using the same command as described under "Sheet Stage01_Data". We revert to these two sheets further below.

Sheet Stage03_UniqueValues
We name a new sheet "Stage03_UniqueValues" and paste into it the first five rows copied from "Stage03_Data". In rows 6 and downwards, we create for each variable its list of unique values, following the steps described in the textbox in the sheet. The two sorting and cleaning steps should take place in a temporary sheet, which we may delete after copying the final result back into "Stage03_UniqueValues".

[6] It would be more correct to speak of the set of all *distinct* values used in a variable. However, the Excel user community seems to prefer the "unique" terminology for a value that is distinct from others although it may occur multiple times in the same column.

Figure 8: The user-written function UNIQUE



R6C1 =UNIQUE(INDIRECT(R5C),ROW(RC)-5)

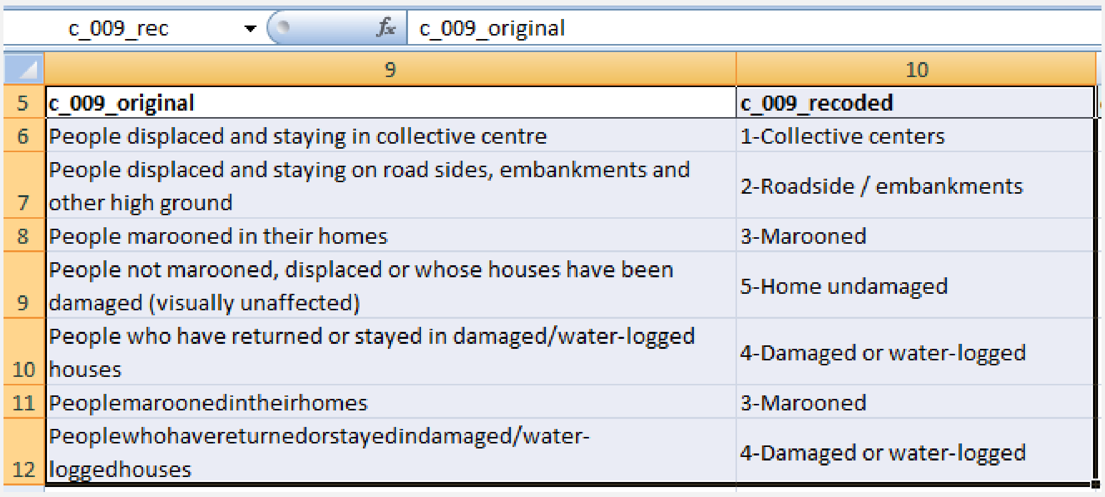
	1	2	3	4	5
4	New Number	00.03 Date	00.04 Team Nu	00.05 District	00.06 Upazilla
5	c_001	c_002	c_003	c_004	c_005
6	1	8-Sep-11	1	Jessore	Abhynagar
7	2	9-Sep-11	2	Kulna	Assasuni
8	3	10-Sep-11	3	Shatkira	Debhata
9	4	11-Sep-11	4		Jhikargacha
10	5		5		Kalaroa
11	6		6		Keshabpur
12	7		7		Monirampur
13	8		8		Paikgacha
14	9		9		Shaktira_Sadar
15	10		10		Tala
16	11		11		
17	12				

Sheet Stage03_RecodingArea

We make a copy of "Stage03_UniqueValues" and name it "Stage03_RecodingArea". This is the sheet in which we inspect the sets of unique values rendered for each variable and create recoding tables, as opportune.

By way of example, see column 9 in the screenshot below. This is the same set of unique values as in Table 4 above. If we decide to replace them, we insert a column to the right (which becomes column 10) in this sheet as well as in "Stage03_Data". In this column in "Stage03_RecodingArea", we write the replacement values (e.g., "1-Collective centers" to replace "People displaced and staying in collective centre", with the numeric prefix to force the sort order in any table based on the recoded variable). We change the headers in row 5 to "c_009_original" and "c_009_recoded" for clarity's sake. We select R5C9:R12C10 and name it "c_009_rec" as the concerned recoding table.

Figure 9: Defining a recoding table



c_009_rec c_009_original

	9	10
5	c_009_original	c_009_recoded
6	People displaced and staying in collective centre	1-Collective centers
7	People displaced and staying on road sides, embankments and other high ground	2-Roadside / embankments
8	People marooned in their homes	3-Marooned
9	People not marooned, displaced or whose houses have been damaged (visually unaffected)	5-Home undamaged
10	People who have returned or stayed in damaged/water-logged houses	4-Damaged or water-logged
11	Peoplemaroonedintheirhomes	3-Marooned
12	Peoplewhohavereturnedorstayedindamaged/water-loggedhouses	4-Damaged or water-logged

Back to "Stage03_Data", we name the new variable in column 10 "c_009_recoded" and calculate it with the lookup formula

=VLOOKUP(RC[-1], c_009_rec, 2, FALSE),

where RC[-1] points to the lookup value in the next column to the left, c_009_rec references the relevant recoding table, 2 instructs Excel to return the value in the table's second column, and FALSE requires an exact match[7].

We immediately document this recoding operation in "Stage03_Variables", by inserting a row below the original row for "c_009" and by adjusting the numbering of variables in column 1. The descriptive statistical formulas for these two renamed variables no longer work, but this is a detail to be fixed later. Right now we wish to keep abreast with the changes in variable names and column numbers in the data table.

We reiterate the operation of recoding, variable by variable, inspecting their unique value sets in "Stage03_RecodingArea", creating recoding tables for those variables to be recoded, recoding them in "Stage03_Data" using VLOOKUP, and documenting the changes in "Stage03_Variables". Variables that only need trivial changes (e.g. "don't know" to blank, as in column 34 of "Stage03_RecodingArea") should be fixed by simple "Ctrl+H" replacements.

Result: Stage03_Variables, Stage03_Data

At the end of this stage, the data table has all variables that needed recoding, in both the original and the recoded versions. The lookup formulas are still there. The changes are also highlighted in the variables table. (We will not waste time fixing the statistical formulas for the recoded variables; in the next step we will delete the original versions and retain the recoded ones.)

Sheets Stage04_Variables, Stage04_Data

In the following step, we get rid of the formulas, by replacing them with their values, and subsequently of the original variables that have been recoded.

However, to keep the recoding process reproducible, we advise to keep the Stage03 sheets and work strictly with copies of the data and variables tables.

We begin in sheet Stage04_Data. We collectively replace all formulas with their values. We delete the variables with the suffix "_original". In the recoded variables, we chop off the suffix "_recoded". As before, in order to avoid confusion in named ranges, we change the prefix of the variable names. This time we go from "c_" to "d_". We create named ranges of the data columns, in the manner as before. We modify Stage04_Variables accordingly so that all the descriptive statistics appear.

4.4 Final shape of the data table

What

In the data table we get rid of the questions, replacing them with meaningful labels. If there is an interest in preserving the original question numbering (for easy referencing), we create new short variable names incorporating those numbers. We take the data table to its final shape and document it accordingly.

[7] For a generic explanation of the syntax of VLOOKUP, consult the Excel Help.

How

At first, work in sheet Stage04_Data, inserting two rows below row 4.
In row 5, extract the number of the question / response option and give it a prefix. Do this, using the formula like

= CONCATENATE("q_", LEFT(R[-1]C, 5))

in R5C1:R5C119. Replace these formulas by their values, and replace the remaining period with an underscore. As a result, we get symbol strings from "q_00_01" all the way to " q_37_07".
In row 6, create meaningful, orthographically correct variable labels, such as replacing the verbose question *"04.00 Are there vectors evident where people are staying (mosquitoes, rats etc)"* [column 15] with the succinct *"Disease vectors"* - the kind of text that we want to appear in tables and charts.

[As before, we have given the short variable names a different prefix, this time "d_", to avoid confusion in named ranges. This hardly matters any more - these variable names are going to be replaced by our new "q_xx_xx" series.]

- Then give the data table its final shape, in sheet Stage05_Data:
  - Delete the two rows holding the questions and response options.
  - Move the "q_" variable names to row 4, bold-face them. Optionally, replace the prefix "q" with more specific prefixes, such as "i" for the identifier variables, "t" for team observation variables, etc., which can be helpful in Pivot table manipulations.
  - Optionally, keep the old "d_" variable names in row 3 (not very useful).
  - Improve the cosmetics, such as by boldfacing the variable labels in row 2, setting a uniform column width, and reducing the cell background color diversity.
  - Select the range R4C1:R67C119, name the data column ranges collectively and name the entire range "Database"
- In sheet Stage05_Variables, reflect the changes, as necessary. The formulas in the descriptive statistics area now all refer to column 5 (the variable names). Select the pink range and replace the element "RC6" by "RC5" in all cells.
- Optionally, for historical reference, to the right of the statistics area, transpose the original question wording and response options.

Why

By stage 4, our data table has values that are edited, but the variable names are still messy and named after the questionnaire, rather than in the terms that should appear in statistical tables and charts. The final formatting needs to take care of this.

Moreover, we anticipate a problem with the Pivot table field list. The wizard has narrow boxes into which fields are dragged. The list area can be stretched to the left, but only at the expense of compressing the rest of the screen. Also, for our rapid descriptive statistics, we wish to maintain short, compact names for the data column ranges.

We therefore use a distinction, commonly made in statistics program, between *variable names* and *variable labels*. Variable names are short, without blanks and without special characters (except underscore). They should express either the essential or literal meaning of a variable ("subdist" for "Sub-district") or some ordering system for easy navigation (e.g. the column number) or special analytic intent. Names that incorporate the earlier questionnaire numbering are a useful type, too. Labels are meaningful, of manageable length for tables and charts, and exactly as they shall appear there.

As before, for descriptive statistics, we perform the range-naming operation. We name the entire data area (including the variable names) "Database" because we intend to pass this name to future Pivot tables.

Our data preparation is now complete. The various data and variables sheets demonstrated its essential stages. At the end, Stage05_Data and Stage05_Variables will survive to the analysis phase.

5. Auxiliary information

For convenience, sheet "Stage05_Variables" in column 14 offers STATA commands that can be gathered into a do-file editor should a STATA user wish to name the variables with these labels. Columns 15 and 16 hold the original question and response option wording, for historic reference.

The sheet "TransferToSTATA" holds the data in a format that the application can import.

Finally, the sheet "Named_Ranges" lists all the named ranges and their definitions in this workbook. This large number of names was created because, for demonstration purposes, we present subsequent stages of the data preparation in different worksheets, each time creating new variable names and new ranges named after them. In the real world, the process would not be condensed into one sheet, but certainly into fewer than in this workbook.

[Sidebar:] Alternating between Excel and STATA

We have demonstrated the data preparation within an Excel workbook. There is no need to supplement those steps with any performed in a statistical program such as SPSS, R or STATA. Nevertheless, the analyst may resort to a statistical program for additional data preparation features that may be desirable in certain circumstances. Some may be produced faster and more conveniently than in Excel. Others may anticipate elements of basic analysis, such in quick cross-tabulations to check the sample structure, before the work in Excel is pushed too far ahead.

By way of example, we mention some of the data management procedures that STATA offers. It may be productive to create a STATA data table early on, such as from the range R5C1: R68C119 in the sheet Stage01_Data. We may be interested to have a list of sites that are not sufficiently distinguished on the basis of geographic names - this means, in terms of the Bangladesh dataset, sites that are duplicates in terms of district, subdistrict, Union and village names. This can be done in Excel through a custom sort and visual inspection. STATA's commands ***distinct [varlist], joint*** will immediately let us know whether such duplicates exist, and ***duplicates list [varlist]*** will produce a list of them. Without bothering, at this stage, to label the variables in STATA, we obtain:

Table 5: STATA sample output - list of nominal duplicate sites

```
. distinct a_004 a_005 a_006 a_008, joint

        Observations
      total   distinct
         63         60

. duplicates list a_004 a_005 a_006 a_008, sep(0)

Duplicates in terms of a_004 a_005 a_006 a_008

  +----------------------------------------------------------------------+
  | group:   obs:      a_004            a_005          a_006       a_008 |
  |----------------------------------------------------------------------|
  |      1      6    Jessore      Jhikargacha          Bakra     Dikdana |
  |      1      7    Jessore      Jhikargacha          Bakra     Dikdana |
  |      2     32   Shatkira         Assasuni      Kadakhati    Kadakati |
  |      2     33   Shatkira         Assasuni      Kadakhati    Kadakati |
  |      3     51   Shatkira   Shaktira_Sadar   Municipality   6 No Ward |
  |      3     52   Shatkira   Shaktira_Sadar   Municipality   6 No Ward |
  +----------------------------------------------------------------------+
```

which will prompt the analyst to look for other distinguishing attributes in order to establish that every observation in these pairs of duplicates represents a genuine site.

This example apart, STATA users can think of a variety of data management commands that come in handy, depending on situations. ***codebook*** describes variables; ***generate*** and ***egen*** (extensions to generate) create new variables (particularly ***tag*** and ***group*** can be very helpful to manage more complex data structures); ***encode*** converts string variables to numeric ones, which are needed in certain analysis commands; ***sencode*** (super-encode) does the same, yet gives the user far-reaching control of the sort order in which the values of the variable will be encoded (in Excel, we achieve controlled sort orders through numeric prefixes). ***append*** should be tried for appending datasets when regional data entry operators messed with the order of variables; ***reshape*** commutes tables from long to wide shape and vice versa; ***renvars*** is a flexible variable-renaming command[8]. The list is, of course, incomplete.

In all that, keep in mind that the philosophies of Excel and STATA have their own emphases. Excel is much closer to the dirty work of data cleaning and table reformatting. STATA presses for analysis; it does not like verbose cell entries, with commas and other special symbols. One would not normally want to do much data preparation in STATA unless he had in mind some subsequent analysis work for it as well. As we indicated, there may, however, be situations in which early work in STATA is productive - either in parallel or by re-exporting to Excel intermediary results (e.g. tagging variables) produced with STATA routines.

The Excel worksheet ready for export to STATA, "TransferToSTATA", was derived from the final stage of the data preparation process, in other words: from sheet "Stage05_Data". Column 14 of "Stage05_Variables" holds the commands that label the variables in STATA; they can be copied all at once and pasted into the STATA do-file editor. Such ***label*** commands can conveniently be manufactured using the Excel function CONCATENATE, with a temporary substitute for the double quotes (e.g., temporarily use xxx for "; then, after replacing formulas with their values, replace all xxx's with" using the replace (Ctrl+H) command).

[8] ***distinct, sencode*** and ***renvars*** are user-written commands that you may need to download, e.g. by typing ***findit distinct.***

6. Conclusion

What is called data management or data preparation here essentially boils down to editing and formatting operations. In this process, the conversational logic of the questionnaire and the transfer logic of data entry give way to the logic of documentation, analysis and ultimately reporting. In slow-moving surveys, pre-tested templates and built-in validation features take care of many of the steps to which, in fast-moving assessments without that luxury, the analyst himself must pay close attention.

The challenges vary from situation to situation, but some are more common than others. Some of the typical ones have been highlighted in this note, and the resolution of a few has been demonstrated with data from a real assessment. Data management in spreadsheets, particularly with Excel, gives much flexibility and freedom. The price of freedom is eternal vigilance, which, to our pleasant surprise, can be entrusted, in part, to the devices of the very same application. Ultimately, however, it is the skill and creativity of the team that determine how fast and how effectively the assessment transits from data entry to analysis and reporting, and how efficiently these will advance on the foundations of well-prepared data.