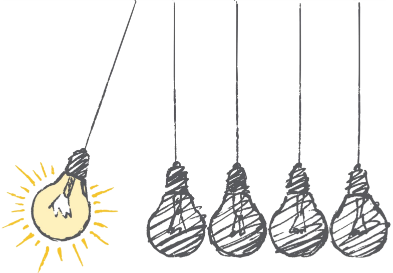
Data Cleaning



DIME RA Onbarding Course

February 13, 2020

Development Impact Evaluation (DIME)
The World Bank

Introduction

Data Work task breakdown

- We divide the data work process into four stages:
  1. De-identification
  2. Data cleaning
  3. Variable construction
  4. Data analysis
- Each of these stages has well-defined inputs and outputs
- For each stage, there should be a code folder and a corresponding data set
- The names of codes, data sets and outputs for each stage should be consistent
- The code, data and outputs of each of these stages should go through at least one round of code review.

Overview

Introduction

De-identification

Data cleaning

Unique ID

Field correctoins

Creating an annotated data set

Documenting your data set

Other data cleaning tasks

File saving conventions

Appendix

Data collection and importing

- For this presentation, we will assume you've already collected and imported your data
- In practice, however, data cleaning starts before data collection is over
- The following data collection tasks that, when done properly, make data cleaning a lot easier:
  1. Survey programming
  2. Data quality monitoring
  3. Data import

De-identification

Input: the raw data

- Contains only information received directly from the field
- Raw data files should be stored in the raw data folder exactly as they were received
- Be mindful of how and where they are stored: they cannot be re-created and nearly always contain confidential data
- The raw data files should never be edited directly
- If the raw data contains confidential information, it must be encrypted
- Make sure to have a back up of the raw data (hopefully, you will never need to use it)

Output: the de-identified data

- Working version of the data set that can be shared within the research team without risk
- Contains only information received directly from the field
- Contains no direct identifiers
- Is not necessarily anonymized
- Typically, de-identification should not affect the usability of the data

Identifying direct identifiers

- The first thing you need to de-identify the data is a list of all direct identifiers in the data set
- Ideally, all potentially identifying variables are flagged during in the questionnaire design
- Useful tools:
  - JPAL's `PII scan`
  - World Bank's `sdcMicro`
  - DIME Analytics' `iecodebook`

Removing direct identifiers

Once you have a list of all direct identifiers, for each one of them, ask yourself (and the pre-analysis plan):

1. Will this variable be needed for the analysis?
    - If the answer is no, just drop it
    - Don't be afraid to drop too many variables: you can always them back (but you can't undo a data leak)
2. Can I encode or otherwise construct a variable that masks the PII?

Data cleaning

Introduction

During data cleaning, you will look carefully at each variable in your data set. The objectives of this process are

1. Making the data set easily usable and understandable
2. Documenting individual data points and patterns that may bias the analysis

Introduction

- Cleaning is probably the most time-consuming data work task, and you will be tempted to skip it
- However, this is the time when really get to know your data
- Explore your data set using tabulations, summaries, and descriptive plots
- Knowing your data set well will make it possible to do analysis
- Cleaning your data well will save you time

Output: the cleaned data set

- At the end of this process, you should have a data set that is essentially the same as the one you downloaded from the server
- The main difference is that the clean data set should be easier to understand for anyone that's opening it for the first time
- It should also be easily traced back to the survey instrument
- Typically, one cleaned data set will be created for each data source or survey instrument
- The cleaned, de-identified data set, plus the documentation to support it, are first data output of your project: a publishable data set

Output: documentation

A few pieces of documentation should accompany the cleaned data set:

- A variable dictionary, or codebook, listing details about the variables in the data set
- The instruments used to collect the data
- A complete record of any corrections made to the raw data, including careful explanation about the decision-making process involved
- A report documenting any additional irregularities and distributional patterns encountered in the data

Unique ID

Unique ID

The first thing you want to look for every single time you open a new data set for the first time is

1. Unit of observation
2. Uniquely and fully identifying ID variable

Before you separate the identifiable from the de-identified data, make sure you know how to cross both using the unique ID

Desirable properties of an ID variables

1. Uniquely identifying
2. Fully identifying
3. Anonymous
4. Constant within a project

How would you test if a variable is uniquely and fully identifying?

Unique ID

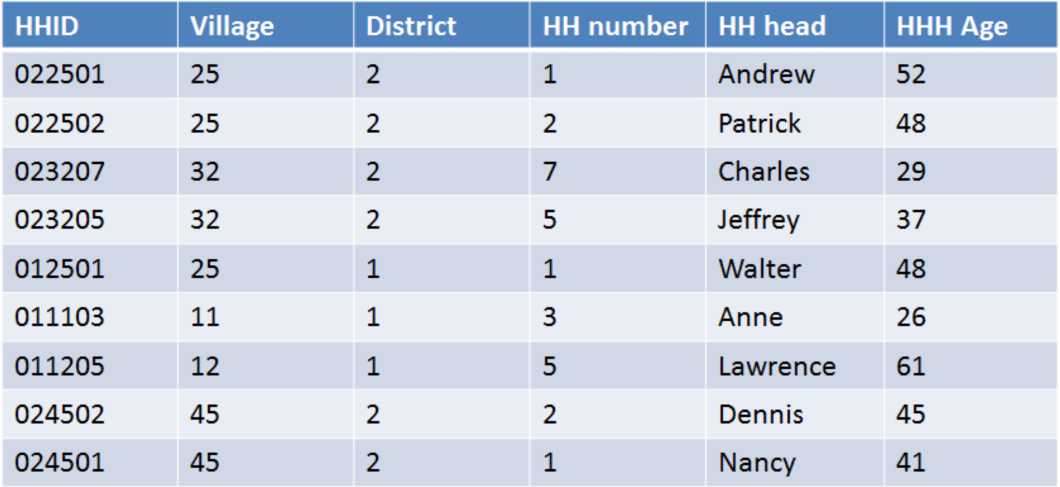
Commands for testing that the variable in Stata:

- `isid`
- `codebook`

Commands for testing that the variable in R:

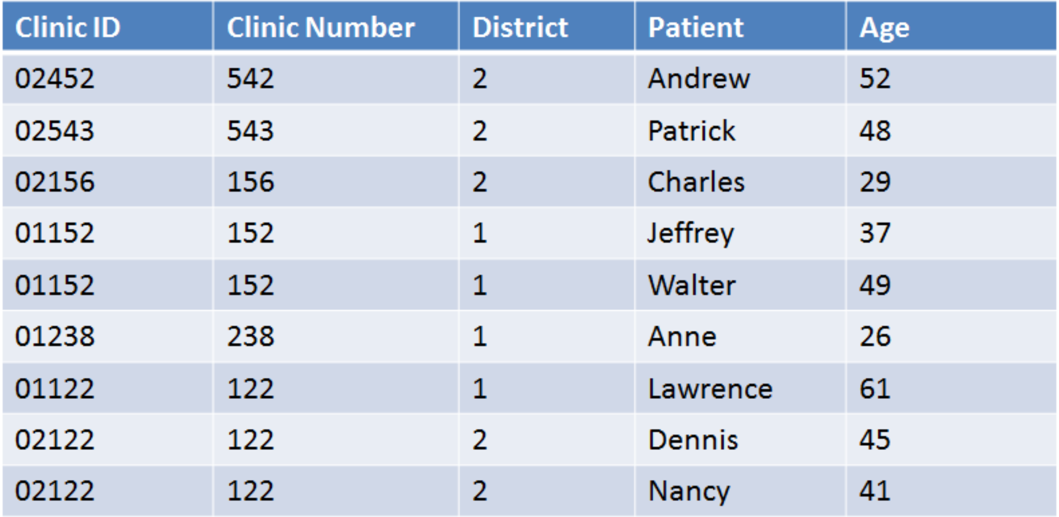
- `n_distinct()`
- `is.na()`
- `unique()`
- `length()`
- `dim()`

What is the unit of observation?



HHID	Village	District	HH number	HH head	HHH Age
022501	25	2	1	Andrew	52
022502	25	2	2	Patrick	48
023207	32	2	7	Charles	29
023205	32	2	5	Jeffrey	37
012501	25	1	1	Walter	48
011103	11	1	3	Anne	26
011205	12	1	5	Lawrence	61
024502	45	2	2	Dennis	45
024501	45	2	1	Nancy	41

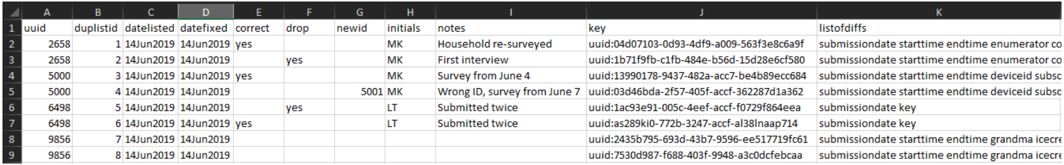
What is the unit of observation?



Clinic ID	Clinic Number	District	Patient	Age
02452	542	2	Andrew	52
02543	543	2	Patrick	48
02156	156	2	Charles	29
01152	152	1	Jeffrey	37
01152	152	1	Walter	49
01238	238	1	Anne	26
01122	122	1	Lawrence	61
02122	122	2	Dennis	45
02122	122	2	Nancy	41

Using `iefieldkit` to solve duplicated entries

- The `ieduplicates` command helps identify and resolve duplicates in raw survey data
- The command outputs a report of all the duplicated entries of a variable (in Excel), and removes the duplicates from the data set until they are resolved
- The Excel report is used to document the cases of duplicated interviews and how they were solved



	A	B	C	D	E	F	G	H	I	J	K
1	uuid	duplistid	datelisted	datefixed	correct	drop	newid	initials	notes	key	listofdiffs
2	2658	1	14Jun2019	14Jun2019	yes			MK	Household re-surveyed	uuid:04d07103-0d93-4df9-a009-563f3e8c6a9f	submissiondate starttime endtime enumerator co
3	2658	2	14Jun2019	14Jun2019		yes		MK	First interview	uuid:1b71f9fb-c1fb-484e-b56d-15d28e6cf580	submissiondate starttime endtime enumerator co
4	5000	3	14Jun2019	14Jun2019	yes			MK	Survey from June 4	uuid:13990178-9437-482a-acc7-be4b89ecc684	submissiondate starttime endtime deviceid subsc
5	5000	4	14Jun2019	14Jun2019			5001	MK	Wrong ID, survey from June 7	uuid:03d46bda-2f57-405f-accf-362287d1a362	submissiondate starttime endtime deviceid subsc
6	6498	5	14Jun2019	14Jun2019		yes		LT	Submitted twice	uuid:1ac93e91-005c-4eef-accf-f0729f864eea	submissiondate key
7	6498	6	14Jun2019	14Jun2019	yes			LT	Submitted twice	uuid:as289ki0-772b-3247-accf-al38lnaap714	submissiondate key
8	9856	7	14Jun2019	14Jun2019						uuid:2435b795-693d-43b7-9596-ee517719fc61	submissiondate starttime endtime grandma icecre
9	9856	8	14Jun2019	14Jun2019						uuid:7530d987-f688-403f-9948-a3c0dcfebcaa	submissiondate starttime endtime grandma icecre

Data entry corrections

Data entry corrections

- During data collection, particularly during primary data collection, it's likely that issues will be reported by the enumerators and supervisors
- During data quality monitoring, you will likely also identify problems that need to be addressed
- Examples of that are typos, incorrect IDs, and re-surveys
- It's important to record all these issues and the communications about them
- This is the only case, apart from duplicated IDs, when you will change data points during data cleaning
- Make all corrections in a script, not manually, and remember to document where the information is coming from

Creating an annotated data set

Label variables

When cleaning a data set, you should make sure that all variables are properly labeled, so that it is easy to understand what each variable represents:

- Check that all variables have variable labels (in English)
- Variable labels should explain what the variable is and, if that's the case, what unit it is in
- Labels cannot be longer than 80 characters

Encode variables

- The clean data set should contain no string variables, except for
  1. Proper nouns that are not categories
  2. Digits with leading zeros or long IDs (over 15 digits)
- That means categorical string variables must transformed into labeled variables or factors
- Be mindful of open-ended questions: they present a much higher risk of statistical disclosure
- Check that all categorical variables have value labels (in English)

Encoding variables in Stata

- In Stata, the best practice is to use `encode` with both the `label` and the `noextend` options

Example: `encode dist_name, generate(dist_id) label(district) noextend`

- Other useful commands: `label define`, `label value`, `label dir`, `label list`, `labelbook`
- If you used Survey CTO, used the column `label:stata` and the data was properly imported, this step may not be necessary

Encoding variables in R

- The `tidyverse` package `forcats` includes tools to deal with categorical variables
- Note that, unlike in Stata, in R you cannot choose the underlying numeric value of a factor variable
- This is not a problem, as you can refer to factors by their labels, and don't need to know what is the underlying numeric value
- R transforms strings into factor in alphabetical order, so remember to use `ordered` factors if you care about how categories are ordered

Extended missing values

- During primary data collection, use codes like -88, -9,-777 to represent different reasons for missing data such as “don’t know”, “declined to answer” etc
- These values need to be removed since they will otherwise bias the means
- If we change them all to missing, we will lose information
- Use extended missing values to keep the information but still tell Stata to treat them like missing

Extended missing values in Stata

- Regular missing value in Stata: **.**
- Use extended missing to represent the same reason for missing data across the project
  - .d = ”Don’t know”
  - .r = ”Refused to answer”
  - .s = ”Skipped”
- Missing values can also be labeled

Extended missing values in Stata

- To Stata, *numbers* $<$. $<$ *.a* $<$ *.b* $<$ *.c* $<$ ... $<$ *.z*
- So replace this

  ```
  sum HH_income if employment != .
  ```

  With this

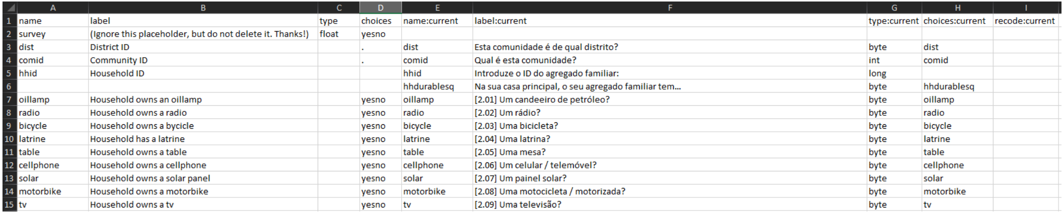
  ```
  sum HH_income if employment < .
  sum HH_income if !missing(employment)
  ```

Renaming variables

- Do not change the names of variables coming from a survey, even if you do not like the naming conventions used in the questionnaire
- Renaming variables will make it harder to find the correspondence between variables and survey questions
- There are two exceptions for this:
  1. **Identifying roster variable**: renaming harvest_1_1, harvest_2_1 to harvest_s1_c1, harvest_s2_c1 will make the variables' contents clearer
  2. **Roster number**: if a household cultivated several crops, but will only be asked about the 5 most important ones, then their codes will not correspond to the crop codes, but to their importance. So harvest_c1 means the harvested quantity of the most important crop, not of the crop whose code is 1. This may be changed to reflect the regular code for each crop

Using `iefieldkit` to annotate the data set

- The `iecodebook` command helps you perform most of the tasks describe above (with the exception of encoding)
- The command outputs (in Excel) a list of all variables in the data set and their labels, and applies changes to them so the process is simplified
- The Excel report is used to document the modifications made to the data set while cleaning



	A	B	C	D	E	F	G	H	I
1	name	label	type	choices	name:current	label:current	type:current	choices:current	recode:current
2	survey	(Ignore this placeholder, but do not delete it. Thanks!)	float	yesno					
3	dist	District ID		.	dist	Esta comunidade é de qual distrito?	byte	dist	
4	comid	Community ID		.	comid	Qual é esta comunidade?	int	comid	
5	hhid	Household ID			hhid	Introduze o ID do agregado familiar:	long		
6					hhdurablesq	Na sua casa principal, o seu agregado familiar tem...	byte	hhdurablesq	
7	oillamp	Household owns an oillamp		yesno	oillamp	[2.01] Um candeeiro de petróleo?	byte	oillamp	
8	radio	Household owns a radio		yesno	radio	[2.02] Um rádio?	byte	radio	
9	bicycle	Household owns a bycicle		yesno	bicycle	[2.03] Uma bicicleta?	byte	bicycle	
10	latrine	Household has a latrine		yesno	latrine	[2.04] Uma latrina?	byte	latrine	
11	table	Household owns a table		yesno	table	[2.05] Uma mesa?	byte	table	
12	cellphone	Household owns a cellphone		yesno	cellphone	[2.06] Um celular / telemóvel?	byte	cellphone	
13	solar	Household owns a solar panel		yesno	solar	[2.07] Um painel solar?	byte	solar	
14	motorbike	Household owns a motorbike		yesno	motorbike	[2.08] Uma motocicleta / motorizada?	byte	motorbike	
15	tv	Household owns a tv		yesno	tv	[2.09] Uma televisão?	byte	tv	

Documenting your data set

Creating documentation

- Creating a record of what you found as you explored your data during cleaning will help you make data construction and analysis decisions
- As a Research Assistant, you should be more concerned with *finding* strange patterns in the data than with *fixing* them
- Having these issues listed in a document will make it easier to discuss with the rest of your team how to address them
- R and Stata markdown packages are particularly useful to create this documentation

Check variables consistency

- Check that values are consistent across variables
- For example, if an individual is male, then he cannot be pregnant
- This kind of inconsistency should usually be corrected during the high-frequency checks, but often times there's no time when the enumerators are in the field to identify and correct all of them
- So if you find any issues, create flag variables that identify observations with inconsistent values

Identify and document outliers

- We do not want our results to be driven by a few individuals. For example, if the village leaders get all benefits
- There is no exact rule for what is an outlier. Ask if your PI for preference of specific rule
- Identifying outliers often comes down to common sense: can the outlier be explained by typos?This is especially common when selecting units from multiple choice lists
- RAs should try to identify as many discrepant values as possible, even at the cost of not correcting them

Useful Stata commands to identify outliers

- `sum detail`
- `tabulate`
- `inspect`
- `assert`
- `histogram`

Add metadata as notes

- Variable labels must be short and self-explanatory
- That means they must often be different from the survey question itself
- However, you can add any other information to the variable as a note
- Examples of relevant information are question wording, value constraints and relevance conditions
- In Stata, you can use the `notes` command to add this information to your variable
- In R, you can use `attributes` to do the same

Other data cleaning tasks

Recategorize values listed as “others”

- Categorical variables usually have an open-ended “other, specify” option that is saved as a string
- Answers that appear frequently in the open-ended question can be included as a new category in the categorical variable
- That is usually done during the pilot or the high-frequency checks, but it is possible that there are still relevant categories left out

Drop variables from survey

- Some variables are created to be used within the survey and for survey checks
- That is the case of most calculate fields, as well as notes and duration variables
- Variables that are not part of the questionnaire itself may be dropped from the clean data set

Ordering variables

- It is recommended the variables in the final data set follow the some order as in the questionnaire
- If you created new variables during the data cleaning, for example to change roster codes, they will probably be out of order
- You may want to reorder those variables so the data set is easier to read and to compare to the questionnaire

File saving conventions

Saving files

- During the data cleaning process, you might have saved multiple intermediate files, for example if you cleaned long modules separately to make your code more readable
- After cleaning your data and merging it back together, you'll want to save a final cleaned data set, containing all variables from your survey
- This new data set will probably be quite heavy. Use `compress` to save your variables in the most economic format
- It's often desirable to save your data set in a previous Stata version, so other members of your team will not have version conflicts. To do this, use `saveold`

Naming files

- Make sure all output files, datasets and others are clearly and uniquely labeled, i.e.: “desc_stats_tmt_only.xls” “input_plan_adm_data.dta”
- It’s often desirable to have the names of your data sets and do-files linked, so it is easy to understand which do-files is creating which data set, such as “merge.do” and “merged.dta” or “cleaning.do” and “clean.dta”
- Do not use _v1, _v2 etc. for any final files. This leads to bugs in do-files that depend on these files when a new versions is added.
- It’s ok to use _v1, _v2 etc. for old versions of files if you **really** need to keep an archive

Appendix

Version control

- Track changes to the data and the code at the same time so you notice when significant changes are made

Publising data

- DIME *strongly recommends* that teams submit their cleaned data sets to the Microdata Catalog
- DIME Analytics can perform a review of the data cleaning code at this stage