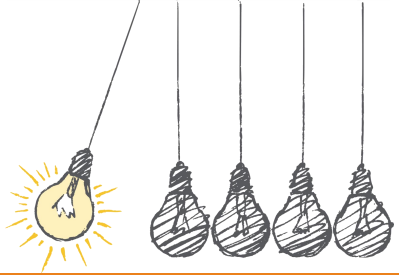
Data Construction



DIME RA Onbarding Course

February 13, 2020

Development Impact Evaluation (DIME)
The World Bank

WORLD BANK GROUP

i2i DIME
TRANSFORM DEVELOPMENT

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

Constructing analytical variables

Preventing mistakes

Panel data sets

Construction outputs

Exercises

What is data construction

- Constructing variables means processing the data points as provided in the raw data to make them suitable for analysis
- It is at this stage that the raw data is transformed into analysis data
- This is done by creating derived variables (e.g. dummies, indices, interactions)
- It is the only stage when changes will be made to data points
- Construction is closely linked to research design and questionnaire design
- Ideally, indicator construction should be done right after data cleaning, according to the pre-analysis plan

What is data construction

- Construction is closely linked to research design and questionnaire design
- Ideally, indicator construction should be done right after data cleaning, according to the pre-analysis plan
- This is when you will use the knowledge of the data you acquired and the documentation you created during the cleaning step the most
- It is often useful to start looking at comparisons and other documentation outside the code editor

What is data construction

Inputs

- One or more cleaned data sets
- Master data sets

Outputs

- One or more analysis (constructed) data sets
- One codebook for each analysis data set
- Construction documentation

Tasks

- Unit of observation (in the survey) $\rightarrow$ Unit of analysis
- Survey question $\rightarrow$ Analysis indicator

Why is construction a separate task from data cleaning?

1. To clearly differentiate the data originally collected from the result of data processing decisions
2. To ensure that variable definition is consistent across data sources
    - During data cleaning, we create one output for each input
    - During data construction, we may have multiple inputs and outputs
    - For example, we may have multiple rounds of data collection that will be cleaned separately, but we want all of them to be constructed in the same manner

What to plan ahead

- What are the final indicators needed to answer a research question
- How they are defined and calculated
- What are the steps to get there
- How to deal with different rounds of data collections

Why is construction a separate task from data analysis?

- In practice, data construction often times happens at the same time as data analysis
- As you analyze the data, different constructed variables will become necessary, as well as subsets and other alterations to the data
- However, even if construction ends up coming before analysis only in the order the code is run, it's important to think of them as different steps

Why is construction a separate task from data analysis?

Constructing analytical variables

Create new variables

- Create new variables instead of overwriting the original information
- Constructed variables should have intuitive and functional names
- Order the data set so that related variables are close to each other

Addressing outliers

During construction, you will address the issues you observed in the data during cleaning, including outliers and missing values

Addressing outliers

During construction, you will address the issues you observed in the data during cleaning, including outliers and missing values

- The one thing you don't want to do is to drop a whole observations because of outliers
- Common ways to address outliers are trimmings and winsorizing

Addressing outliers

How to treat outliers and impute missing values are research questions, but there are a few things to keep in mind

- Make sure to document what was the approach chosen by the team, and why you decided to use it in any particular case
- These decisions may affect variable distribution and observed results, so keep the original variable in the data set instead of replacing it

Standardizing units

Make sure there is consistency across constructed variables:

- We recommend coding yes/no questions as either 1 and 0 or TRUE and FALSE, so they can be used numerically as frequencies in means and as dummies in regressions
- For non-binary categorical variables, check that labels and levels have the same correspondence across variables that use the same options
- Numeric variables that are compared or aggregated need to be converted to the same scale or unit of measure
    - When converting units, set conversion rates in the master do-file using globals

Creating aggregate measures

- The most simple case of new variables to be created are aggregate indicators
- Jumping to the step where you actually create this variables seems intuitive, but it can also cause you a lot of problems, as overlooking details may affect your results
- It is important to check and double-check the value assignments of questions, as well as their scales, before constructing new variables based on them
- Look at the distributions of both the original and the constructed variables

Creating aggregate measures

- The most simple case of new variables to be created are aggregate indicators
- Jumping to the step where you actually create this variables seems intuitive, but it can also cause you a lot of problems, as overlooking details may affect your results
- It is important to check and double-check the value assignments of questions, as well as their scales, before constructing new variables based on them
- Look at the distributions of both the original and the constructed variables

Creating aggregate measures

- Often times it's easier to deal with long data sets when aggregating
- Be mindful of how missing values are treated
- In R, missing values are "contagious" $\rightarrow$ use `na.rm` to explicitly decide how to deal with missings

Creating aggregate measures

In Stata, different commands treat missings differently:

- `gen income_total = income_wage + income_rent + income_sales`
- `egen income_total = rowtotal(income_wage income_rent income_sales)`
- `egen income_total = rowtotal(income_wage income_rent income_sales), m`
- `collapse (sum) income_wage_hh = income_wage`
- `collapse (mean) income_wage_hh_mean = income_wage`
- `collapse (median) income_wage_hh_median = income_wage`

Merging data sets

- Merging may change both the number of observations and the value of variables
- Be careful when merging data sets that don't have the same identifiers
- Stata and R treat conflicting values differently: R creates two variables, and Stata keeps the master data set entries

Merging data sets

options	Description
Options	
keepusing(*varlist*)	variables to keep from using data; default is all
generate(*newvar*)	name of new variable to mark merge results; default is **_merge**
nogenerate	do not create **_merge** variable
nolabel	do not copy value-label definitions from using
nonotes	do not copy notes from using
update	update missing values of same-named variables in master with values from using
replace	replace all values of same-named variables in master with nonmissing values from using (requires **update**)
noreport	do not display match result summary table
force	allow string/numeric variable type mismatch without error
Results	
assert(*results*)	specify required match results
keep(*results*)	specify which match results to keep
sorted	do not sort; datasets already sorted

Merging data sets

join {dplyr} R Documentation

Join two tbls together

Description

These are generic functions that dispatch to individual tbl methods - see the method documentation for details of individual data sources. x and y should usually be from the same data source, but if copy is TRUE, y will automatically be copied to the same source as x.

Usage

```
inner_join(x, y, by = NULL, copy = FALSE, suffix = c(".x", ".y"),
  ...)

left_join(x, y, by = NULL, copy = FALSE, suffix = c(".x", ".y"), ...)

right_join(x, y, by = NULL, copy = FALSE, suffix = c(".x", ".y"),
  ...)

full_join(x, y, by = NULL, copy = FALSE, suffix = c(".x", ".y"), ...)

semi_join(x, y, by = NULL, copy = FALSE, ...)

nest_join(x, y, by = NULL, copy = FALSE, keep = FALSE, name = NULL,
  ...)

anti_join(x, y, by = NULL, copy = FALSE, ...)
```

Reshaping observations

- Reshape changes the unit of observation
- In Stata, reshape has a very unique syntax
- You need to have identifying variables in the data set to be able to reshape it
- Reshape will create observations in the long data set even when variables are missing

Reshaping observations

```
30
31      use "${hh_ml_encrypt}/Raw Identified Data/SLWRMP - Household Midline - Raw Identified.dta", clear
32
33      * List all plot level variables
34      local plotVars  pm_match plotsize plottravel plottime plotresp ///
35                      plotkityn plotowns plotused plotreallocation ///
36                      obtainmth obtainyear kitplotpaid kitplotvalue ///
37                      kitplotrent kitother kitotherper kitothermth ///
38                      kitotheryear kitotherpaid kitotherrent kitexchange
39
40      * Prepare locals with list of variables to keep and to reshape
41      local keepVars  hhid comid comname dist anycrop_*
42
43      foreach var of local plotVars {
44          local keepVars      `" `keepVars' `var'* "'
45          local reshapeVars   `" `reshapeVars' `var' "'
46      }
47
48      * Keep only plot level variables
49      keep `keepVars'
50
51      * Make long per plot
52      reshape long `reshapeVars' anycrop_1 anycrop_2, i(hhid) j(plot) string
53
54      * Clean variable that identifies the plots
55      replace  plot = subinstr(plot, "_", "", .)
56      destring plot, replace
57
58      * Drop blank observations created by the reshape
59      drop if missing(plotdesc)    // blank obs
```

Preventing mistakes

Where things go wrong

- The more complex construction tasks involve changing the structure of the data, such as sample and unit of observation
- Merges, reshapes and collapses may change the number of observation and create missing entries
- Make sure to read about how each command treats missing observations
- If you are subsetting your data, drop observations explicitly, indicating why you are doing that and how the data set changed

Write pseudocode

- Describe the steps to create your indicator in plain English
- Refine the sub-steps involved
- When you are getting into too much detail, write code
- Think about possible errors that may come up at each sub-step

Think about expected results

- Think about how the command you are using treats missing values
- Try to predict the result you will get
  - Will all observations merge?
  - Will the number of observations change?
  - Will missing values be created?

Document the observed results

- Explore the actual results from the operation
- Write down in comments what happened
- Add comments to the code explaining unexpected consequences

Document the observed results

Here's an example from Targetting the Ultra-Poor in Afghanistan by Camila Ayala and Thomas Escande

```
1
2       * Merge
3       use `time_pf', clear
4       merge 1:1 hhid using `time_pm'
5       /*
6       Result                           # of obs.
7       -----------------------------------------
8       not matched                             511
9           from master                         503   (_merge==1)
10          from using                            8   (_merge==2)
11      matched                               1,980   (_merge==3)
12      -----------------------------------------
13          Household not matching from using are either because they have no lady
14      of the HH (1493, 1558, 1720, 5437, 5976) or the Lady of the HH reported
15      "don't answer" in age (1151, 1286, 6561)
16          Household not matching from master because households have no primary
17      male.
18      */
19
20      drop _merge
21
```

Build checks into your code

- Test the unit of observation and ID variable
- Throw error messages or break the code if
  - Confirm your expected results
  - Check if the outputs are changing when you run the code again
- Use `assert` in Stata and `stopifnot()` in R

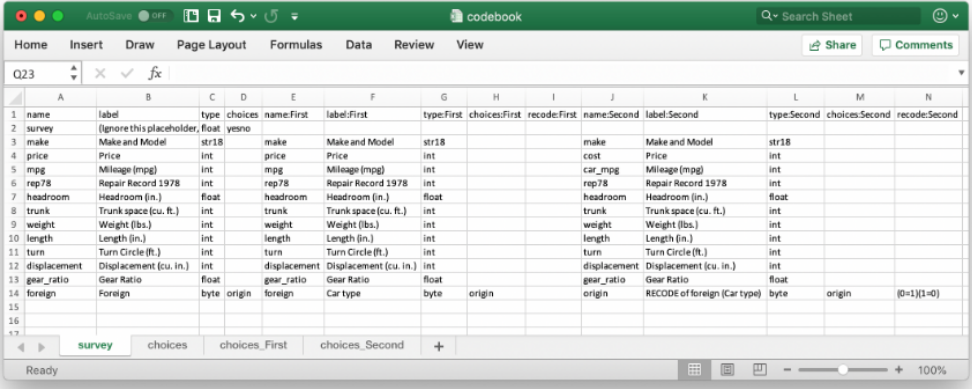
Build checks into your code

```
/*******************************************************************************
    PART 3: Append rounds
*******************************************************************************/

    cap erase "${panel_doc}/Codebooks/Intermediate/Household/SLWRMP - Household surveys - Household level - Reconcile_appended.xlsx"
    iecodebook append    `baseline' `midline' `endline' ///
                         using "${panel_doc}/Codebooks/Intermediate/Household/SLWRMP - Household surveys - Household level - Reconcile.xlsx", ///
                surveys(Baseline Midline Endline) ///
                clear

    merge m:1   hhid using "${hh_dt}/hh_master_reconciled"

    * Check merge: 6446 creating problems here
    qui count if _merge == 1
    assert r(N) == 1

    * Check that HHs that are in master but not in panel:
    * Either they were never surveyed or they were surveyed in midline correction
    * surveys and the data needs to be collapsed to hh level in midline
    assert  !(_merge == 2 & (d_surveyed_bl == 1 | d_surveyed_ml == 1 | d_surveyed_el == 1) & ///
            !inlist(hhid, 3248, 3249, 4739, 6250, 6463))
```

Panel data sets

Creating panel data sets

The `iecodebook append` subcommand was created to help reconcile and append data sets



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	name	label	type	choices	name:First	label:First	type:First	choices:First	recode:First	name:Second	label:Second	type:Second	choices:Second	recode:Second
2	survey	(Ignore this placeholder,	float	yesno										
3	make	Make and Model	str18		make	Make and Model	str18			make	Make and Model	str18		
4	price	Price	int		price	Price	int			cost	Price	int		
5	mpg	Mileage (mpg)	int		mpg	Mileage (mpg)	int			car_mpg	Mileage (mpg)	int		
6	rep78	Repair Record 1978	int		rep78	Repair Record 1978	int			rep78	Repair Record 1978	int		
7	headroom	Headroom (in.)	float		headroom	Headroom (in.)	float			headroom	Headroom (in.)	float		
8	trunk	Trunk space (cu. ft.)	int		trunk	Trunk space (cu. ft.)	int			trunk	Trunk space (cu. ft.)	int		
9	weight	Weight (lbs.)	int		weight	Weight (lbs.)	int			weight	Weight (lbs.)	int		
10	length	Length (in.)	int		length	Length (in.)	int			length	Length (in.)	int		
11	turn	Turn Circle (ft.)	int		turn	Turn Circle (ft.)	int			turn	Turn Circle (ft.)	int		
12	displacement	Displacement (cu. in.)	int		displacement	Displacement (cu. in.)	int			displacement	Displacement (cu. in.)	int		
13	gear_ratio	Gear Ratio	float		gear_ratio	Gear Ratio	float			gear_ratio	Gear Ratio	float		
14	foreign	Foreign	byte	origin	foreign	Car type	byte	origin		origin	RECODE of foreign (Car type)	byte	origin	(0=1)(1=0)
15														
16														
17														

Construction indicators in panel data sets

- It is common to construct indicators soon after receiving data from a new survey round
- However, creating indicators for each round separately increases the risk of using different definitions every time
- Having a well-established definition for each constructed variable helps prevent that mistake
- But the best way to guarantee it won't happen is to merge all rounds of data collection first and create the indicators for all rounds in the same script
- In addition to preventing inconsistencies, this process will also save you time and give you an opportunity to review your original construction code

Construction outputs

Constructed data sets

- A constructed data set is built to answer an analysis question
- Different pieces of analysis may require different samples or different units of observation
- You may have as many constructed data sets as required for analysis
- Don’t worry if you cannot create a single, “canonical” analysis data set
- If you have multiple constructed data sets, name them carefully so you know when to use each of them

Documenting construction

- Documentation is an output of construction as relevant as the code and the data
- Someone unfamiliar with the project should be able to understand the contents of the analysis data sets, and the steps taken to create them
- Data construction involves translating concrete data points to more abstract measurements
- Document exactly how each variable is derived or calculated
- Carefully record how specific variables have been combined, recoded, and scaled, and refer to those records in the code
- This can be part of a wider discussion with your team about creating protocols for variable definition, which will guarantee that indicators are defined consistently across projects

Exercises

Exercise

Write template code for reshape, merge and aggregation (in long and wide data sets)

- Open help file
- Think of treatment of missings
- Think how the operation affects the number of observation
- Write pseudocode
- Predict results
- Document the output
- Add checks into the code