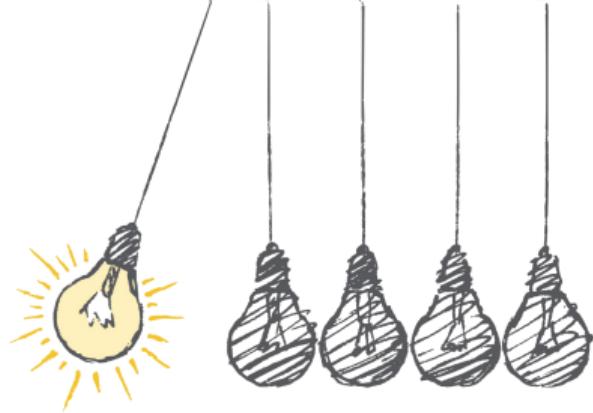




iefieldkit: Stata commands for primary data collection and data cleaning



Kristoffer Bjärkefur, Luíza Cardoso de Andrade, and Benjamin Daniels

July 11, 2019

Development Impact Evaluation (DIME)
The World Bank



A brief introduction to iefieldkit

iefieldkit is designed to apply many of the lessons from the last presentation to other common tasks in the DIME data collection workflow.

- Working with primary data from developing contexts.
- Large teams with many projects and diverse skillsets.
- Standardization of *easy* tasks adds value and avoids error.



A brief introduction to iefieldkit

Data collection is a process that has traditionally suffered from low levels of documentation, standardization, and replicability.

iefieldkit is meant to bring that mindset to these tasks, which are often partially conducted by staff with little or no Stata skills.

We think it is a great example of using Stata to bring ideals of standardization and replicability to one of our core tasks that is usually considered less technical.



A brief introduction to iefieldkit

Data collection requires lots of analytical work that we don't necessarily want to keep in Stata dofiles. iefieldkit provides a start-to-finish data collection and cleaning workflow that is self-documenting.

Core commands:

- `ietestform` ensures that ODK surveys are Stata-optimized
- `ieduplicates` & `iecompdup` identify and resolve duplicates in data
- `iecodebook` uses spreadsheet codebooks to clean or append data

All `iefieldkit` commands automatically output human- and machine-readable spreadsheet documentation as a functional part of the intended workflow.

<https://dimewiki.worldbank.org/iefieldkit>



ietestform

`ietestform`: Collecting Stata-optimized data in ODK

We believe that it is best to have automated quality control in place, even before data is ready for Stata. Stata is a very convenient tool for this purpose because our teams are already familiar with it. This idea can extend to any workflow involving structured non-Stata components.

- Open Data Kit (ODK) is a common data collection software in the field
- Many of our teams use SurveyCTO, a proprietary variant of ODK, and almost all our teams use Stata for data analysis
- But ODK data isn't naturally prepared for Stata, and Stata doesn't know what ODK data can look like
- Therefore it is very easy to make “non-errors” in ODK programming that are time-consuming and challenging to fix for Stata after the data is already collected

ietestform: Collecting Stata-optimized data in ODK

ODK data collection (and proprietary implementations like SurveyCTO) are common in primary data collection.

- Structured “pseudo-code” in spreadsheet format is built into survey
- Material is both human and machine-readable
- Lots of options for controlling data format

A	B	C	labelAmharic
1	type	name	
2	start	start	
3	note	Ask_consent	Please ask respondent's consent to being interviewed and being part of the network mapping section of the survey
4	select_one_yesno	consent_survey	Did the job seeker consent to being interviewed and being part of the network mapping section of the survey
5	note	note_1	Welcome to the Hawassa Industrial Park Pre Recruitment Application Form
6	text	screening_id	ID
7	date	date	Date
8	begin_group	#2	Applicant personal information
9	text	ji_name	Applicant Full Name
10	integer	ji_idno	Applicant Identification Card Number
11	integer	ji_age	Age
12	select_one_gender	ji_gender	Gender
13	select_one_marital_st	ji_marital	Marital status
14	select_one_region	ji_region	Region
15	select_one_zone	ji_zone	Zone
16	text	ji_city	City
17	select_one_woreda	ji_woreda	Woreda
18	text	ji_kebele	Kebele
19	integer	ji_houseno	House number (Enter zero if the respondent doesn't have a house number)
20	select_one_ruralurban	ji_rururb	Do you live in a rural or urban area?
21	integer	ji_phone1	Phone number 1
22	integer	ji_phone2	Phone number 2
23	select_one_education	ji_educ	Education:
24	select_multiple_language	ji_language	What language(s) are you comfortable speaking?
25	text	ji_language_oth	Please specify other
26	select_one_employment	ji_empl	Current employment status (job seeker / employed)
27	integer	ji_hhsize	How many household members does your household have?
28	begin_repeat	r1	Name household members
29	calculator	ji_pos	
30	text	ji_name_hhm	Full name household member #\\$[i_pos]
31	end_repeat	r1	
32	select_one_yesno	ji_fem_hhm	Do you have a female household member between 14-35 years old (e.g. sister)
33	integer	ji_fem_hhno	How many female household members between 14-35 years old do you have?
34	end_group	#2	Applicant personal information
35			

`ietestform`: Collecting Stata-optimized data in ODK

BUT... the survey forms are primarily built and operated by field staff or survey firms, not by Stata coders!

So we designed `ietestform` to read the survey definition file and give instructions on best practices and likely errors that are easier to fix during survey design than after data collection.

Major tests for Stata optimization:

- All variable names are Stata-compliant, including auto-generated ones in rosters and other dynamic fields
- All variables use multi-language support to create a “Stata” variable label that is *not* the full text of the question
- All value labels are Stata-compliant

ietestform: Generating a flags report

Simple syntax:

```
ietestform  
, surveyform("/path/to/survey.xlsx")  
report("/path/to/report.csv")
```

- CSV format supports version control in Git
- Flags report likely errors
- Sometimes functionality may be desired, so you do not necessarily want an “empty” report

The screenshot shows a Microsoft Excel spreadsheet titled "report". The first few rows contain metadata about the report's creation:

1	This report was created by user bbdaniels on 28Jun2019 using the Stata command ietestform							
2								
3	Use either of these links to read more about this command:							
4	https://github.com/worldbank/iefieldkit							
5	https://dimewiki.worldbank.org/wiki/ietestform							
6								
7								
8	Form ID	decie_eth_hawassainustrialparkie_screening_form_June11						
9	Form Title	DECIETH_HawassaIndustrialParkIE_Screening_Form_June11						
10	Form Version	1906111933						
11	Form File	/Users/bbdaniels/Downloads/Data entry_screening form_translated (1).xlsx						
12								
13	-----							
14	-----							
15	TEST: SPACES BEFORE OR AFTER STRING (choice sheet)							
16	-----							
17								
18	The string values in [label] column in the choice sheet are imported as strings and has leading or trailing spaces in the Excel file in the following cases:							
19								
20	row	label						
21	10	Gamo						
22	78	Amharic						
23	79	Oromifa						
24	80	Tigrigna						
25	81	Harari/Aderi						
26								

ietestform: Generating a flags report

Additional syntax checks ensure machine-compatibility after import. All flags are linked with a complete explanation for the practice on <https://dimewiki.worldbank.org/ietestform>.

- All groups and loops open and close correctly
- No leading or trailing spaces in fields
- No repeated values or value labels, and no unused values in value labelling

The screenshot shows an Excel spreadsheet with the following content:

row	label
22	10 Gamo
23	78 Amharic
24	79 Oromifa
25	80 Tigrigna
26	81 Harari/Aderi

Cell A1 contains "This report was created by user bbdaniels on 28Jun2019 using the Stata command ietestform". Cells A2-A3 contain links to GitHub and the dimewiki page. Cells A4-A5 list form details: ID, Title, Version, and File path. Cell A15 contains "TEST: SPACES BEFORE OR AFTER STRING (choice sheet)". Cells A18-A19 explain a bug where string values in the choice sheet have leading/trailing spaces. The bottom status bar shows "Ready" and "100%".



ieduplicates & iecompdup

ieduplicates: Real-time data quality assurance

- Primary data coming in from the field can be very messy!
- Cleaning raw data and doing quality assurance is time-sensitive: it has to be done while the survey team is still on site
- Entries with duplicated identifier variables are particularly bad: they prevent the team from knowing the results of other quality checks, and therefore from efficiently implementing things like followup surveys
- Therefore there is a huge value to our team for having a standardized and pre-programmed process for handling duplicates coming in from the field

Additional challenges in this phase include interfacing with non-technical staff in the field; and creating documentation of the resolution of issues.

ieduplicates: Real-time data quality assurance

ieduplicates implements a standard self-documenting workflow using Excel data output and input. The command outputs a report of duplicates into Excel, and the user responds in pre-defined ways to each flagged observation.

- Run ieduplicates on the raw data. If there are no duplicates, you are done!
- If there are duplicates in the Excel report, analyze them using Stata and/or field records to find out the correct resolution.
- Enter the resolutions on the corresponding observations in the report outputted by ieduplicates.
- After entering the corrections, save the report in the same location with the same name.

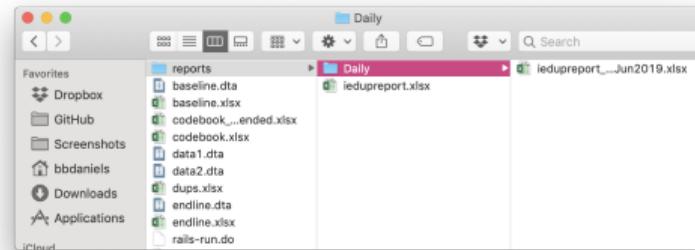
Why Excel? Because it is easier for everyone to read and understand when there are large numbers of information to process, rather than de-coding Stata code.

ieduplicates: Listing duplicates in data

On the first run, `ieduplicates` does two main tasks:

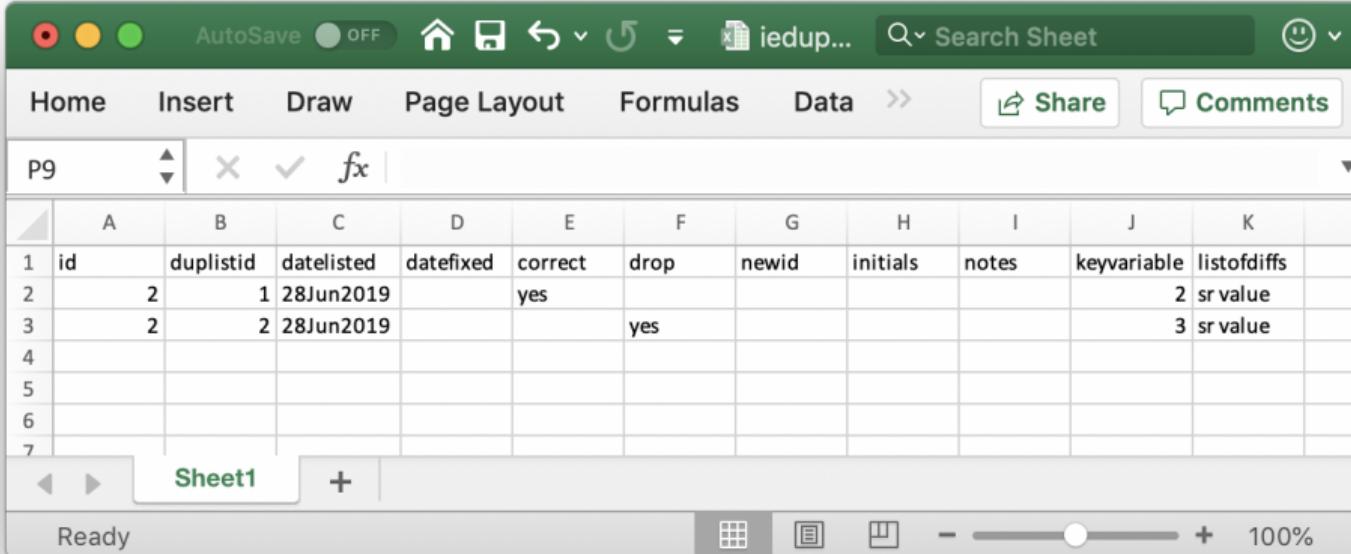
- Lists all duplicates in data into a file called `iedupreport.xlsx` and backs up a dated copy
- Removes all copies of duplicates from the data so other quality-assurance tasks like back-checks can be performed on unambiguous portion of data

```
ieduplicates idvariable  
, folder("/path/to/folder/")  
uniquevars(keyvariable)
```



ieduplicates: Correcting duplicates in data

ieduplicates expects standardized, structured responses to the observations flagged, so that they can be written and read quickly by any staff.



The screenshot shows a spreadsheet application interface with a green header bar. The menu bar includes AutoSave (OFF), Home, Insert, Draw, Page Layout, Formulas, Data, Share, and Comments. The active sheet is Sheet1. The table has columns labeled A through L. Row 1 contains the headers: id, duplistid, datelisted, datefixed, correct, drop, newid, initials, notes, keyvariable, and listofdiffs. Rows 2 and 3 show data entries. Row 2 has values: 2, 1, 28Jun2019, (empty), yes, (empty), (empty), (empty), (empty), 2, sr value. Row 3 has values: 2, 2, 28Jun2019, (empty), (empty), yes, (empty), (empty), (empty), 3, sr value. Rows 4 through 7 are empty. The bottom navigation bar includes back, forward, and new sheet buttons, along with a zoom slider set at 100%.

	A	B	C	D	E	F	G	H	I	J	K	L
1	id	duplistid	datelisted	datefixed	correct	drop	newid	initials	notes	keyvariable	listofdiffs	
2	2	1	28Jun2019		yes					2	sr value	
3	2	2	28Jun2019			yes				3	sr value	
4												
5												
6												
7												

`ieduplicates`: Real-time data quality assurance

On *future* runs, `ieduplicates` will first apply all corrections in the current version of the duplicates report to the raw data – accept as correct, drop, or change ID.

- Run `ieduplicates` on the raw data again. The corrections you have entered will be applied, and only duplicates that are still not resolved are removed this time. Note that the raw data is unchanged, and therefore the report leaves a record of how all duplicates were resolved in the creation of the final dataset.
- Repeat these steps every time you get new data. Our recommendation is that this is done every day that you have new data.

iecompdup: Analyzing duplicates in data

Used on the raw data, `iecompdup` will return basic information about how duplicate observations are the same or different (with the relevant information stored in `return` for programming of reports). Naturally there is no way to fully automate the resolution process, but we look for three main groups:

Case 1. Double submission of the same observation, with the same data.

Resolution: Keep only one of the entries.

Case 2. Double submission of the same observation, with different data.

Resolution: Return to field team for audit.

Case 3. Incorrect ID variable.

Resolution: Return to field team to obtain correct ID.

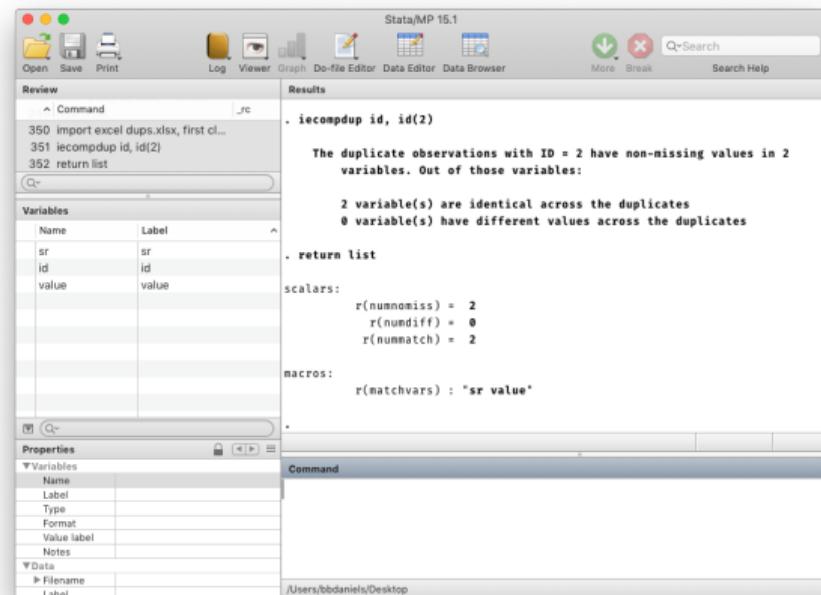
iecompdup: Analyzing duplicates in data

Syntax:

```
iecompdup idvariable, id(idvalue)
```

Notes:

- Only accepts pairwise comparisons; any help on reporting about larger groups would be appreciated!
- No other output or documentation; intended to encourage careful review and documentation in the main ieduplicates report



The screenshot shows the Stata/MP 15.1 interface with the following details:

- Review:** Shows the command history:
 - 350 Import excel dups.xlsx, first cl...
 - 351 iecompdup id, id(2)
 - 352 return list
- Variables:** A table showing variables and their labels:

Name	Label
sr	sr
id	id
value	value
- Properties:** A panel showing variable properties for 'sr':
 - Name: sr
 - Label: sr
 - Type:
 - Format:
 - Value label:
 - Notes:
- Results:** The output of the iecompdup command:

```
. iecompdup id, id(2)

The duplicate observations with ID = 2 have non-missing values in 2
variables. Out of those variables:

    2 variable(s) are identical across the duplicates
    0 variable(s) have different values across the duplicates

. return list

scalars:
    r(numnomin) =  2
    r(numndiff) =  0
    r(nummatch) =  2

macros:
    r(matchvars) : "sr value"
```
- Command:** Shows the full command entered: `/Users/bbdaniels/Desktop`



iecodebook

Three tasks for reproducible data construction

- **Data cleaning:** `iecodebook apply`
Reads an Excel codebook that specifies renames, recodes, variable labels, and value labels, and applies them to the current dataset.
- **Dataset combination:** `iecodebook append`
Reads an Excel codebook that specifies how variables should be harmonized across two or more datasets - rename, recode, variable labels, and value labels - applies the harmonization, and appends the datasets.
- **Data documentation:** `iecodebook export`
Creates an Excel codebook that describes the current dataset, and optionally produces an export version of the dataset with only variables used in specified dofiles.

<https://dimewiki.worldbank.org/iecodebook>

iecodebook apply: Data cleaning made easy

iecodebook apply runs an arbitrary number of rename, recode, and label commands in a single line of code.

- Operates on dataset in memory
- Commands in structured spreadsheet for future reference
- Eliminates repetitive coding

Syntax:

```
iecodebook apply  
using "/path/to/codebook.xlsx"
```



iecodebook apply: Setting up a template

```
// Load data  
sysuse auto.dta , clear  
  
// Create cleaning template  
iecodebook template  
using "/path/to/codebook.xlsx"
```

The template subcommand sets up the spreadsheet based on the data in memory.

iecodebook apply: A step back

The codebook is nothing more than a structured way to write common Stata commands outside of Stata. Why did we decide to spend time implementing this additional layer of abstraction?

- These are easy tasks: any Stata user can write `rename`, `recode`, and `label`
- But doing it over and over again is extremely boring, and demands attention to detail (such as the ordering of the commands) that can cause silly errors
- Development datasets are often really large and messy (such as a dataset received from a partner agency)
- The goal of the Excel codebook is therefore to allow users to input large amounts of information quickly;
- and to make sure that information is structured so that other users can review it efficiently

iecodebook apply: Filling out the template

The screenshot shows a Microsoft Excel spreadsheet titled "codebook — Saved to my Mac". The ribbon menu includes Home, Insert, Draw, Page Layout, Formulas, Data, Review, and View. The top-left cell contains "K15". The table has columns labeled A through J. Row 1 defines the structure: A (name), B (label), C (type), D (choices), E (name:current), F (label:current), G (type:current), H (choices:current), I (recode:current), and J. Row 2 contains a note: "survey (ignore this placeholder, but do not delete it. Thanks!)". Rows 3 through 13 list car attributes with their descriptions and data types. Row 14 defines "domestic" as "Domestic Make and Model" with type "yesno". Row 15 is empty. Row 16 is the footer row with cells for survey, choices, choices_current, and a plus sign (+). The bottom of the screen shows standard Excel navigation and zoom controls.

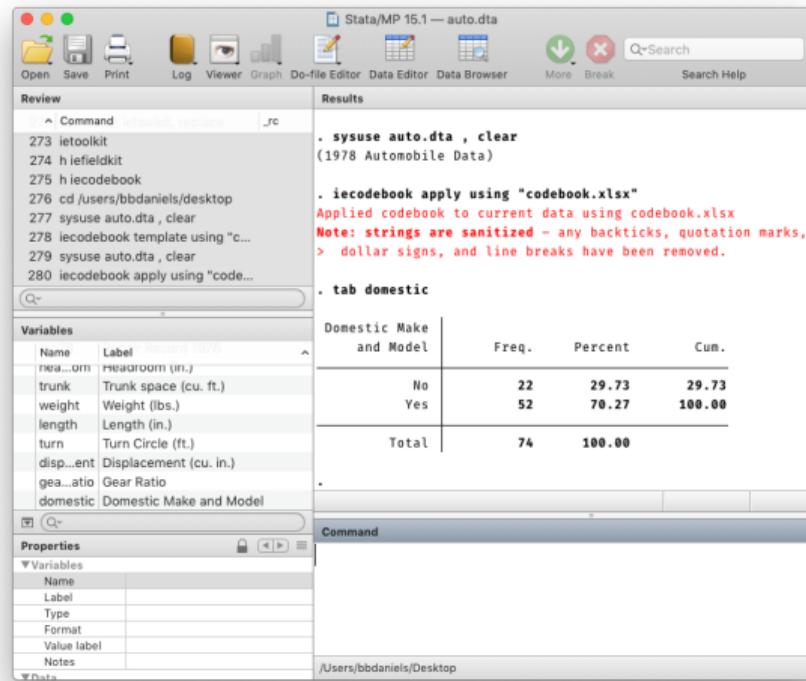
A	B	C	D	E	F	G	H	I	J
1	name	label							
2	survey	(ignore this placeholder, but do not delete it. Thanks!)	float	yesno					
3				make	Make and Model	str18			
4				price	Price	int			
5				mpg	Mileage (mpg)	int			
6				rep78	Repair Record 1978	int			
7				headroom	Headroom (in.)	float			
8				trunk	Trunk space (cu. ft.)	int			
9				weight	Weight (lbs.)	int			
10				length	Length (in.)	int			
11				turn	Turn Circle (ft.)	int			
12				displacement	Displacement (cu. in.)	int			
13				gear_ratio	Gear Ratio	float			
14	domestic	Domestic Make and Model	yesno	foreign	Car type	byte	origin	(0=1)(1=0)	
15									
16									

iecodebook apply: Applying the spreadsheet codebook to the data

```
// Load data  
sysuse auto.dta , clear  
  
// Apply cleaning template  
iecodebook apply  
    using "/path/to/codebook.xlsx"
```

Simply changing template to apply in the command gives the basic syntax.

The drop option removes all un-named variables. The missingvalues() option specifies extended missing value codes for the whole dataset.



iecodebook append: Data combination

DIME Training Baseline

Field	Question	Answer			
name <i>(required)</i>	What is your name?				
quest <i>(required)</i>	What is your quest?				
airspeed <i>(required)</i>	What is the average airspeed of an unladen swallow?				
color <i>(required)</i>	What is your favorite color?	<table border="1"><tr><td>1 Red</td></tr><tr><td>2 Blue</td></tr><tr><td>3 Green</td></tr></table>	1 Red	2 Blue	3 Green
1 Red					
2 Blue					
3 Green					

DIME Training Endline

Field	Question	Answer				
resp_name <i>(required)</i>	What is your name?					
resp_quest <i>(required)</i>	What quest are you on?					
resp_color <i>(required)</i>	Among the following, which is your favorite color:	<table border="1"><tr><td>1 White</td></tr><tr><td>2 Blue</td></tr><tr><td>3 Red</td></tr><tr><td>4 Green</td></tr></table>	1 White	2 Blue	3 Red	4 Green
1 White						
2 Blue						
3 Red						
4 Green						
swallow_speed <i>(required)</i>	What is the average airspeed of an unladen African swallow?	<table border="1"><tr><td>1 0-25 km/h</td></tr><tr><td>2 26-50 km/h</td></tr><tr><td>3 51-75 km/h</td></tr><tr><td>4 76-100 km/h</td></tr></table>	1 0-25 km/h	2 26-50 km/h	3 51-75 km/h	4 76-100 km/h
1 0-25 km/h						
2 26-50 km/h						
3 51-75 km/h						
4 76-100 km/h						



iecodebook append: Data combination

iecodebook append runs an arbitrary number of rename, recode, and label commands on two or more datasets with the intention of harmonizing them. It then runs an append command on the harmonized data.

- Operates on datasets on disk
- All data structures in one spreadsheet for future reference

```
// Create codebook template  
iecodebook template  
"baseline.dta" "endline.dta"  
using "/path/to/codebook.xlsx"  
, surveys(Baseline Endline)
```

The template subcommand sets up the spreadsheet based on multiple datasets on disk. The surveys() option names the datasets in the template and is required.

iecodebook append: Setting up a spreadsheet template

The screenshot shows a Microsoft Excel spreadsheet titled "codebook". The spreadsheet contains survey questions and their corresponding data types. The columns are labeled A through O, and the rows are numbered 1 through 15. The data is organized into several rows:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	name	label	type	choices	name:Baseline	label:Baseline	type:Baseline	choices:Baseline	recode:Baseline	name:Endline	label:Endline	type:Endline	choices:Endline	recode:Endline	
2	survey	{ignore}	float	yesno											
3					name	What is your name?	str7								
4					quest	What is your quest?	str16								
5					airspeed	What is the average airspeed of an unladen swallow?	byte								
6					color	What is your favorite color?	byte	color							
7										resp_name	What is your name?	str7			
8										resp_quest	What quest are you on?	str16			
9										resp_color	Among the following, which is your favorite color:	byte	resp_color		
10										swallow_speed	What is the average airspeed of an unladen African swallow?	byte	swallow_speed		
11															
12															
13															
14															
15															

At the bottom of the spreadsheet, there are tabs for "survey", "choices", "choices_Baseline", "choices_Endline", and a plus sign icon.

iecodebook append: Filling out the template

The screenshot shows a Microsoft Excel spreadsheet titled "codebook — Saved to my Mac". The spreadsheet has a green header bar with standard Office icons and a title bar. Below the header is a menu bar with Home, Insert, Draw, Page Layout, Formulas, Data, Review, and View. To the right of the menu are Share and Comments buttons. The main area is a grid of cells labeled from A1 to O10. Row 1 contains column headers: name, label, type, choices, name:Baseline, label:Baseline, type:Baseline, choices:Baseline, recode:Baseline, name:Endline, label:Endline, type:Endline, choices:Endline, recode:Endline. Rows 2 through 6 contain survey items with their labels, types, and descriptions. Row 7 is empty. Row 8 contains a formula: (1/25=1)(25/50=2). Row 9 contains a formula: (2=2)(1=3)(3=4). Row 10 contains a formula: (1=1)(2=2)(3=3)(4=4). The bottom navigation bar includes tabs for survey, choices, choices_Baseline, choices_Endline, and a plus sign, along with zoom controls.

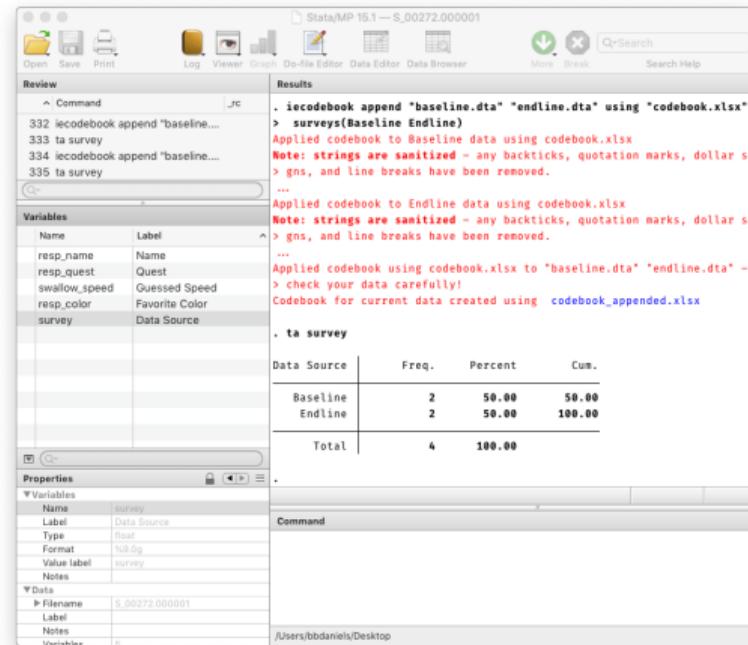
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	name	label		type	choices	name:Baseline	label:Baseline	type:Baseline	choices:Baseline	recode:Baseline	name:Endline	label:Endline	type:Endline	choices:Endline	recode:Endline
2	survey	(ignore this placeholder, but float)		yesno											
3	resp_name	Name				name	What is your na	str7			resp_name	What is your name?	str7		
4	resp_quest	Quest				quest	What is your qu	str16			resp_quest	What quest are you	str16		
5	swallow_speed	Guessed Speed			swallow_speed	airspeed	What is the avei	byte		(1/25=1)(25/50=2)	swallow_speed	What is the average	byte		swallow_speed
6	resp_color	Favorite Color			resp_color	color	What is your fav	byte	color	(2=2)(1=3)(3=4)	resp_color	Among the following	byte		resp_color
7															
8															
9															
10															

iecodebook append: Apply codebook to data

```
// Append the datasets  
iecodebook template  
    "baseline.dta" "endline.dta"  
        using "/path/to/codebook.xlsx"  
, surveys(Baseline Endline)
```

Simply changing template to append in the command gives the basic syntax.

All un-named variables are removed by default; nodrop cancels this. The missingvalues() option specifies extended missing value codes for the whole dataset.



iecodebook export: Document data for release

Function 1: Just create the codebook
for documentation

Function 2: Trim dataset based on
variables used in dofiles:

- Reads your dofiles
- Keeps only the variables that are used in analysis
- Creates a minimal codebook
- Rewards good syntax – you must:
 Spell variable names completely
 Avoid wildcards or lists: * ? -

Syntax:

```
iecodebook export [if] [in]  
    using "/path/to/codebook.xlsx"
```

and optionally:

```
, trim(  
    "/path/to/dofile1.do"  
    ["/path/to/dofile2.do"]  
    [...]  
)
```

Thank you!

**For more information about
iefieldkit, contact us or visit our
online resources:**

dimeanalytics@worldbank.org

dimewiki.worldbank.org/iefieldkit

github.com/worldbank/iefieldkit



DIME
analytics

