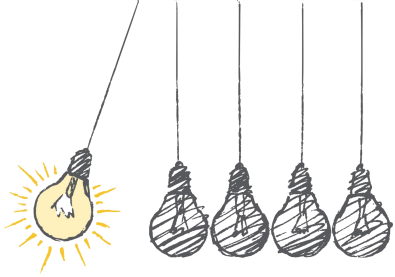
`iefieldkit`: Stata commands for primary data collection and data cleaning



Kristoffer Bjärkefur, Luíza Cardoso de Andrade, and Benjamin Daniels

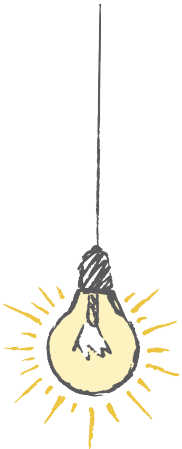
July 11, 2019

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.

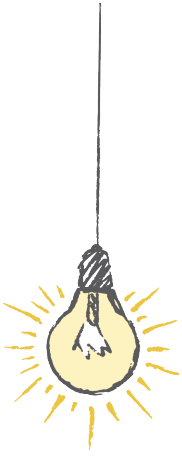


A brief introduction to `iefieldkit`

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

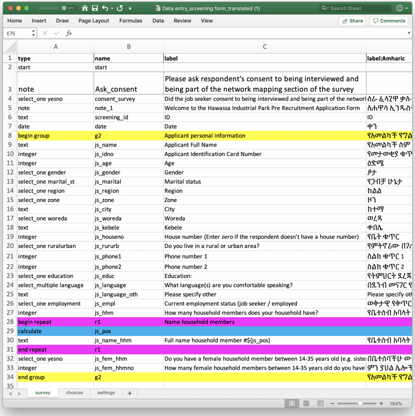
We believe that it is best to have automated quality control in place, even before data is ready for Stata. Stata is a very convenient tool for this purpose because our teams are already familiar with it. This idea can extend to any workflow involving structured non-Stata components.

- 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 preared 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	
1	type	name	label	label:Amharic
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 networl	ስራ ፈላጊዋ ቃለ-
5	note	note_1	Welcome to the Hawassa Industrial Park Pre Recruitment Application Form	ለሐዋሳ ኢንዱስ
6	text	screening_id	ID	ID
7	date	date	Date	ቀን
8	begin group	g2	Applicant personal information	የአመልካች የግል
9	text	js_name	Applicant Full Name	የአመልካች ስም
10	integer	js_idno	Applicant Identification Card Number	የመታወቂያ ቁጥ
11	integer	js_age	Age	ዕድሜ
12	select_one gender	js_gender	Gender	ፆታ
13	select_one marital_st	js_marital	Marital status	የጋብቻ ሁኔታ
14	select_one region	js_region	Region	ክልል
15	select_one zone	js_zone	Zone	ዞን
16	text	js_city	City	ከተማ
17	select_one woreda	js_woreda	Woreda	ወረዳ
18	text	js_kebele	Kebele	ቀበሌ
19	integer	js_houseno	House number (Enter zero if the respondent doesn't have a house number)	የቤት ቁጥር
20	select_one ruralurban	js_rururb	Do you live in a rural or urban area?	የምትኖሪው በገ
21	integer	js_phone1	Phone number 1	ስልክ ቁጥር 1
22	integer	js_phone2	Phone number 2	ስልክ ቁጥር 2
23	select_one education	js_educ	Education:	የትምህርት ደረጃ
24	select_multiple language	js_language	What language(s) are you comfortable speaking?	በደንብ መናገር የ
25	text	js_language_oth	Please specify other	Please specify oth
26	select_one employment	js_empl	Current employment status (job seeker / employed	ወቅታዊ የቅጥር
27	integer	js_hhm	How many household members does your household have?	የቤተሰብ አባላት
28	begin repeat	r1	Name household members	
29	calculate	js_pos		
30	text	js_name_hhm	Full name household member #${js_pos}	የቤተሰብ አባላት
31	end repeat	r1		
32	select_one yesno	js_fem_hhm	Do you have a female household member between 14-35 years old (e.g. siste	በቤተሰባችሁ ው
33	integer	js_fem_hhmno	How many female household members between 14-35 years old do you have	ምን ያህል ሴሎ
34	end group	g2		የአመልካች የግል
35				

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:

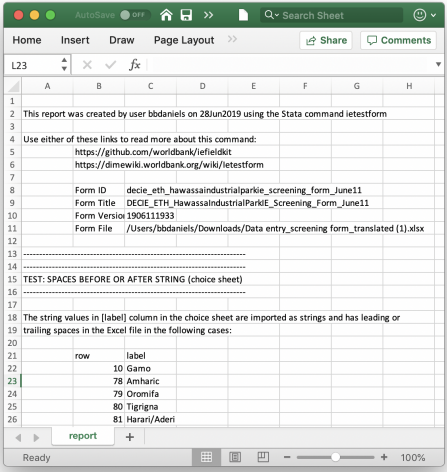
- All variable names are Stata-compliant, including auto-generated ones in rosters and other dynamic fields
- All variables use multi-language support to create a “Stata” variable label that is *not* the full text of the question
- All value labels are Stata-compliant

`ietestform`: Generating a flags report

Simple syntax:

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

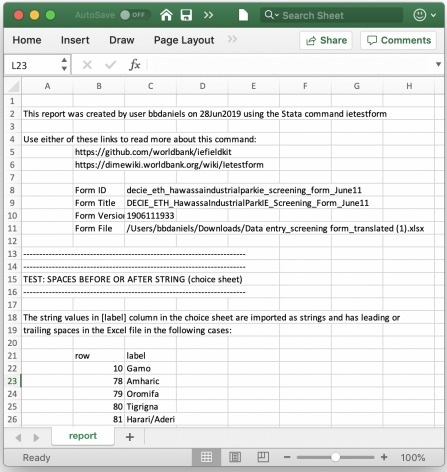
- CSV format supports version control in Git
- Flags report likely errors
- Sometimes functionality may be desired, so you do not necessarily want an “empty” report



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



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.

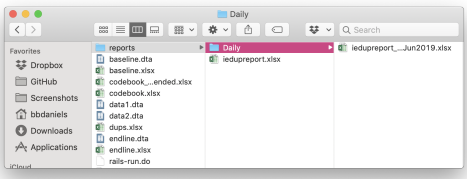
Why Excel? Because it is easier for everyone to read and understand when there are large numbers of information to process, rather than de-coding Stata code.

`ieduplicates`: Listing duplicates in data

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

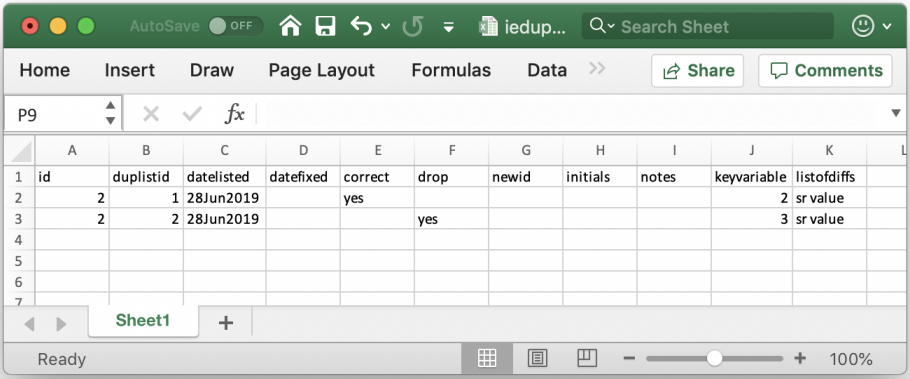
- Lists all duplicates in data into a file called iedupreport.xlsx and backs up a dated copy
- Removes all copies of duplicates from the data so other quality-assurance tasks like back-checks can be performed on unambiguous portion of data

```
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.



	A	B	C	D	E	F	G	H	I	J	K
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											

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.

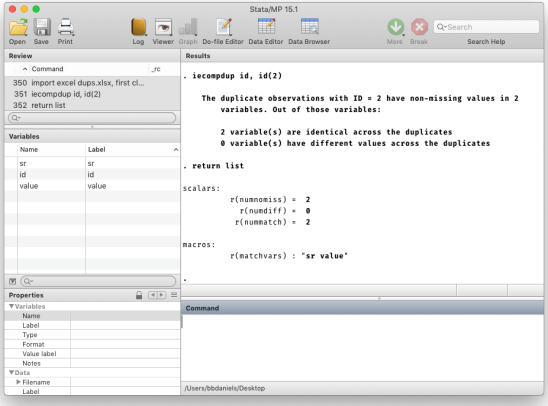
`iecompdup`: Analyzing duplicates in data

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



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`

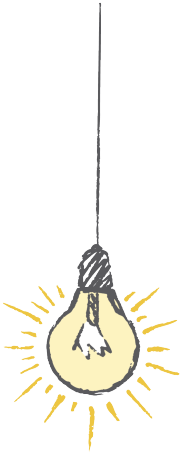
`iecodebook apply`: Data cleaning made easy

`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:

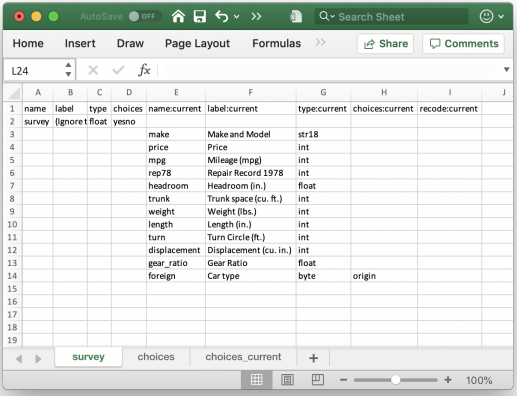
```
iecodebook apply
using "/path/to/codebook.xlsx"
```



`iecodebook apply`: Setting up a template

```
// Load data
sysuse auto.dta , clear

// Create cleaning template
iecodebook template
  using "/path/to/codebook.xlsx"
```



	name	label	type	choices	name:current	label:current	type:current	choices:current	recode:current
2	survey	(ignore t	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					foreign	Car type	byte	origin	

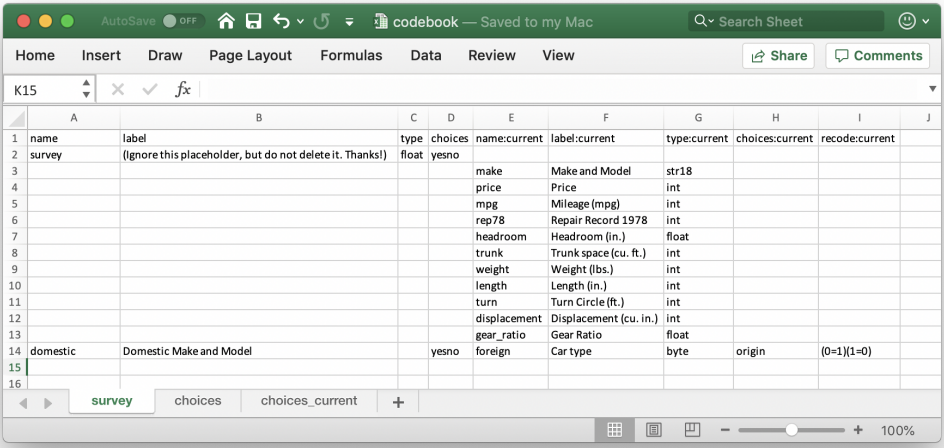
The `template` subcommand sets up the spreadsheet based on the data in memory.

`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 recieved 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



	A	B	C	D	E	F	G	H	I	J
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					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										

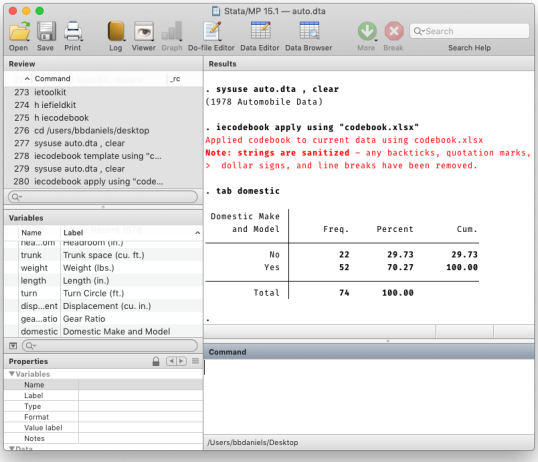
`iecodebook apply`: Applying the spreadsheet codebook to the data

```
// 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 *(required)*	What is your name?	
quest *(required)*	What is your quest?	
airspeed *(required)*	What is the average airspeed of an unladen swallow?	
color *(required)*	What is your favorite color?	1 Red 2 Blue 3 Green

DIME Training Endline

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

`iecodebook append`: Data combination

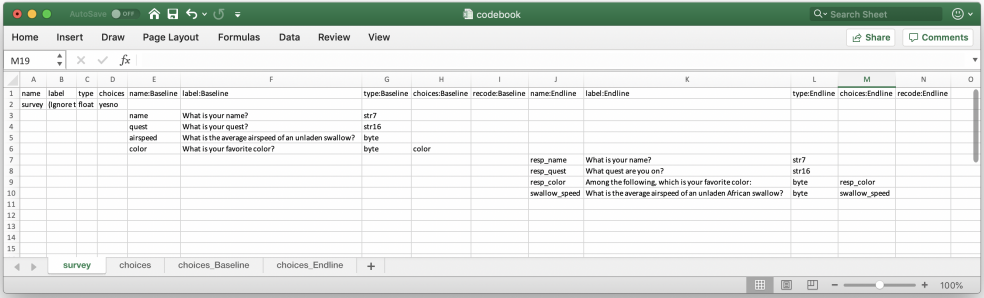
`iecodebook append` runs an arbitrary number of `rename`, `recode`, and `label` commands on two or more datasets with the intention of harmonizing them. It then runs an `append` command on the harmonized data.

- 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)
```

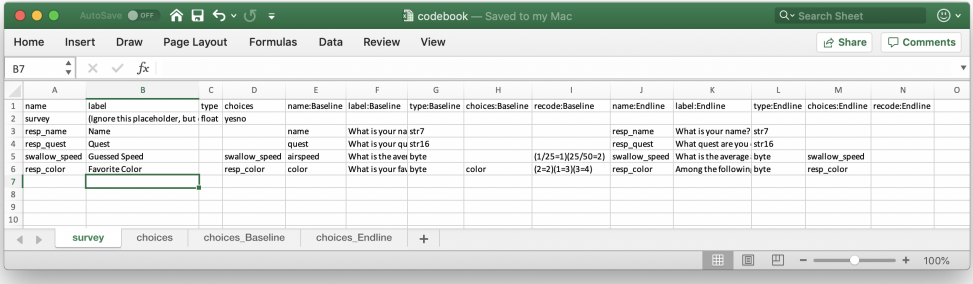
The `template` subcommand sets up the spreadsheet based on multiple datasets on disk. The `surveys()` option names the datasets in the template and is required.

`iecodebook append`: Setting up a spreadsheet template



	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 t	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															

`iecodebook append`: Filling out the template



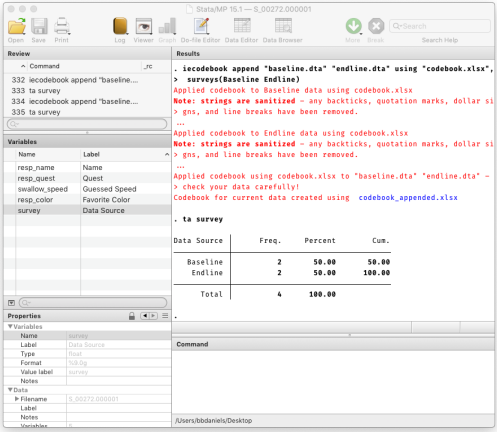
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	name	label	type	choices	name:Baseline	label:Baseline	type:Baseline	choices:Baseline	recode:Baseline	name:Endline	label:Endline	type:Endline	choices:Endline	recode:Endline	
2	survey	(Ignore this placeholder, but	float	yesno											
3	resp_name	Name			name	What is your 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 aver	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 fa	byte	color	(2=2)(1=3)(3=4)	resp_color	Among the followin	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