Exercises.md - Grip

Note: This exercise is intended to familiarize you with the `high-frequency-check` templates. All files (except for the ado files) are located here in the `exercise` folder in the root directory.

Jump to: Installation - Background - The Data - Instructions

Installation

In order to complete the exercise you will need to have Stata 13.0 or higher. You will also need to have the `ipacheck.pkg` package installed (for help see the installation page) and both `readreplace` and `cfout` .

Background

A **high-frequency check** is a check of some element of the data collection process, completed on a regular basis as new data comes in. At IPA, high-frequency checks are typically implemented in Stata, after the data flow is complete. High-frequency checks can provide information about any of the following elements of data collection:

- The quality of the data
- Enumerator performance
- The DDC survey program (are there programming errors?)
- The data flow (are there systemic flaws?)

Given how much information they can provide about the quality of the data collection, high-frequency checks are one of the major benefits of DDC. It's hard to overstate how important these checks are. High-frequency checks are different from DDC logic checks, which are programmed into the DDC survey program and not in Stata. Typically, high-frequency checks are checks that can't be implemented in a DDC program. For instance, while DDC logic checks are restrictions on a field or the relationship between fields within a survey, high-frequency checks often check trends across surveys.

IPA recommends the following minimum checks be included in any high frequency check files (**Note:** these are all included in the `high-frequency-check` template)

Daily logic checks

1. Check that all interviews were completed
2. Check that there are no duplicate observations
3. Check that all surveys have consent
4. Check that certain critical variables have no missing values
5. Check that follow up record ids match original
6. Check skip patterns and constraints
7. Check that no variable has all missing values
8. Check hard/soft constraints
9. Check specify other vars for items that can be included
10. Check that date values fall within survey range
11. Check that there are no outliers for unconstrained vars

Enumerator checks (dashboard)

1. Check the percentage of "don't know" and "refusal" values for each variable by enumerator
2. Check the percentage giving each answer for key filter questions by enumerator

3. Check the percentage of survey refusals by enumerator
4. Check the number of surveys per day by enumerator
5. Check average interview duration by enumerator
6. Check the duration of consent by enumerator
7. Check the duration of other (anthropometrics, games, etc)

Research checks (dashboard)

1. Survey progress towards recruitment goals.
2. Summary of key research variables.
3. Two way summaries of survey variables by demographic/geographic characteristics.
4. Refusal/not found rates by treatment status.
5. Maps/GIS

The Data

We will be working with the `survey_data.dta` data set. This data is adapted from a baseline survey in Zambia. The data were collected using SurveyCTO and imported to Stata using `odkmeta`. All PII have been removed and the gps points have been anonymized. Open the data set in Stata and take a moment to familiarize yourself with the variables.

Instructions

Section 1: Template Overview

1. Open Stata and navigate to the **high-frequency-checks** folder. You can use the `cd` command, e.g.

```
cd "C:/your/directory/path/here/high-frequency-checks"
```

2. Now that we're in the proper directory. Let's take a look at the files in the template. You can either use the file browser or simply type `ls` in the command window. Note the following files:
   - **`hfc_inputs.xlsx`** - this is the input Excel file; inside you'll find a convenient form for configuring the HFC commands.
   - **`hfc_replacements.xlsx`** - the is the replacement Excel file; it is a running list of edits/corrections based on HFC outputs; these replacements can be automatically added to your workflow using `readreplace`.
   - **`hfc_outputs.xlsx`** - this is the output Excel file; all check violations will be listed here after running `master_check.do`.
   - **`hfc_enumerators.xlsx`** - this is the enumerator dashboard file; it contains summary survey statistics by enumerator and can be useful for field management/incentives.
   - **`master_check.do`** - this is the master do-file; it reads the inputs, makes any replacements, runs the HFC checks, outputs violations and produces the dashboard.
3. Open the `master_check.do` file and review the major sections: local definitions; pre-process import data; import globals from Excel; replacements and corrections; high frequency checks; additional user checks; create enumerator dashboard; and research dashboard.

```
doedit master_check.do
```

   Try to understand what each section does: most sections are set up to run automatically, but it helps to have an idea of what is going on.

Section 2: Configure the Inputs

1. Open the `hfc_inputs.xlsx` file in Excel. This is where you'll configure each of the 12 daily logic checks. Each logic check has its own worksheet. You'll notice, we've tried to make the input file more user-friendly by adding automatic formatting and help boxes in each sheet. In this exercise we're going to configure checks for the `survey_data.dta` data set.

2. Open the `1. incomplete` sheet and review the help boxes. This check verifies that all surveys have been completed. It corresponds with the `ipacheckcomplete` command in the `master_check.do` file. **For our data set, a value of 2 in the variable `intstatus` indicates a complete interview. Update the `variable` and `complete_value` columns to reflect this.**

3. Open the `2. duplicates` sheet and review the help boxes. This check verifies that there are no duplicate surveys. The inputs are loaded to the `ipacheckdups` command in the `master_check.do` file. **For our data set, the variable `id` should contain no duplicates. Update the `variable` column to reflect this.**

4. Open the `3. consent` sheet and review the help boxes. This check verifies that all surveys have consent. The inputs are loaded to the `ipacheckconsent` command in the `master_check.do` file. **For our data set, the value 1 for variables `consent` and `consentsign` indicate consent. Update the `variable` and `consent_value` columns to reflect this.**

5. Open the `4. no miss` sheet and review the help boxes. This check verifies that certain variables have no missing values. The inputs are loaded to the `ipachecknomiss` command in the `master_check.do` file. **For our data set, the variables `gpsLatitude, gpsLongitude, enumid, starttime, endtime, SubmissionDate, ward, gender, age` should have no missing values. update the `variable` column to reflect this.**

6. Open the `5. follow up` sheet and review the help boxes. This check verifies that follow up data matches data in the master tracking sheet. The inputs are loaded to the `ipacheckfollowup` command. **This survey was not a follow up so this check is not relevant. You can leave it blank.**

7. Open the `6. skip` sheet and review the help boxes. This check verifies survey logic and skip patterns. The inputs are loaded to the `ipacheckskip` command. **Update the `variable`, `assert`, and `if_condition` columns with the logic checks in the table below.**

variable	assert	if_condition
pregnant	pregnant==.	gender==0
salary	salary==.	employyear==4
occupation	occupation==.	employyear==4
employyear	employmt==0	employyear==4

8. Open the `7. all miss` sheet and review the help boxes. This check verifies that certain variables are not all missing. The inputs are loaded to the `ipacheckallmiss` command in the `master_check.do` file. **For our data set, check all survey variables to see if any are all missing.**

9. Open the `8. constraints` sheet and review the help boxes. This check verifies hard and soft constraints. The inputs are loaded to the `ipacheckconstraints` command in the `master_check.do` file. **Update the `variable`, `soft_min`, and `soft_max` columns with the logic checks in the table below.**

variable	soft_min	soft_max
age	18	24
salary	0	10000
childnum	0	5

10. Open the `9. specify` sheet and review the help boxes. This check lists all non-missing specify other values to identify possible recodes or new categories. The inputs are loaded to the `ipacheckspecify` command. **Update the `specify_variable` column with all specify other variables.**

11. Open the `10. dates` sheet and review the help boxes. This check looks for common survey date errors. The inputs are loaded to the `ipacheckdates` command. **Update the `startdate`, `enddate`, and `surveystart` columns with the data in the table below.**

startdate	enddate	surveystart
startdate	enddate	11/01/2015

12. Open the `11. outliers` sheet and review the help boxes. This check looks for potential outliers in continuous variable values. The inputs are loaded to the `ipacheckoutliers` command. **For our data set, we define a value 3.0 times the interquartile range as an outlier for the variables `salary` and `childnum` indicate consent. Update the `variable` and `iqr_multiplier` columns to reflect this.**

13. Open the `enumdb` sheet and review the help boxes. This check creates the enumerator dashboard (`hfc_enumerators.xlsx`). The inputs are loaded to the `ipacheckenum` command. **Update the columns with the data in the table below.**

dkrf_variable	missing_variable	duration_variable	exclude_variable	submission_date
eduattain	consent	ta_*	SubmissionDate	SubmissionDate
occupation	consentsign		starttime	
employyear	gpsLatitude		endtime	
childnum	gpsLongitude		ta_*	
intstatus				

Section 3: Run the HFCs

1. Now that the inputs are set, we are ready to run the HFCs. Save the input file and open the `master_check.do` file.

2. Normally, at this point we'd need to make sure the top section (the local definitions and the pre-processing) of the `master_check.do` file is correctly specified and matches the survey input. In this case these have already been done for you.

3. Run the `master_check.do` file by either clicking the icon at the top of the do-file editor or by typing the following in the command window.

```
do master_check.do
```

Section 4: Review the Output

1. Once `master_check.do` has finished running, you should have an updated `hfc_output.xlsx` available. This file contains lists of check violations encountered by the HFC program. Open this file and inspect the contents. You'll notice it is arranged in the same format as the input with a separate sheet for each check. The output also includes a summary with overall violation counts.

2. Using the `hfc_output.xlsx` file. Answer the following questions:

    - **How many interviews have been conducted?**
    - **How many incomplete interviews are there? Inspect any incomplete observations and see if you can figure out what is going on. (Hint use the `list` command)**
    - **How many duplicates are there? Inspect these duplicates and see if you can figure out what is going on.**
    - **How may variables have missing values that shouldn't be missing? What can we do about these?**
    - **How many skip pattern/logic violations are there? What can be done to prevent/resolve these?**
    - **How many constraint violations are there? Do any values appear to be nonsensical? What should be done?**
    - **Do you see any specify options that could be recoded or new categories?**

3. Open the `hfc_enumerators.xlsx` file and inspect the contents. **Do you notice any significant differences between the enumerators? What should be done in response to these findings?**

Section 5: Make Replacements

1. Open the `hfc_replacements.xlsx` workbook and read the help boxes. This file is used to make batch corrections/edits to the survey data set based on errors or violations found via the HFC template. You may notice that it has the same column headings as the output file... this is by design. You can copy output violations directly into the replacement file and make corrections by updating the `newvalue` column if/when a solution is discovered after consultation with the field team.

2. After communicating the output of the HFCs with your field teams you discover that several values of the `salary` variable that were caught by the constraint and outlier checks were incorrectly input. Back check data confirms that the enumerator entered an extra zero. **To update the survey data enter the corrections below in the replacements file. Don't forget you can copy and paste the first columns from the constraints sheet of the output file.**

id	enumerator	variable	label	value	message	notes	drop	newvalue
510	36	salary	...	15000	...			1500
640	25	salary	...	20000	...			2000
877	23	salary	...	-999	...			.d
1004	24	salary	...	23000	...			2300
1612	35	salary	...	13000	...			1300
1671	34	salary	...	12000	...			1200

3. Open the `master_check.do` file and scroll down to the replacements section. You'll notice the following line is currently commented out.

```
/*
readreplace using "hfc_replacements.xlsx", ///
id("id") ///
variable("variable") ///
value("newvalue") ///
excel ///
import(firstrow)
*/
```

   This line makes the replacements in the `hfc_replacements.xlsx` file by calling the `readreplace command` . Uncomment these lines so that the replacements in item 2. can be made by removing `/*` and `*/` lines. Save the updated `master_check.do` file.

Section 6: Re-run the HFCs

1. Re-run the `master_check.do` and verify that the replacements were made and the errors no longer appear in the output file. **Inspect the output file for other potential replacements that can be made and add them to the list.**