ietoolkit

What we have learned from working with 100+ researchers assistants and field coordinators

Kristoffer Bjarkefur, Luiza Cardoso de Andrade, Benjamin Daniels

July 11, 2019

Development Impact Evaluation (DIME)
The World Bank Group
What does DIME Analytics do?
What is DIME?

- DIME is the department for impact evaluations at the World Bank
- Currently 203 Impact Evaluations in 52 countries
- Currently 70 Research Assistants (RAs) and Field Coordinators (FCs)
What is DIME Analytics?

- DIME Analytics is a part of DIME. We support the research teams within DIME and develop data work resources for those teams.
- We support in the day-to-day data work, to share experiences across the team and off-load the economists.

- Kristoffer Bjarkefur - Data Coordinator
- Luiza Cardoso de Andrade - Data Coordinator
- Benajamin Daniels - Data Coordinator
- Maria Jones - Survey Specialist
- Roshni Khincha - Data Coordinator
- Mrijan Rimal - Data Scientist
DIME Analytics’ resources:

- Data for Development Impact - https://worldbank.github.io/d4di/
- ietoolkit - ssc install ietoolkit
- iefieldkit - ssc install iefieldkit

Everything we develop we share publicly!
Institutional memory in code
Primary data -> Difficult!

- Almost all data used in DIME is **primary data** that we collect in **developing countries**
- *Primary data is difficult* and working in developing countries is often difficult
- We love that challenge, but what can DIME Analytics do to **make what’s difficult easier**? And what can we do to **prevent hard tasks from leading to errors**?
Collectively DIME has a lot of experience, DIME Analytics’ tasks boils down to generate and disseminate institutional memory. The wiki and the book are obviously important resources for institutional memory. But what if we can also build institutional memory into the code that the RAs and FCs use? This is the objective of ietoolkit and iefieldkit, and the topic of this presentation.
Institutional memory in code - What does it mean? What is it a solution to?

1. Make people use the collective experience when coding even if they are not aware that they are using it

2. Automatize away human error. Make humans spend their time on what is - still - their comparative advantage

3. Also applicable on tasks most users think they already do really well. But in reality, many of those tasks are often much more difficult or time consuming to do well than what must users think
Coding for the mass market
What is different when you share institutional memory through code?

- Most Stata commands are made by expert users mostly for other expert users
- Harmonization, simplification and automation of less advanced tasks are also useful
- Many of DIME Analytics’ commands solve tasks that most users were already able to solve in their own way
- It is not about solving a new problem, but remembering all best practices related to an old problem
Uptake matters!

- How are majority users different from expert users like you and me?
- We have worked with 100+ of them...
Easier matters more than better

• Make the task **easier and faster** to implement with the command

• Both DIME Analytics and the RAs/FCs want better quality data, but what drives the willingness to change behavior is very different

• **Making something easier and faster to implement is a much more effective way to change behavior.**

• It is very hard to make majority users change behavior only based on the aspect of improved data quality
Rely on error messages instead of the help file

- You and I read help files, but the **majority user does not read help files**.
- **The majority user reads error messages**
  - Helpful error messages increases uptake
  - If the error message does not provide an immediate solution, you risk losing the user’s attention right there
- **Links in error messages to help files is a good compromise**
- **There are too many uninformative error messages in the Stata world!**
  - The majority user is not willing or able to debug errors like `invalid syntax` - it takes them too much time and effort
No excuse to not write help files

- This is not an excuse to not write an help file
- You should still write help files, as some people read them!
- The point is that help files are often not enough no matter how good and informative it is
Test the input extensively

Write your command with these assumptions:

1. Users do not read your help file
2. Users figure out new commands though trial and error, not documentation
3. They will blame your command if it does not work unless they are provided with a on-screen solution to the problem

Solution:

- Test the input specifications extensively and provide helpful error messages
- Test the input data extensively and provide helpful error messages
- Ask yourself, *if my command would be used incorrectly, would the user always find that out?*
Feedback makes a good command great
"Helpful" and "easy" are subjective

- What is easier to implement?
- What is a helpful error message?

*Helpful* and *easy* are subjective. Expert users like you and me will never be the best judge of what is easy and what is helpful.
I would like to hear examples of methods to get feedback, as we do not have a silver bullet for how to get feedback.

- We are lucky as we work closely with a lot of users and we drag feedback out of them
- We use GitHub https://github.com/worldbank/ietoolkit
- We have our email dimeanalytics@worldbank.org everywhere
Statalist is great and has taught me much of what I know.

But from the perspective of this presentation, most comments there are made by users like you and me, and we are only one dimension of the conversation.

I think that Stata could become an even better product if the expert users - who dominates the conversation - understood the majority users better. Can the Statalist play a role in that?
Example: ietoolkit
ietoolkit - a package full of institutional memory

- Example of outcome from our work on disseminating *institutional memory in code*: ietoolkit

- We have identified tasks that frequently lead to errors, and where we have found it applicable, we have created commands that are built with the collective experience of DIME in mind
- Any RAs and FCs can create folders. But a surprising amount of errors come from poorly organized folders. iefolder provides a solution to that.

- The point is not that everyone should use our folder structure. The point is that there are huge data quality gains in big teams from systematizing how data work folders are set up.
Most RAs and FCs can run regressions and use packages like `estout` to create balance tables.

We combined these into a single command with options for advanced specifications:

```bash
iebaltab divorce marriage death medage, ///
grpvar(treatment) ///
savetex("$folder/iebaltab-ex-latex.tex")
```

<table>
<thead>
<tr>
<th>Variable</th>
<th>Control</th>
<th>Treatment</th>
<th>T-test Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>divorce</td>
<td>26</td>
<td>24</td>
<td>-5647.631</td>
</tr>
<tr>
<td></td>
<td>20968.577 (3504.741)</td>
<td>26616.208 (6380.678)</td>
<td></td>
</tr>
<tr>
<td>marriage</td>
<td>26</td>
<td>24</td>
<td>-6812.212</td>
</tr>
<tr>
<td></td>
<td>44431.538 (7316.072)</td>
<td>51243.750 (10803.806)</td>
<td></td>
</tr>
<tr>
<td>death</td>
<td>26</td>
<td>24</td>
<td>-2274.420</td>
</tr>
<tr>
<td></td>
<td>38382.538 (8035.616)</td>
<td>40656.958 (8861.143)</td>
<td></td>
</tr>
<tr>
<td>medage</td>
<td>26</td>
<td>24</td>
<td>0.117</td>
</tr>
<tr>
<td></td>
<td>29.596 (0.421)</td>
<td>29.479 (0.213)</td>
<td></td>
</tr>
</tbody>
</table>

F-test of joint significance (F-stat) 0.509
F-test, number of observations 50

Notes: The value displayed for t-tests are the differences in the means across the groups. The value displayed for F-tests are the F-statistics. ***, **, and * indicate significance at the 1, 5, and 10 percent critical level.
- Make sure no strange settings are used in any team members’ session of Stata

- The example that is most important to us who need replicable randomization, is that it make sure that the full team sets the same version
Summary
Summary:

- Code can also be used to spread best practices and institutional memory across an institution or a big team
- Commands does not have to be cutting edge or introduce something novel to be very helpful to majority users
- The majority users have a very short attention span
Scientific advances are the result of a long, cumulative process of building knowledge and methodologies – or, as the cliché goes, “standing on the shoulders of giants”. One often overlooked, but crucial part of this climb is a long tradition of standardization of everything from mathematical notation and scientific terminology, to format for academic articles and references.
For more information or further questions please contact:
Kristoffer Bjarkefur (kbjarkefur@worldbank.org)
DIME Analytics (dimeanalytics@worldbank.org)
Thank You!