DATA ANALYSIS TIPS and CONSIDERATIONS

Professor C. Leigh Anderson, Principal Investigator
Professor Travis Reynolds, co-Principal Investigator

January 3, 2018

EPAR conducts data analyses when we have access to publicly-available datasets with information relevant to our research questions. Below we include a series of data analysis tips and considerations drawing from our experience with data analysis projects, which you can explore on our website. In addition, we include a series of links to other helpful resources for research and statistical analysis.

Data Analysis Tips and Considerations:

- Find background information about your dataset and how the data were collected. Include relevant information about the dataset along with any results that you produce.
- Review the instrument used to collect the data, and look through the enumerator guide and metadata (if provided) to better understand what is included in the data.
- If you will be using multiple datasets, understand what each contains, how they can be linked to one another (what are the unique identifiers), and the different levels of analysis.
- Identify exactly which pieces of data are needed to answer your question, and understand how they are organized and how to interpret them. Write out how you will need to modify the raw data to create the variables you need for your analysis before writing the code to construct the variables, and consider the potential tradeoffs in your variable construction decisions.
- Browse the raw data for all variables of interest to identify any anomalies that might need correction (ex: missing values, data entry errors, etc.).
- In general some amount of data cleaning will be needed. Cleaning decisions should first follow common sense, such as not allowing nonsensical values like more than 24 hours in a day. Consider tradeoffs in how you will approach cleaning the data – replacing a nonsensical value with a "missing" value will have a different effect on your results than replacing it with the highest possible logical value or with the median value, for example. Clearly document all of your cleaning decisions, and try to be consistent in how you apply those decisions across variables.
- For continuous variables, consider how you will approach dealing with outliers (on either the high or low end of the distribution). Common approaches to identifying outliers include looking at percentiles of the distribution (e.g., observations below the 1st percentile or above the 99th percentile), looking at standard deviations from the median (e.g., observations more than 2 standard deviations away from the median), or determining a maximum or minimum possible threshold (based on previous research or commonly-accepted thresholds for given variables). You should then consider how to trim those outliers, and use the same approach consistently. Common approaches to trimming outliers include winsorizing (replacing observations above/below a threshold with the value at that threshold), median absolute deviation (MAD; replacing outlier observations with the median value from the distribution),

EPAR uses an innovative student-faculty team model to provide rigorous, applied research and analysis to international development stakeholders. Established in 2008, the EPAR model has since been emulated by other UW schools and programs to further enrich the international development community and enhance student learning.

Please direct comments or questions about this research to Principal Investigators Leigh Anderson and Travis Reynolds at eparinfo@uw.edu.

or replacing outliers with "missing" values. Treatment of outliers can significantly influence your results, so these decisions should be clearly documented in your code and your research methods.

- Another consideration for dealing with outliers is when to apply your outlier trimming decisions. EPAR generally trims only the final constructed variables (e.g. a ratio constructed from two variables in the data), rather than trimming outliers for intermediate variables instead of or in addition to the final constructed variables, following common FAO and World Bank practice.
- Use a syntax file (such a Stata .do file or R .R script) to create and edit code for your analysis rather than directly entering code into the command interface or using the menu options for your statistical software package. This will allow you to keep a detailed record of how you prepared your variables and conducted your analysis, which can support collaborative data analysis and also increase transparency around research methods.
- Include comments and notes in your syntax file, so that an outside reviewer could follow the code and your decisions. Add headings to make it easier to follow the structure of the syntax file – particularly for long files creating many variables and conducting a variety of analyses. Create labels for any variables you create to make your final dataset more accessible to a new user.
- Following these documentation protocols will also help you to retrace your decisions later in the analysis stages, and to go back and look for potential sources of error or odd results.
- If possible, consider using a single syntax file with all the code for your variable construction and analysis, so that an outside user can trace through all the steps taken to arrive at a given result.
- Use meaningful labeling when naming variables to make it easier to follow what they represent, without always needing to refer to the variable label.
- If the data include survey weights that can be used to make the sample data representative of an entire population, make sure to use the weights when producing estimates or running analyses.
- Use global or local commands to specify file paths or groupings of variables to simplify your code.
- After completing your code and running your analyses, scan your results to see if the figures make sense, and look for potential coding errors or decisions that may have led to any unexpected or odd results. In particular, consider potential effects of your data cleaning and outlier trimming decisions. Before formalizing your results into a report or other output, read back through your code to check for any potential errors.
- Consider commands that may facilitate the process of exporting the results of your analyses. For example, in Stata we frequently use PutExcel to export results (for example, summary statistics, graphs, or matrices) to an Excel workbook, and EstOut to export regression or summary statistics tables in a publication-ready format.
- If you are not sure the best way to achieve a certain goal in your data analysis with the software tool you are using, turn to Google. There are many active discussion forums with helpful guides and chunks of code that may be relevant to a given coding problem.
- If you are working on a data analysis project with a group, consider using a GitHub repository for version control. GitHub allows you to keep track of code changes in project files, include who made changes and what changes were made. Create a private repository for your project on GitHub online, and use GitHub Desktop to clone that repository to your local machine. Then, use GitHub Desktop to continuously update the local clone repository to reflect others' changes and to push your changes to the master online repository. See our Software Tools Resources for more information on using GitHub for version control, and for using GitHub to share a citable version of your project code.

Resources for Research and Statistical Analysis:

- University of Washington Center for Studies in Demography and Ecology (CSDE): CSDE supports provides training and resources for data analyses, including occasional courses and workshops on Stata, R, and GIS.
- University of Washington Center for Statistics and the Social Sciences (CSSS): CSSS provides free statistical consulting to current UW faculty, staff, and students working on social science problems, at any stage in the research process. They also offer courses in various aspects of statistical analysis at the undergraduate and graduate levels.
- Institute for Digital Research and Education (IDRE), UCLA: IDRE provides useful guides on learning and using Stata, the primary statistical software used by EPAR.
- UNC Carolina Population Center Stata Tutorial: Includes function-oriented guides for using Stata, focusing on the data-management tasks most needed by data analysts working with sample survey data. It works up from basic tasks, such as how to drop variables, to the tasks needed for complex file organization, such as how to reshape and merge data files. The examples in the guides can be followed by downloading the provided sample data files that are available
- StataCorp provides a variety of community-contributed and official resources for leaning Stata.
- A variety of websites provide free tutorials for using R: Cyclismo, R-tutor, R-bloggers, TryR, etc.
- Some paid websites offer a set number of free tutorials for different software packages, including interactive assignments and coding examples: DataCamp, Code School, etc.
- The GitHub Guide provides a helpful overview of how to use GitHub for version control. A StataCorp slideshow presents a guide to using GitHub for version control with Stata code.
- Abdul Latif Jameel Poverty Action Lab (J-PAL): J-PAL and Innovations for Poverty Action (IPA) curate a research resources page that includes best practices for data and code management and other resources covering research design, measurement and data collection, working with data, transparency, randomization, and software and tools.