The Workflow of Data Analysis: Principles and Practice

The Workflow of Data Analysis: Principles and Practice

J. SCOTT LONG
Department of Sociology
Indiana University–Bloomington

A Stata Press Publication
StataCorp LP
College Station, Texas

Published by Stata Press, 4905 Lakeway Drive, College Station, Texas 77845
Typeset in LaTeX 2_ε
Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

ISBN-10: 1-59718-047-5
ISBN-13: 978-1-59718-047-4

To Valerie

Contents

Tables

Figures

Preface

This book is about methods that allow you to work efficiently and accurately when you analyze data. Although it does not deal with specific statistical techniques, it discusses the steps that you go through with any type of data analysis. These steps include planning your work, documenting your activities, creating and verifying variables, generating and presenting statistical analyses, replicating findings, and archiving what you have done. These combined issues are what I refer to as the *workflow of data analysis*. A good workflow is essential for replication of your work, and replication is essential for good science.

My decision to write this book grew out of my teaching, researching, consulting, and collaborating. I increasingly saw that people were *drowning* in their data. With cheap computing and storage, it is easier to create files and variables than it is to keep track of them. As datasets have become more complicated, the process of managing data has become more challenging. When consulting, much of my time was spent on issues of data management and figuring out what had been done to generate a particular set of results. In collaborative projects, I found that problems with workflow were multiplied. Another motivation came from my work with Jeremy Freese on the package of Stata programs known as SPost (Long and Freese 2006). These programs were downloaded 10,000 times last year, and we were contacted by hundreds of users. Responding to these questions showed me how researchers from many disciplines organize their data analysis and the ways in which this organization can break down. When helping someone with what appeared to be a problem with an SPost command, I often discovered that the problem was related to some aspect of the user's workflow. When people asked if there was something they could read about this, I had nothing to suggest.

A final impetus for writing the book came from Bruce Fraser's *Real World Camera Raw with Adobe Photoshop CS2* (2005). A much touted advantage of digital photography is that you can take a lot of pictures. The catch is keeping track of thousands of pictures. Imaging experts have been aware of this issue for a long time and refer to it as *workflow*—keeping track of your work as it flows through the many stages to the final product. As the amount of time I spent looking for a particular picture became greater than the time I spent taking pictures, it was clear that I needed to take Fraser's advice and develop a workflow for digital imaging. Fraser's book got me thinking about data analysis in terms of the concept of a workflow.

After years of gestation, the book took two years to write. When I started, I thought my workflow was very good and that it was simply a matter of recording what I did. As writing proceeded, I discovered gaps, inefficiencies, and inconsistencies in what I did.

Sometimes these involved procedures that I knew were awkward, but where I never took the time to find a better approach. Some problems were due to oversights where I had not realized the consequences of the things I did or failed to do. In other instances, I found that I used multiple approaches for the same task, never choosing one as the best practice. Writing this book forced me to be more consistent and efficient. The advantages of my improved workflow became clear when revising two papers that were accepted for publication. The analyses for one paper were completed before I started the workflow project, whereas the analyses for the other were completed after much of the book had been drafted. I was pleased by how much easier it was to revise the analyses in the paper that used the procedures from the book. Part of the improvement was due to having better ways of doing things. Equally important was that I had a consistent and documented way of doing things.

I have no illusions that the methods I recommend are the best or only way of doing things. Indeed, I look forward to hearing from readers who have suggestions for a better workflow. Your suggestions will be added to the book's web site. However, the methods I present work well and avoid many pitfalls. An important aspect of an efficient workflow is to find one way of doing things and sticking with it. Uniform procedures allow you to work faster when you initially do the work, and they help you to understand your earlier work if you need to return to it at a latter time. Uniformity also makes working in research teams easier since collaborators can more easily follow what others have done. There is a lot to be said in favor of having established procedures that are documented and working with others who use the same procedures. I hope you find that this book provides such procedures.

Although this book should be useful for anyone who analyzes data, it is written within several constraints. First, Stata is the primary computing language because I find Stata to be the best, general purpose software for data management and statistical analysis. Although nearly everything I do with Stata can be done in other software, I do not include examples from other packages. Second, most examples use data from the social sciences, because that is the field in which I work. The principles I discuss, however, apply broadly to other fields. Finally, I work primarily in Windows. This is not because I think Windows is a better operating system than Mac or Linux, but because Windows is the primary operating system where I work. Just about everything I suggest works equally well in other operating systems, and I have tried to note when there are differences.

I want to thank the many people who commented on drafts or answered questions about some aspect of workflow. I particularly thank Curtis Child, Nadine Reibling, and Tait Runnfeldt Medina whose detailed comments greatly improved the book. I also thank Alan Acock, Art Alderson, Myron Gutmann, Patricia McManus, Eliza Pavalko, Jack Thomas, Leah VanWey, Rich Watson, Terry White, and Rich Williams for talking with me about workflow. David M. Drukker at StataCorp answered many of my questions. His feedback made it a better book and his friendship made it more fun to write. Some of the material in this book grew out of research funded by NIH Grant Number R01TW006374 from the Fogarty International Center, the National Institute

of Mental Health, and the Office of Behavioral and Social Science Research to Indiana University–Bloomington. Other work was supported by an anonymous foundation and The Bayer Group. I gratefully acknowledge support provided by the College of Arts and Sciences at Indiana University.

Without the unintended encouragement from my dear friend Fred, I would not have started the book. Without the support of my dear wife Valerie, I would not have completed it. Long overdue, this book is dedicated to her.

A word about fonts, files, commands, and examples

The book uses standard Stata conventions for typography. Things printed in a typewriter-style typeface are Stata commands and options. For example, `use mydata, clear`. Italics indicate information that you should add. For example, `use` *dataset-name*`, clear` indicates that you should substitute the name of your dataset. When I provide the syntax for a command, I generally show only some of the options. For full documentation, you can type `help` *command-name* or check the reference manual. Manuals are referred to with the usual Stata conventions. For example, [R] **logit** refers to the `logit` entry in the *Base Reference Manual* and [D] **sort** refers to the `sort` entry in the *Data Management Reference Manual*.

The book includes many examples that I encourage you to try as you read. If the name of a file begins with `wf` you can download that file. I use (file: *filename*`.do`) to let you know the name of the do-file that corresponds to the example being presented. With few exceptions (e.g., some ado-files), if the name of a file does not begin with `wf` (e.g., `science2.dta`), the file is not available for download. To download the examples, you must be in Stata and connected to the Internet. There are two Workflow packages for Stata 10 (`wf10-part1` and `wf10-part2`) and two for Stata 9 (`wf09-part1` and `wf10-part1`). To find and install the packages, type `findit workflow`, choose the packages you need, and follow the instructions. Although two packages are needed because of the large number of examples, I refer to them simply as the Workflow package. Before trying these examples, be sure to update your copy of Stata as described in [GS] **20 Updating and extending Stata—Internet functionality**. Additional information related to the book is located at http://www.indiana.edu/~jslsoc/workflow.htm.

Bloomington, IN Scott Long
August 2008

1 Introduction

This book is about methods for analyzing your data effectively, efficiently, and accurately. I refer to these methods as the *workflow of data analysis*. Workflow involves the entire process of data analysis including planning and documenting your work, cleaning data and creating variables, producing and replicating statistical analyses, presenting findings, and archiving your work. You already have a workflow, even if you do not think of it as such. This workflow might be carefully planned or it might be ad hoc. Since workflow for data analysis is rarely described in print or formally taught, researchers often develop their workflow in reaction to problems they encounter and from informal suggestions from colleagues. For example, after you discover two files with the same name but different content, you might develop procedures (i.e., a workflow) for naming files. Too often, good practice in data analysis is learned inefficiently through trial and error. Hopefully, my book will shorten the learning process and allow you to spend more time on what you really want to do.

Reactions to early drafts of this book convinced me that both beginners and experienced data analysts can benefit from a more formal consideration of how they do data analysis. Indeed, when I began this project, I thought that my workflow was pretty good and that it was simply a matter of writing down what I routinely do. I was surprised and pleased by how much my workflow improved as a result of thinking about these issues systematically and from exchanging ideas with other researchers. Everyone can improve their workflow with relatively little effort. Even though changing your workflow involves an investment of time, you will recoup this investment by saving time in later work and by avoiding errors in your data analysis.

Although I make many specific suggestions about workflow, most of the things that I recommend can be done in other ways. My recommendations about the best practice for a particular problem are based on my work with hundreds of researchers and students from all sectors of employment and from fields ranging from chemistry to history. My suggestions have worked for me and most have been refined with extensive use. This is not to say that there is only one way to accomplish a given task or that I have the best way. In Stata, as in any complex software environment, there are a myriad of ways to complete a task. Some of these work only in the limited sense that they get a job done but are error prone or inefficient. Among the many approaches that work well, you will need to choose your preferred approach. To help you do this, I often discuss several approaches to a given task. I also provide examples of ineffective procedures since seeing the consequences of a misguided approach can be more effective than hearing about the virtues of a better approach. These examples are all real, based on mistakes I made (and

I have made lots) or mistakes I encountered when helping others with data analysis. You will have to choose a workflow that matches the project at hand, the tools you have, and your temperament. There are as many workflows as there are people doing data analysis, and there is no single workflow that is ideal for every person or every project. What is critical is that you consider the general issues, choose your own procedures, and stick with them unless you have a good reason to change them.

In the rest of this chapter, I provide a framework for understanding and evaluating your workflow. I begin with the fundamental principle of replicability that should guide every aspect of your workflow. No matter how you proceed in data analysis, you must be able to justify and reproduce your results. Next I consider the four steps involved in all types of data analysis: preparing data, running analysis, presenting results, and preserving your work. Within each step there are four major tasks: planning the work, organizing your files and materials, documenting what you have done, and executing the analysis. Since there are alternative approaches to accomplish any given aspect of your work, what makes one workflow better than another? To answer this question, I provide several criteria for evaluating the way you work. These criteria should help you decide which procedures to use, and they motivate many of my recommendations for best practice that are given in this book.

1.1 Replication: The guiding principle for workflow

Being able to reproduce the work you have presented or published should be the cornerstone of any workflow. Science demands replicability and a good workflow facilitates your ability to replicate your results. How you plan your project, document your work, write your programs, and save your results should anticipate the need to replicate. Too often researchers do not worry about replication until their work is challenged. This is not to say that they are taking shortcuts, doing shoddy work, or making decisions that are unjustified. Rather, I am talking about taking the steps necessary so that all the good work that has been done can be easily reproduced at a later time. For example, suppose that a colleague wants to expand upon your work and asks you for the data and commands used to produce results in a published paper. When this happens, you do not want to scramble furiously to replicate your results. Although it might take a few hours to dig out your results (many of mine are in notebooks stacked behind my file cabinets), this should be a matter of retrieving the records, not trying to remember what it was you did or discovering that what you documented does not correspond to what you presented.

Think about replication throughout your workflow. At the completion of each stage of your work, take an hour or a day if necessary to review what you have done, to check that the procedures are documented, and to confirm that the materials are archived. When you have a draft of a paper to circulate, review the documentation, check that you still have the files you used, confirm that the do-files still run, and double check that the numbers in your paper correspond to those in your output. Finally, make sure that all of this is documented in your research log (discussed on page 37).

If you have tried to replicate your own work months after it was completed or tried to reproduce another author's results using only the original dataset and the published paper, you know how difficult it can be to replicate something. A good way to understand what is required to replicate your work is to consider some of the things that can make replication impossible. Many of these issues are discussed in detail later in the book. First, you have to find the original files, which gets more difficult as time passes. Once you have the files, are they in formats that can be analyzed by your current software? If you can read the file, do you know exactly how variables were constructed or cases were selected? Do you know which variables were in each regression model? Even if you have all of this information, it is possible that the software you are currently using does not compute things exactly the same way as the software you used for the original analyses. An effective workflow can make replication easier.

A recent example, illustrates how difficult it can be to replicate even simple analyses. I collected some data that were analyzed by a colleague in a published paper. I wanted to replicate his results to extend the analyses. Due to a drive failure, some of his files were lost. Neither of us could reproduce the exact results from the published paper. We came close, but not close enough. Why? Suppose that 10 decisions were made in the process of constructing the variables and selecting the sample for analysis. Many of these decisions involve choices between options where neither choice is incorrect. For example, do you take the square root of publications or the log after adding .5? With 10 such decisions, there are $2^{10} = 1{,}024$ different outcomes. All of them will lead to similar findings, but not exactly the same findings. If you lose track of decisions made in constructing your data, you will find it very difficult to reproduce what you have done. By the way, remarkably another researcher who was using these data discovered the secret to reproducing the published results.

Even if you have the original data and analysis files, it can be difficult to reproduce results. For published papers, it is often impossible to obtain the original data or the details on how the results were computed. Freese (2007) makes a compelling argument for why disciplines should have policies that govern the availability of information needed to replicate results. I fully support his recommendations.

1.2 Steps in the workflow

Data analysis involves four major steps: cleaning data, performing analysis, presenting findings, and saving your work. Although there is a logical sequence to these steps, the dynamics of an effective workflow are flexible and highly dependent upon the specific project. Ideally, you advance one step at a time, always moving forward until you are done. But, it never works that way for me. In practice, I move up and down the steps depending on how the work goes. Perhaps I find a problem with a variable while analyzing the data, which takes me back to cleaning. Or my results provide unexpected insights, so I revise my plans for analysis. Still, I find it useful to think of these as distinct steps.

1.2.1 Cleaning data

Before substantive analysis begins, you need to verify that your data are accurate and that the variables are well named and properly labeled. That is, you clean the data. First, you must bring your data into Stata. If you received the data in Stata format, this is as simple as a single `use` command. If the data arrived in another format, you need to verify that they were imported correctly into Stata. You should also evaluate the variable names and labels. Awkward names make it more difficult to analyze the data and can lead to mistakes. Likewise, incomplete or poorly designed labels make the output difficult to read and lead to mistakes. Next verify that the sample and variables are what they should be. Do the variables have the correct values? Are missing data coded appropriately? Are the data internally consistent? Is the sample size correct? Do the variables have the distribution that you would expect? Once these questions are resolved, you can select the sample and construct new variables needed for analysis.

1.2.2 Running analysis

Once the data are cleaned, fitting your models and computing the graphs and tables for your paper or book are often the simplest part of the workflow. Indeed, this part of the book is relatively short. Although I do not discuss specific types of analysis, later I talk about ways to ensure the accuracy of your results, to facilitate later replications, and to keep track of your do-files, data files, and log files regardless of the statistical methods you are using.

1.2.3 Presenting results

Once the analyses are complete, you want to present them. I consider several issues in the workflow of presentation. First, you need to move the results from your Stata output into your paper or presentation. An efficient workflow can automate much of this work. Second, you need to document the provenance of all findings that you present. If your presentation does not preserve the source of your results, it can be very difficult to track them down later (e.g., someone is trying to replicate your results or you must respond to a reviewer). Finally, there are a number of simple things that you can do to make your presentations more effective.

1.2.4 Protecting files

When you are cleaning your data, running analyses, and writing, you need to protect your files to prevent loss due to hardware failure, file corruption, or unintentional deletions. Nobody enjoys redoing analyses or rewriting a paper because a file was lost. There are a number of simple things you can do to make it easier to routinely save your work. With backup software readily available and the cost of disk storage so cheap, the hardest parts of making backups is keeping track of what you have. Archiving is distinct from backing up and more difficult since it involves the long-term preservation of files so that they will be accessible years into the future. You need to consider if the file formats and storage media will be accessible in the future. You must also consider the operating system you use (it is now difficult to read data stored using the CP/M operating system), the storage media (can you read 5 1/4” floppy disks from the 1980s or even a ZIP disk from a few years ago?), natural disasters, and hackers.

1.3 Tasks within each step

Within each of the four major steps, there are four primary tasks: planning your work, organizing your materials, documenting what you do, and executing the work. While some tasks are more important within particular steps (e.g., organization while planning), each task is important for all steps of the workflow.

1.3.1 Planning

Most of us spend too little time planning and too much time working. Before you load data into Stata, you should draft a plan of what you want to do and assess your priorities. What types of analyses are needed? How will you handle missing data? What new variables need to be constructed? As your work progresses, periodically reassess your plan by refining your goals and analytic strategy based on the work you have completed. A little planning goes a long way, and I almost always find that planning saves time.

1.3.2 Organization

Careful organization helps you work faster. Organization is driven by the need to find things and to avoid duplication of effort. Good organization can prevent you from searching for lost files or, worse yet, having to reconstruct them. If you have good documentation about what you did, but you cannot find the files used to do the work, little is gained. Organization requires you to think systematically about how you name files and variables, how you organize directories on your hard drive, how you keep track of which computer has what information (if you use more than one computer), and where you store research materials. Problems with organization show up when you have not been working on a project for a while or when you need something quickly. Throughout the book, I make suggestions on how to organize materials and discuss tools that make it easier to find and work with what you have.

1.3.3 Documentation

Without adequate documentation, replication is virtually impossible, mistakes are more likely, and work usually takes longer. Documentation includes a research log that records what you do and codebooks that document the datasets you create and the variables they contain. Complete documentation also requires comments in your do-files and labels and notes within data files. Although I find writing documentation to be an onerous task, certainly the least enjoyable part of data analysis, I have learned that time spent on documentation can literally save weeks of work and frustration later. Although there is no way to avoid time spent writing documentation, I can suggest things that make documenting your work faster and more effective.

1.3.4 Execution

Execution involves carrying out specific tasks within each step. Effective execution requires the right tools for the job. A simple example is the editor used to write your programs. Mastering a good text editor can save you hours when writing your programs and will lead to programs that are better written. Another example is learning the most effective commands in Stata. A few minutes spent learning how to use the `recode` command can save you hours of writing `replace` commands. Much of this book involves selecting the right tool for the job. Throughout my discussion of tools, I emphasize standardizing tasks and automating them. The reason to standardize is that it is generally faster to do something the way you did it before than it is to think up a new way to do it. If you set up templates for common tasks, your work becomes more uniform, which makes it easier to find and avoid errors. Efficient execution requires assessing the trade-off between investing the time in learning a new tool, the accuracy gained by the new tools, and the time you save by being more efficient.

1.4 Criteria for choosing a workflow

As you work on the various tasks in each step of your workflow, you will have choices of different ways to do things. How do you decide which procedure to use? In this section, I consider several criteria for evaluating your current workflow and choosing from among alternative procedures for your work.

1.4.1 Accuracy

Getting the correct answer is the *sine qua non* of a good workflow. Oliveira and Stewart (2006, 30) make the point very well, “If your program is not correct, then nothing else matters.” At each step in your work, you must verify that your results are correct. Are you answering the question you set out to answer? Are your results what you wanted and what you think they are? A good workflow is also about making mistakes. Invariably, mistakes will happen, probably a lot of them. Although an effective workflow can prevent some errors, it should also help you find and correct them quickly.

1.4.2 Efficiency

You want to get your analyses done as quickly as possible, given the need for accuracy and replicability. There is an unavoidable tension between getting your work done and the need to work carefully. If you spend so much time verifying and documenting your work that you never finish the project, you do not have a viable workflow. On the other hand, if you finish by publishing incorrect results, both you and your field suffer. You want a workflow that gets things done as quickly as possible without sacrificing the accuracy of your results. A good workflow, in effect, increases the time you have to do your work, without sacrificing the accuracy of what you do.

1.4.3 Simplicity

A simpler workflow is better than a more complex workflow. The more complicated your procedures, the more likely you will make mistakes or abandon your plan. But what is simple for one person might not be simple for another. Many of the procedures that I recommend involve programming methods that may be new to you. If you have never used a loop, you might find my suggestion of using a loop much more complex than repeating the same commands for multiple variables. With experience, however, you might decide that loops are the simplest way to work.

1.4.4 Standardization

Standardization makes things easier because you do not have to repeatedly decide how to do things and you will be familiar with how things look. When you use standardized formats and procedures, it is easier to see when something is wrong and ensure that you do things consistently the next time. For example, my do-files all use the same structure for organizing the commands. Accordingly, when I look at the log file, it is easier for me to find what I want. Whenever you do something repeatedly, consider creating a template and establishing conventions that become part of your routine workflow.

1.4.5 Automated

Procedures that are automated are better because you are less likely to make mistakes. Entering numbers into your do-file by hand is more error prone than using programming tools to transfer the information automatically. Typing the same list of variables multiple times in a do-file makes it easy to create lists that are supposed to be the same but are not. Again automation can eliminate this problem. Automation is the backbone for many of the methods recommended in this book.

1.4.6 Usability

Your workflow should reflect the way you like to work. If you set up a workflow and then ignore it, you do not have a good workflow. Anything that increases the chances of maintaining your workflow is helpful. Sometimes it is better to use a less efficient approach that is also more enjoyable. For example, I like experimenting with software and prefer taking longer to complete a task while learning a new program than getting things done quicker the old way. On the other hand, I have a colleague who prefers using a familiar tool even if it takes a bit longer to complete the task. Both approaches make for a good workflow since they compliment our individual styles of work.

1.4.7 Scalability

Some ways of work are fine for small jobs but do not work well for larger jobs. Consider the simple problem of alphabetizing ten articles by author. The easiest approach is to lay the papers on a table and pick them up in order. This works well for ten articles but is dreadfully slow with 100 or 1,000 articles. This issue is referred to as *scalability*—how well do procedures work when applied to a larger problem? As you develop your workflow, think about how well the tools and practices you develop can be applied to a larger project. An effective workflow for a small project where you are the only researcher might not be sustainable for a large project involving many people. Although you can visually inspect every case for every variable in a dataset with 25 measures of development in 80 countries, this approach does not work with the National Longitudinal Survey that has thousands of cases and thousands of variables. You should strive for a workflow that adapts easily to different types of projects. Few procedures scale perfectly. As a consequence you are likely to need different workflows for projects of different complexities.

1.5 Changing your workflow

This book has hundreds of suggestions. Decide which suggestions will help you the most and adapt them to the way you work. Suggestions for minor changes to your workflow can be adopted at any time. For example, it only takes a few minutes to learn how to use `notes`, and you can benefit from this command almost immediately. Other suggestions might require major changes to how you work and should only be made when you have the time to fully integrate them into your work. It is a bad idea to make major changes when a deadline is looming. On the other hand, make sure you find time to improve your workflow. Time spent improving your workflow should save time in the long run and improve the quality of your work. An effective workflow is something that evolves over time, reflecting your experience, changing technology, your personality, and the nature of your current research.

1.6 How the book is organized

This book is organized so that it can be read front to back by someone wanting to learn about the entire workflow of data analysis. I also wanted it to be useful as a reference for people who encounter a problem and who want a specific solution. For this purpose, I have tried to make the index and table of contents extensive. It is also useful to understand the overall structure of this book before you proceed with your reading.

Chapter 2 – Planning, organizing, and documenting your work discusses how to plan your work, organize your files, and document what you have done. Avoid the temptation of skipping this chapter so that you can get to the “important” details in later chapters.

Chapter 3 – Writing and debugging do-files discusses how do-files should be used for almost all of your work in Stata. I provide information on how to write more effective do-files and how to debug programs that do not work. Both beginners and advanced users should find useful information here.

Chapter 4 – Automating Stata is an introduction to programming that discusses how to create macros, run loops, and write short programs. This chapter is not intended to teach you how to write sophisticated programs in Stata (although it might be a good introduction), rather it discusses tools that all data analysts should find useful. I encourage every reader to study the material in this chapter before reading chapters 5–7.

Chapter 5 – Names and labels discusses both principles and tools for creating names and labels that are clear and consistent. Even if you have received data that is labeled, you should consider improving the names and labels in the dataset. This chapter is long and includes a lot of technical details that you can skip until you need them.

Chapter 6 – Cleaning data and constructing variables discusses how to check whether your data is correct and how to construct new variables and verify that they were created correctly. At least 80% of the work in data analysis involves getting the data ready, so this chapter is essential.

Chapter 7 – Analyzing, presenting, and replicating results discusses how to keep track of the analyses used in presentations and papers, issues to consider when presenting your results, and ways to make replication simpler.

Chapter 8 – Saving your work discusses how to back up and archive your work. This seemingly simple task is often frustratingly difficult and involves subtle problems that can be easy to overlook.

Chapter 9 – Conclusions draws general conclusions about workflow.

Appendix A reviews how the Stata program operates; considers working with a networked version of Stata, such as that found in many computer labs; explains how to install user-written programs, such as the Workflow package; and shows you how to customize the way in which Stata works.

Additional information about workflow, including examples and discussion of other software, is available at http://www.indiana.edu/~jslsoc/workflow.htm.