Development Research in Practice

Kristoffer Bjarkefur, Luiza Cardoso de Andrade, Benjamin Daniels, Maria Ruth Jones

# DEVELOPMENT RESEARCH IN PRACTICE: THE DIME ANALYTICS DATA HANDBOOK

# Acknowledgments and notes

We want to thank all the people who helped us get here, especially Arianna Legovini, for her leadership at DIME, unending support of our work, and detailed comments on this book; and Florence Kondylis, for her leadership in founding and growing DIME Analytics and supporting this project from the very first. We also thank the following members of DIME Analytics for their contributions to the ideas in this book and their help organizing them: Roshni Khincha, Avnish Singh, Patricia Paskov, Radhika Kaul, Mizuhiro Suzuki, Yifan Powers, and Maria Arnal Canudo. This work has been financially supported by the United Kingdom Foreign, Commonwealth & Development Office (FCDO) through the DIME i2i Umbrella Facility for Impact Evaluation at the World Bank.

Our graditude to the many people who read and offered feedback as the book took shape: Stephanie Annijas, Maria Camila Ayala Guerrero, Kaustubh Chahande, Thomas Escande, Aram Gassama, Steven Glover, Nausheen Khan, Robert Norling, Michael Orevba, Caio Piza, Francesco Raffaelli, Daniel Rogger, Ankriti Singh, Ravi Somani, and Leonardo Viotti. Although they number far too many to name individually, we also thank all the members of DIME and its teams across all the years for the innovative work they have done, the lessons learned, and the team spirit that makes our work so fruitful and rewarding.

This published version of the book has been revised repeatedly since its internal release in June 2019, with extensive feedback from readers and experts. We would additionally like to thank Vincenzo di Maro (Manager, DIME3) for his support throughout this process, as well as peer reviewers David McKenzie (Lead Economist, DECFP), Holly Krambeck (Program Manager, DECAT), Alaka Holla (Program Manager, HEDGE), Jim Shen (Senior Manager, Abdul Latif Jameel Poverty Action Lab), Federica di Battista (Trialling Lead, UK FCDO), and Gabriel Vicente, Rajee Kanagavel, and Maksim Pecherskiy from the Development Data Partnership team.

This book is a living product that is written and maintained publicly. The code and edit history are at: . You can get a PDF copy at: . The website includes updated instructions for providing feedback, as well as notes on updates to the content. Whether you work with DIME, the World Bank, or another organization or university, we ask that you read the contents of this book critically. We welcome feedback and corrections to improve the book. Please visit to provide feedback. You can also email us at [dimeanalytics@worldbank.org](mailto:dimeanalytics@worldbank.org), and we will be very thankful. We hope you enjoy *Development Research in Practice*!

# Abbreviations

**2SLS** – Two-Stage Least Squares **AEA** – American Economic Association **CAPI** – Computer-Assisted Personal Interviewing **DD or DiD** – Differences-in-Differences **DIME** – Development Impact Evaluation **DOI** – Digital object identifier **eGAP** – Evidence in Governance and Politics **EU** – European Union **GDPR** – Global Data Protection Regulation **HFC** – High-Frequency Checks **IPA** – Innovations for Poverty Action **IRB** – Institutional Review Board **IV** – Instrumental Variables **J-PAL** – The Abdul Latif Jameel Poverty Action Lab **MD** – Minimum Detectable Effect **NGO** – Non-Governmental Organization **ODK** – Open Data Kit **OLS** – Ordinary Least Squares **OSF** – Open Science Foundation **PI** – Principal Investigator **PII** – Personally-Identifying Information **RA** – Research Assistant **RD** – Regression Discontinuity **RCT** – Randomized Control Trial **SSC** – Statistical Software Components

# Preface

Welcome to *Development Research in Practice: The DIME Analytics Data Handbook*. This book is intended to teach all users of development data how to handle data effectively, efficiently, and ethically. An empirical revolution has changed the face of development research rapidly over the last decade. Increasingly, researchers are working not just with complex data, but with *original* data: datasets collected by the research team themselves or acquired through a unique agreement with a project partner. Research teams must carefully document how original data is created, handled, and analyzed. These tasks now contribute as much weight to the quality of the evidence as the research design and the statistical approaches do. At the same time, the scope and scale of empirical research projects is expanding: more people are working on the same data over longer timeframes. For that reason, the central premise of this book is that data work is a “social process”. This means that the many different people on a team need to have the same ideas about what is to be done, and when and where and by whom, so that they can collaborate effectively on a large, long-term research project.

Despite the growing importance of managing data work, little practical guidance is available for practitioners. There are few guides to the conventions, standards, and best practices that are fast becoming a necessity for empirical research. *Development Research in Practice* aims to fill that gap. It covers the full data workflow for a complex research project using original data. We share the lessons, tools, and processes developed within the World Bank’s Development Impact Evaluation (DIME) department, and compile them into a single narrative of best practices for data work. This book is not sector-specific; it will not teach you econometrics, or how to design an impact evaluation. There are many excellent existing resources on those topics. Instead, it will teach you how to think about all aspects of your research from a data perspective, how to structure research projects to ensure data quality, and how to institute transparent and reproducible workflows. We realize that adopting these workflows may have significant upfront learning costs, but we are convinced that these investments pay off quickly, as you will both save time and improve the quality of your research going forward.

## How to read this book

This book aims to be a highly practical resource so the reader can immediately begin to collaborate more effectively on large, long-term research projects that use the methods and tools discussed. This introduction outlines the basic philosophies that motivate this book and our approach to research data. We want all readers to understand at the outset our mindset that research data work is primarily about communicating effectively within a team and that standardization and simplification of data tasks is a major enabler of effective collaboration. The main chapters of this book will walk you through the data workflow at each stage of an empirical research project, from design to publication. The figure in this introduction visualizes the data workflow. Chapters 1 and 2 contextualize the workflow, and set the stage for the hands-on data tasks which are described in detail in Chapters 3 to 7.

**Chapter 1** This book aims to be a highly practical resource so the reader can immediately begin to collaborate more effectively on large, long-term research projects that use the methods and tools discussed. This introduction outlines the basic philosophies that motivate this book and our approach to research data. We want all readers to understand at the outset our mindset that research data work is primarily about communicating effectively within a team and that standardization and simplification of data tasks is a major enabler of effective collaboration. The main chapters of this book will walk you through the data workflow at each stage of an empirical research project, from design to publication. The figure in this introduction visualizes the data workflow. Chapters 1 and 2 contextualize the workflow, and set the stage for the hands-on data tasks which are described in detail in Chapters 3 to 7.

**Chapter 2** teaches you to structure your data work for collaborative research, while ensuring the privacy and security of research participants. It discusses the importance of planning data work and associated tools in advance, long before any data is acquired. It also describes ethical concerns common to development data, common pitfalls in legal and practical management of data, and how to respect the rights of research participants at all stages of data work

**Chapter 3** turns to the measurement framework, and how to translate research design to a data work plan. It details DIME’s data map template, a set of tools to communicate the project’s data requirements both across the team and across time. It also discusses how to implement random sampling and random assignment in a reproducible and credible manner.

**Chapter 4** covers data acquisition. It starts with the legal and institutional frameworks for data ownership and licensing, to ensure that you are aware of the rights and responsibilities of using data collected by the research team or by others. It provides a deep dive on collecting high-quality primary electronic survey data, including developing and deploying survey instruments. Finally, it discusses secure data handling during transfer, sharing, and storage, which is essential in protecting the privacy of respondents in any data.

**Chapter 5** describes data processing tasks. It details how to construct “tidy” data at the appropriate units of analysis, how to ensure uniquely identified datasets, and how to routinely incorporate data quality checks into the workflow. It also provides guidance on de-identification and cleaning of personally-identified data, focusing on how to understand and structure data so that it is ready for indicator construction and analytical work.

**Chapter 6** discusses data analysis tasks. It begins with data construction, or the creation of new variables from the raw data acquired or collected in the field. It introduces core principles for writing analytical code and creating, exporting, and storing research outputs such as figures and tables reproducibly using dynamic documents.

**Chapter 7** outlines the publication of research outputs, including manuscripts, code, and data. This chapter discusses how to effectively collaborate on technical writing using dynamic documents. It also covers how and why to publish datasets in an accessible, citable, and safe fashion. Finally, it provides guidelines for preparing functional and informative reproducibility packages that contain all the code, data, and meta-information needed for others to evaluate and reproduce your work.

Each chapter starts with a box which provides a summary of the most important points, takeaways for different types of readers, and a list of key tools and resources for implementing the recommended practices. After reading each chapter, you should understand what tasks will be performed at every stage of the workflow, and how to implement them according to best practices. You should also understand how the various stages of the workflow tie together, and what inputs and outputs are required and produced from each. The references and links contained in each chapter will lead you to detailed descriptions of individual ideas, tools, and processes to refer to when you need to implement the tasks yourself.

### Demand for Safe Spaces Case Study

Throughout this Handbook, we will refer to a completed DIME project, *Demand for Safe Spaces: Avoiding Harassment and Stigma*, to provide a case study of the empirical research tasks described in this book. You will find boxes in each chapter with examples of how the practices and workflows described in that chapter were applied in this real-life example. All the code examples and diagrams referenced in the case study can be accessed directly through this book’s GitHub repository. We have made minor adaptations to the original study materials presented for function and clarity. All original materials can be found in the project’s reproducibility package.

**The *Demand for Safe Spaces* study is summarized in its abstract as follows.** What are the costs to women of harassment on public transit? This study randomizes the price of a women-reserved “safe space” in Rio de Janeiro and crowdsource information on 22,000 rides. Women in the public space experience harassment once a week. A fifth of riders are willing to forgo 20 percent of the fare to ride in the “safe space”. Randomly assigning riders to the “safe space” reduces physical harassment by 50 percent, implying a cost of $1.45 per incident. Implicit Association Tests show that women face a stigma for riding in the public space that may outweigh the benefits of the safe space.

The Demand for Safe Spaces study used novel original data from three sources. It collected information on 22,000 metro rides from a crowdsourcing app (referred to as *crowdsourced ride data* in the case study examples), a survey of randomly-sampled commuters on the platform (referred to as the *platform survey*), and data from an *implicit association test*. The research team first elicited revealed preferences for the women-reserved cars, and then randomly assigned riders across the reserved and non-reserved cars to measure differences in the incidence of harassment. The use of a customized app allowed the researchers to assign data collection tasks and vary assigned ride spaces (women-reserved car vs public cars) and associated payout across rides. In addition, the team administered social norm surveys and implicit associations tests on a random sample of men and women commuters to document a potential side eﬀect of reserved spaces: stigma against women who choose to ride in the public space.

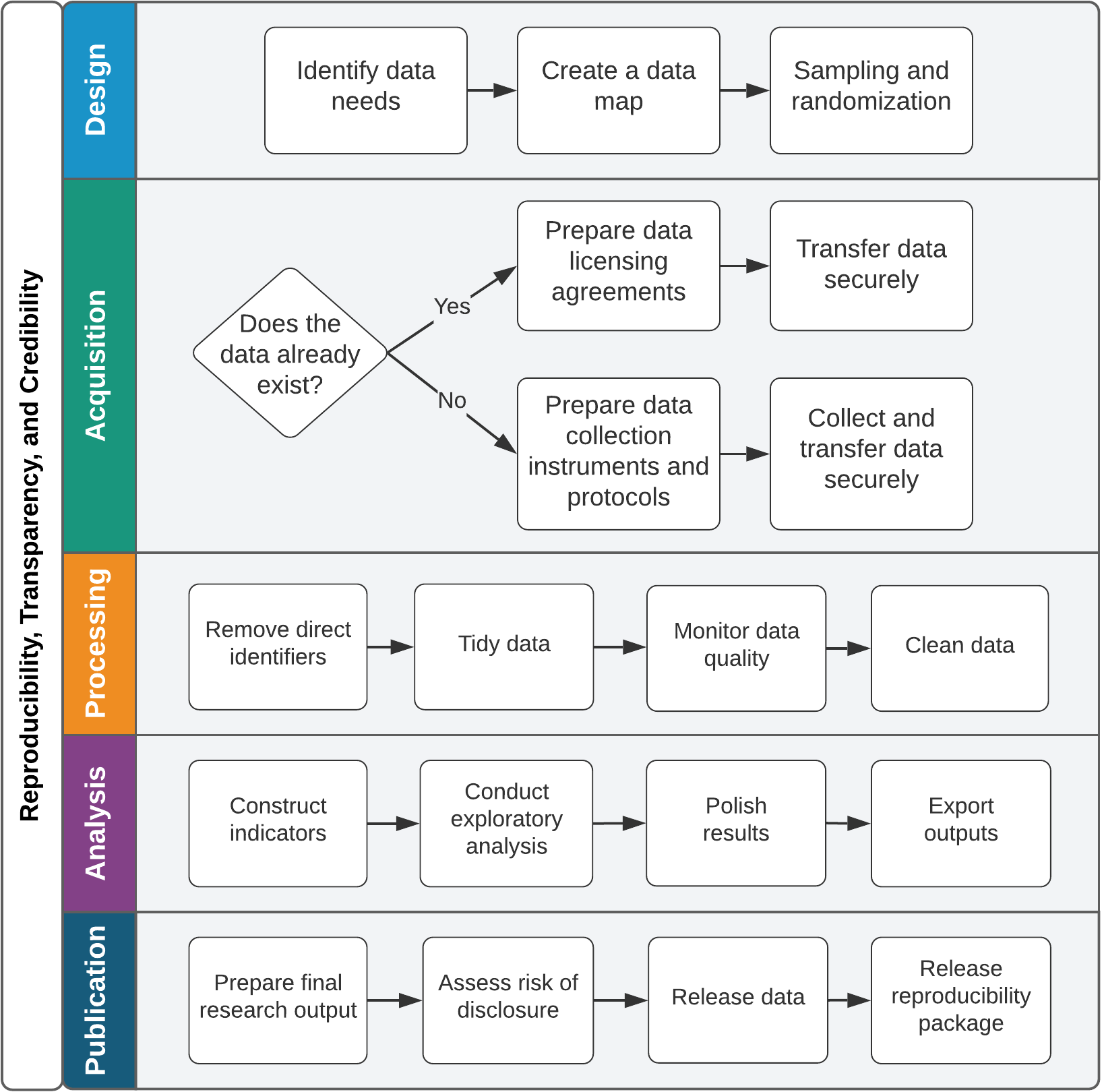
For the purposes of the Handbook, we focus on the protocols, methods, and data used in the *Demand for Safe Spaces study*, rather than the results. To learn more about the findings from this study, and more details on how it was conducted, we encourage readers to download the working paper. The material from all the examples in the book and be found at <https://github.com/worldbank/dime-data-handbook/tree/master/examples>.

The Demand for Safe Spaces study repository can be accessed at <https://github.com/worldbank/rio-safe-space>.

The working paper *Demand for Safe Spaces: Avoiding Harassment and Stigma* is available at <https://openknowledge.worldbank.org/handle/10986/33853>.

## The DIME Wiki: A complementary resource

Throughout the book, you will find many references to the DIME Wiki. The DIME Wiki is a free online collection of impact evaluation resources and best practices. This handbook and the DIME Wiki are meant to go hand-in-hand: the handbook provides the narrative structure and workflow, and the Wiki dives into specific implementation details, offers detailed code examples, and provides a more exhaustive set of references for each topic. Importantly, the DIME Wiki is a living resource that is continuously updated and improved, by the authors of this book and external contributors. We welcome all readers to [register as Wiki users](https://dimewiki.worldbank.org/New_Users) and contribute directly.



Overview of development research data work tasks

## Standardizing data work

In the past, data work was often treated as a “black box” in research. A published manuscript might exhaustively detail research designs, estimation strategies, and theoretical frameworks, but typically reserved very little space for detailed descriptions of how data was actually collected and handled. It is almost impossible to assess the quality of the data in such a paper, and whether the results could be reproduced. In the past decade, this has started to change,[[1]](#footnote-37) in part due to increasing requirements by publishers and funders to release code and data.

Data handling and documentation is a key skill for researchers and research staff. Standard processes and documentation practices are important throughout the research process to accurately convey and implement the intended research design,[[2]](#footnote-38) and to minimizes security risks: better protocols and processes lower the probability of data leakages, security breaches, and loss of personal information. When data work is done in an ad-hoc manner, it is very difficult for others to understand what is being done – a reader has to simply trust that the researchers did these things right. Most importantly, if any part of the data pipeline breaks down, research results become unreliable[[3]](#footnote-39) and cannot be faithfully interpreted as being an accurate picture of the intended research design.[[4]](#footnote-40) Because we almost never have “laboratory” settings[[5]](#footnote-42) in this type of research, such a failure has a very high cost: we will have wasted the investments that were made into knowledge generation, and the research opportunity itself, where we intended to conduct the study.[[6]](#footnote-43)

Accurate and reproducible data management and analysis is essential to the success and credibility of modern research. Standardizing and documenting data handling processes is essential to be able to evaluate and understand the data work alongside any final research outputs. An important component of this is **process standardization**.[[7]](#footnote-44) Process standardization means that there is little ambiguity about how something ought to be done, and therefore the tools to do it can be set in advance. Standard processes help other people understand your work, and they also make your work easier to document. Process standardization and documentation should allow readers of your code to: (1) quickly understand what a particular process or output is supposed to be doing; (2) evaluate whether or not it does that thing correctly; and (3) modify it either to test alternative hypotheses or to adapt into their own work. This book will discuss specific standards recommended by DIME Analytics, but we are more interested in convincing the reader to discuss the adoption of *a* standard within research teams than to necessarily use *the* particular standards that we recommend.

## Standardizing coding practices

Modern quantitative research relies heavily on standardized statistical software tools, written with various coding languages, to standardize analytical work. Outputs like regression tables and data visualizations are created using code in statistical software for two primary reasons. The first is that using a standard command or package ensures that the work is done right, and the second is that it ensures the same procedure can be confirmed or checked at a later date or using different data. Keeping a clear, human-readable record of these code and data structures is critical. While it is often *possible* to perform nearly all the relevant tasks through an interactive user interface or even through software such as Excel, this practice is strongly advised against. In the context of statistical analysis, the practice of writing all work using standard code is widely accepted. To support this practice, DIME now maintains portfolio-wide standards about how analytical code should be maintained and made accessible before, during, and after release or publication.

Over the last few years, DIME has extended the same principles to preparing data for analysis, which often comprises just as much (or more) of the manipulation done to the data over the life cycle of a research project. A major aim of this book is to encourage research teams to think of the tools and processes they use for designing, collecting, and handling data just as they do for analytical tasks. Correspondingly, a major contribution of DIME Analytics has been tools and standard practices for implementing these tasks using statistical software.

While we assume that you are going to do nearly all data work using code, many development researchers come from economics and statistics backgrounds and often understand code to be a means to an end rather than an output itself. We believe that this must change somewhat: in particular, we think that development practitioners must think about their code and programming workflows just as methodologically as they think about their research workflows, and think of code and data as research outputs, just as manuscripts and briefs are.

This approach arises because we see the code as the “recipe” for the analysis. The code tells others exactly what was done, how they can do it again in the future, and provides a roadmap and knowledge base for further original work.[[8]](#footnote-46) Performing every task through written code creates a record of every task you performed.[[9]](#footnote-47) It also prevents direct interaction with the data files that could lead to non-reproducible processes.[[10]](#footnote-48) Finally, DIME Analytics has invested a lot of time in developing code as a learning tool: the examples we have written and the commands we provide are designed to provide a framework for common practice across the entire DIME team, so that everyone is able to read, review, and provide feedback on the work of others starting from the same basic ideas about how various tasks are done.

Most specific code tools have a learning and adaptation process, meaning you will become most comfortable with each tool only by using it in real-world work. To support your process of learning reproducible tools and workflows, we reference free and open-source tools wherever possible, and point to more detailed instructions when relevant. **Stata**,[[11]](#footnote-49) as a proprietary software, is the notable exception here due to its persistent popularity in development economics and econometrics. This book also includes, in an appendix, the **DIME Analytics Coding Guide** which includes instructions for how to write good code, instructions on how to use the code examples in this book, as well as our Stata Style Guide. DIME projects are strongly encouraged to explicitly adopt and follow coding style guides in their work. Style guides harmonize code within and across teams making it easier to understand and reuse code, which ultimately helps teams to build on each other’s best practices. Some of the programming languages used at DIME already have well-established and commonly used style guides, such as the Tidyverse style guide for R and PEP-8 for Python.[[12]](#footnote-50). Stata has relatively few resources of this type available, which is why we have created and included one here that we hope will be an asset to all Stata users.

## The team behind this book

DIME is the Development Impact Evaluation department of the World Bank.[[13]](#footnote-53) Its mission is to generate high-quality and operationally relevant data and research to transform development policy, help reduce extreme poverty, and secure shared prosperity.[[14]](#footnote-55) DIME develops customized data and evidence ecosystems to produce actionable information and recommend specific policy pathways to maximize impact. The department conduct research in 60 countries with 200 agencies, leveraging a US$180 million research budget to shape the design and implementation of US$18 billion in development finance. DIME also provides advisory services to 30 multilateral and bilateral development agencies.[[15]](#footnote-56) DIME research is organized into four primary topic pillars: Economic Transformation and Growth; Gender, Economic Opportunity, and Fragility; Governance and Institution Building; and Infrastructure and Climate Change. Over the years, DIME has employed dozens of research economists, and hundreds of full-time research assistants, field coordinators, and other staff. The team has conducted over 325 impact evaluations. *Development Research in Practice* exists to take advantage of that concentration and scale of research, to synthesize many resources for data collection and research, and to make DIME tools available to the larger community of development researchers.

As part of its broader mission, DIME invests in public goods to improve the quality and reproducibility of development research around the world. One key early innovation at DIME was the creation of [DIME Analytics](https://www.worldbank.org/en/research/dime/data-and-analytics), the team responsible for writing and maintaining this book. DIME Analytics is a centralized unit that develops and ensures adoption of high quality research practices across the department’s portfolio. This is done through an intensive, collaborative innovation cycle: DIME Analytics onboards and supports research assistants and field coordinators, provides standard tools and workflows to all teams, delivers hands-on support when new tasks or challenges arise, and then develops and integrates lessons from those engagements to bring to the full team. Resources developed and tested in DIME are converted into public goods for the global research community, through open-access trainings and open-source tools. The DIME Analytics Resource Directory appendix provides an introduction to public materials.

DIME Analytics has invested many hours over the past years learning from data work across DIME’s portfolio, identifying inefficiencies and barriers to success, developing tools and trainings, and standardizing best-practice workflows adopted in DIME projects. It has also invested significant energy in the language and materials used to teach these workflows to new team members, and, in many cases, in software tools that support these workflows explicitly. DIME team members often work on diverse portfolios of projects with a wide range of teammates, and we have found that standardizing core processes across all projects results in higher-quality work with fewer opportunities to make mistakes. In that way, the Analytics team is DIME’s method of institutionalizing tools and practices, developed and refined over time, that give the department a common base of knowledge and practice. In 2018, for example, DIME adopted universal reproducibility checks conducted by the Analytics team; the lessons from this practice helped move the DIME team from where 50% of submitted papers in 2018 required significant revision to pass to where 64% of papers passed in 2019 without revision required.

## Looking ahead

While adopting the workflows and mindsets described in this book requires an up-front cost, it will save you (and your collaborators) a lot of time and hassle very quickly. In part this is because you will learn how to implement essential practices directly; in part because you will find tools for the more advanced practices; and most importantly because you will acquire the mindset of doing research with a high-quality data focus.

For some readers, the amount of new tools and practices recommended in this book may seem daunting. We know from experience at DIME that full-scale adoption is possible; in the last few years, the full DIME portfolio has transitioned to transparent and reproducible workflows, with a fair share of hiccups along the way. The authors of this book supported that at-scale transition, and we hope that by sharing our lessons learned and resources, the learning curve for readers will be less steep. In the summary boxes at the beginning of each chapter, we provide a list of the key tools and resources to help readers prioritize. We will also offer “second-best” practices in many cases, suggesting easy-to-implement suggestions to increase transparency and reproducibility, in cases where full-scale adoption of the recommended workflows is not immediately feasible. In fact, we encourage teams to adopt one new practice at a time rather than rebuild their whole workflow from scratch right away. We hope that by the end of the book, all readers will have learned how to handle data more efficiently, effectively and ethically at all stages of the research process.

# Conducting reproducible, transparent, and credible research

Policy decisions are made every day using the results of development research, and these have wide-reaching effects on the lives of millions. As the emphasis on evidence-based policy grows, so too does the scrutiny under which research methods and results are placed. Three major components make up this scrutiny: credibility, transparency, and reproducibility. These three components contribute to one simple idea: research should be high quality and well-documented. Research consumers, including the policy makers who will use the evidence to make decisions, should be able to easily examine and recreate such evidence. In this framework, it is useful to think of research as a public service that requires researchers as a group to be accountable for their methods. This means acting to collectively protect the credibility of development research by following modern practices for research planning and documentation. Across the social sciences, the open science movement has been fueled by concerns about the proliferation of low-quality research practices, data and code that are inaccessible to the public, analytical errors in major research papers, and in some cases even outright fraud. While the development research community has not yet experienced major scandals, it has become clear that there are necessary improvements to make in the way that code and data are handled as part of research. Moreover, having common standards and practices for creating and sharing materials, code, and data with others will improve the value of the work we do.

In this chapter, we outline principles and practices that help to ensure research consumers can be confident in the conclusions reached. We discuss each of the three components – credibility, transparency, and reproducibility – in turn. The first section covers research credibility. It presents three popular methods to commit to particular research questions or methods, and avoid potential criticisms of cherry-picking results: registration, pre-analysis plans, and registered reports. The second section discusses how to apply principles of transparency to all research processes, which allows research teams to be more efficient, and research consumers to fully understand and evaluate research quality. The final section provides guidance on how to make your research fully reproducible, and explains why publishing replication materials is an important research contribution in its own right.

### Summary: Conducting reproducible, transparent, and credible research

This chapter describes three pillars of a high-quality empirical research project: credibility, transparency and reproducibility. These steps and outputs discussed in this chapter should be prepared at the beginning of a project and revisited through the publication process.

**1. Credibility:** to enhance credibility, you should pre-commit research decisions as much as feasible

* Register research studies to provide a record of every project, so all evidence about a topic can be maintained; pre-register studies to protect design choices from later criticism.
* Write pre-analysis plans for data collection and analysis to both strengthen the conclusions drawn from those analyses and increase efficiency by creating a road map for project data work.
* Publish a registered report to combines the benefits of the two steps above with a formal peer review process and a conditional acceptance of the final results of the specified research.

**2. Transparency:** you should document all data acquisition and analysis decisions during the project lifecycle, with a clear understanding of what will be released publicly and plan for how those will be published

* Develop and publish comprehensive project documentation, especially instruments for data collection or acquisition that may be needed to prove ownership rights and facilitate re-use of the data.
* Retain all original data in an unaltered form and archive it appropriately, in preparation for it to be de-identified and published at the appropriate time.
* Write all data processing and analysis code with public release in mind.

**3. Reproducibility:** Prepare analytical work that can be verified and reproduced by others. This means

* Understanding what archives and repositories are appropriate for your various materials
* Preparing for legal documentation and licensing of data, code, and research products
* Initiating reproducible workflows that will easily transfer within and outside of your research team and the necessary documentation for others to understand and use your materials

#### Takeaways

**TTLs/PIs will:**

* Develop and document the research design and the corresponding data required to execute it
* Guide the research team in structuring and completing project registration
* Understand the team’s future rights and responsibilities regarding data, code, and research publication
* Determine what methods of pre-commitment are appropriate and lead the team in preparing them

**RAs will:**

* Rework outputs and documentation to meet specific technical requirements of registries, funders, publishers, or other governing bodies
* Inform the team leadership whenever methodologies, data strategies, or their planned executions are not sufficiently clear or are not appropriately documented or communicated
* Familiarize themselves with best practices for carrying out reproducible and transparent research, and initiate those practices within the research team

#### Key Resources

* Register your research study: <https://dimewiki.worldbank.org/Study_Registration>
* Create a pre-analysis plan: <https://dimewiki.worldbank.org/Pre-Analysis_Plan>
* Prepare to document research decisions: <https://dimewiki.worldbank.org/Data_Documentation>
* Publish data in a trusted repository: <https://dimewiki.worldbank.org/Publishing_Data>
* Prepare and publish a reproducibility package: <https://dimewiki.worldbank.org/Reproducible_Research>

## Developing a credible research project

The evidentiary value of research is traditionally a function of design choices,[[16]](#footnote-69) such as powered through sampling and randomization, and robustness to alternative specifications and definitions. One frequent target for critics of such research[[17]](#footnote-70) is the fact that most researchers have a lot of leeway in selecting their projects, results, or outcomes *after* already having had the experience of implementing a project or collecting data in the field, which increases the likelihood of finding “false positive” results that are not true outside carefully-selected data.[[18]](#footnote-71) Credible research design methods are key to maintaining credibility in these choices and avoiding serious errors. This is especially relevant for research that relies on original data sources, from innovative big data sources to unique surveys. Development researchers should take these concerns seriously. Such flexibility can be a significant issue for the quality of evidence overall, particularly if researchers believe that certain types of results are substantially better for their careers or their publication chances.

This section presents three popular methods for researchers to commit to particular research questions or methods, and to avoid potential criticisms of cherry-picking results for publication: registration, pre-analysis plans, and registered reports. Each of these methods involves documenting specific research design components, ideally before carrying out the analytical component or extensively exploring the data. Study registration provides formal notice that a study is being attempted and creates a hub for materials and updates about the study results. Pre-analysis plans are a more formal commitment to use specific methods on particular questions. Writing and releasing a pre-analysis plan in advance of working with data is used to protect the credibility of approaches that have a high likelihood of producing false results.[[19]](#footnote-72) Finally, registered reports allow researchers to approach research planning itself as a process at the level of a full peer review. Registered reports enable close scrutiny of a research design, a feedback and improvement process, and a commitment from a publisher to publish the study based on the credibility of the design, rather than the specific results.

### Registering research studies

Registration of research studies is an increasingly common practice, and more journals are beginning to require the registration of studies they publish.[[20]](#footnote-74) Study registration intended to ensure that a complete record of research inquiry is easily available.[[21]](#footnote-75) Registering research studies ensures that future scholars can quickly find out what work has been carried out on a given question, even if some or all of the work done never results in formal publication. Registration of studies is increasingly required by publishers and can be done before, during, or after the study with essential information about the study purpose. Some currently popular registries are operated by the [**AEA**](https://www.socialscienceregistry.org), [**3ie**](https://ridie.3ieimpact.org), [**eGAP**](https://egap.org/content/registration), and [**OSF**](https://osf.io/registries). They all have different target audiences and features, so select one that is appropriate to your work. Study registration should be feasible for all projects, as registries are typically free to access and can be initially submitted with minimal information about the project. A generally-acceptable practice will be to gradually revise and expand the level of detail in a registration over time, adding more information as the planning for the project progresses.

Pre-registering studies before they begin is an extension of this principle.[[22]](#footnote-80) Registration of a study before it goes to implementation or data acquisition, particularly when specific hypotheses are included in the registration, provides a simple and low-effort way for researchers to conclusively demonstrate that a particular line of inquiry was not generated by the process of data collection or analysis itself.[[23]](#footnote-81) Pre-registrations need not provide exhaustive details about how a particular hypothesis will be approached; only that it will be. Pre-registering specific individual elements of research design or analysis can be highly valuable for the credibility of the research and requires only minor time investment or administrative effort. For this reason, the DIME team requires pre-registration of all studies in a public database with at least some primary hypotheses prespecified, prior to providing funding for impact evaluation research.

### Demand for Safe Spaces Case Study: Registering Research Studies

The experimental component of the *Demand for Safe Spaces* study was registered at the Registry for International Development Impact Evaluations (RIDIE) under ID 5a125fecae423.

Highlights from the Registry:

* *Indicated evaluation method:* both primary method (randomized control trial) and additional methods (difference-in-difference/fixed effects).
* *Listed key outcome variables:* take-up of rides in women-only car (binary), occurrence of harassment or crime during ride (binary), self-reported well-being after each ride, overall subjective well-being, Implicit Association Test D-Score
* *Specified primary hypotheses to be tested:* The women-only car reduces harassment experienced by women who ride it; Riding the women’s-only car improves psychological well-being of those who ride it; Women are willing to forego income to ride the women’s-only car
* *Specified secondary research question and methods:* supplementary research methods (implicit association test and platform survey) to help address an additional hypothesis: The women’s-only car is associated with a social norm that assigns responsibility to women for avoiding harassment.
* *Provided sample size for each study arm:* number of individual participants, number of baseline rides, number of rides during price experiment, number of car-assigned rides, number of expected participants in implicit association test
* \*Described data sources: the study relied on data previously collected (through the mobile app) and data to-be-collected (through platform surveys and implicit association tests)
* *Registration status:* categorized as a non-prospective registry, as the crowdsourced data had already been received and processed. It was important to the team to ensure the credibility of additional data collection and secondary research questions by registering the study.

The RIDIE registry can be accessed at <https://ridie.3ieimpact.org/index.php?r=search/detailView&id=588>

### Writing pre-analysis plans

If a research team has a large amount of flexibility to define how they approach a particular hypothesis, study registration may not be sufficient to avoid the criticism of “hypothesizing after the results are known”, or HARKing.[[24]](#footnote-86) Examples of such flexibility include a broad range of concrete measures that could be argued to measure to an abstract concept; future choices about sample inclusion or exclusion; or decisions about how to construct derived indicators. There are a variety of templates and checklists of details to include.[[25]](#footnote-87) When the researcher is collecting a large amount of information and has leverage over even a moderate number of these options, it is almost guaranteed that they can come up with any result they like.[[26]](#footnote-89)

Pre-analysis plans (PAPs) can be used to assuage these concerns by specifying some set of analyses the researchers intend to conduct.[[27]](#footnote-90) The pre-analysis plan should be written up in detail for areas that are known to provide a large amount of leeway for researchers to make later decisions, particularly for things like interaction effects or subgroup analysis.[[28]](#footnote-91) Pre-analysis plans shoud not, however, be viewed as binding the researcher’s hands.[[29]](#footnote-92) Depending on what is known about the study at the time of writing, pre-analysis plans can vary widely in the amount of detail they should include.[[30]](#footnote-93) The core function of a PAP is to carefully and explicitly describe one or more specific data-driven inquiries, as specific formulations are often very hard to justify in retrospect with data or projects that potentially provide many avenues to approach a single theoretical question.[[31]](#footnote-95) Anything outside the original plan is just as interesting and valuable as it would have been if the plan was never published; but having pre-committed to the details of a particular inquiry makes its results immune to a wide range of criticisms of specification searching or multiple testing.[[32]](#footnote-96)

### Demand for Safe Spaces Case Study: Writing Pre-Analysis Plans

Although the *Demand for Safe Spaces* study did not publish a formal pre-analysis plan, the team published a concept note in 2015, which includes much of the same information as a typical pre-analysis plan. The Concept Note was updated in May 2017 to include new secondary research questions. The Concept Note, prepared before fieldwork began, was subject to review and approval within the World Bank and from a technical committee including blinded feedback from external academics. The Concept Note specified the planned study along the following dimensions:

* *Theory of change:* the main elements of the intervention, and the hypothesized causal chain from inputs, through activities and outputs, to outcomes.
* *Hypotheses* derived from the theory of change
* *Main evaluation question(s)* to be addressed by the study
* *List of main outcomes of interest,* including outcome name, definition, level of measurement
* *Evaluation design,* including a precise description of the identification strategy for each research questions and description of treatment and control groups
* *Sampling strategy and sample size calculation,* detailing the assumptions made
* *Description of all quantitative data collection instruments*
* *Data processing and analysis:* the statistical methods to be used, the exact specification(s) to be run, including clustering of standard errors; key groups for heterogeneity analysis; adjustments for multiple hypothesis testing; strategy to test (and correct) for bias.

A version of the study’s Concept Note is available at <https://github.com/worldbank/rio-safe-space/blob/master/Online%20Appendices/Supplemental%20Material/Project%20Concept%20Note.pdf>

### Publishing registered reports

**Registered reports** take the process of pre-specifying a complex research design to the level of a formal publication.[[33]](#footnote-100) In a registered report, a journal or other publisher will peer review and conditionally accept a specific study for publication, typically then guaranteeing the acceptance of a later publication that carries out the analysis described in the registered report. While far stricter and more complex to carry out than ordinary study registration or pre-analysis planning, the registered report has the added benefit of peer review and expert feedback on the design and structure of the proposed study.[[34]](#footnote-102) Registered reports are never required, but they are designed to reward researchers who are able to provide a large amount of advance detail for their projects, researchers who want to secure publication interest regardless of results, or researchers who want to use methods that may be novel or unusual.

This process is in part meant to combat the “file-drawer problem”[[35]](#footnote-104) and ensure that researchers are transparent in the sense that all promised results obtained from registered-report studies are actually published. This approach has the advantage of pre-specifying in great detail a complete research and analytical design, and securing a commitment for publication regardless of the outcome. This may be of special interest for researchers studying events or programs where either there is a substantial risk that they would either not be able to publish a null or negative result,[[36]](#footnote-105) or where they may wish to avoid any pressure toward finding a particular result, for example when the program or event is the subject of substantial social or political pressures. As with pre-registration and pre-analysis, nothing in a registered report should be understood to prevent a researcher from pursuing additional avenues of inquiry once the study is complete, either in the same or separate research outputs.

## Conducting research transparently

Transparent research exposes not only the code, but all research processes involved in developing the analytical approach. This means that readers are able to judge for themselves whether the research was done well and the decision-making process was sound. If the research is well-structured, and all of the relevant documentation[[37]](#footnote-107) is shared, it is easy for the reader to understand the analysis fully. Researchers that expect process transparency also have an incentive to make better decisions, be skeptical and thorough about their assumptions, and save themselves time, because transparent research methods are labor-saving over the complete course of a project.

Clearly documenting research work is necessary to allow others to evaluate exactly what data was acquired and how it was used to obtain a particular result. Many development research projects are purpose-built to address specific questions, and often use unique data, novel methods, or small samples. These approaches can yield new insights into essential academic questions, but need to be transparently documented so they can be reviewed or replicated by others in the future.[[38]](#footnote-109) Unlike disciplines where data is more standardized or where research is more oriented around secondary data, the exact data used in a development project has often not been observed by anyone else in the past and may not be able to be re-collected by others in the future. Regardless of the novelty of study data, transparent documentation methods help ensure that data was collected and handled appropriately and that studies and interventions were implemented correctly. As with study registrations, project and data documentation should be released on external **archival repositories**[[39]](#footnote-110) so they can always be accessed and verified.

### Documenting data acquisition and analysis

Documenting a project in detail greatly increases transparency. Many disciplines have a tradition of keeping a “lab notebook”, and adapting and expanding this process to create a lab-style workflow in the development field is a critical step towards more transparent practices. This means explicitly noting decisions as they are made, and explaining the process behind the decision-making. Careful documentation will also save the research team a lot of time during a project, as it prevents you from having the same discussion twice (or more!), since you have a record of why something was done in a particular way. There are a number of available tools that will contribute to producing documentation, but project documentation should always be an active and ongoing process, not a one-time requirement or retrospective task. New decisions are always being made as the plan begins contact with reality, and there is nothing wrong with sensible adaptation so long as it is recorded and disclosed.

Email, however, is *not* a documentation service, because communications are rarely well-ordered, can be easily deleted, and are not available for future team members. At the very least, emails and other decision-making communications need to be archived and preserved (as, say, PDFs) in an organized manner so that they can be easily accessed and read by others in the future. There are also various software solutions for building proper documentation over time. Some work better for field records such as implementation decisions, research design, and survey development; others work better for recording data work and code development. The **Open Science Framework**[[40]](#footnote-112) provides one such solution, with integrated file storage, version histories, and collaborative wiki pages. **GitHub**[[41]](#footnote-114) provides a transparent documentation system through commit messages, issues, README files, and pull requests,[[42]](#footnote-116) in addition to version histories and wiki pages. Such services offer multiple different ways to record the decision process leading to changes and additions, track and register discussions, and manage tasks. These are flexible tools that can be adapted to different team and project dynamics. Services that log your research process can show things like modifications made in response to referee comments, by having tagged version histories at each major revision. They also allow you to use issue trackers to document the research paths and questions you may have tried to answer as a resource to others who have similar questions. Each project has specific requirements for data, code, and documentation management, and the exact transparency tools to use will depend on the team’s needs, but they should be agreed upon prior to project launch. This way, you can start building a project’s documentation as soon as you start making decisions.

### Cataloging and archiving data

Data and data collection methods should be fully cataloged, archived, and documented, whether you are collecting data yourself or receiving it from an outside partner. In some cases this is as simple as uploading a survey instrument or an index of datasets and a codebook to an archive. In other cases this will be more complex. Proper documentation of data collection will often require a detailed description of the overall sampling procedure.[[43]](#footnote-119) For example, settings with many overlapping strata, treatment arms, excluded observations, or resampling protocols might require extensive additional field work documentation. This documentation should be continuously updated and kept with the other study materials; it is often necessary to collate these materials for an appendix for publication in any case.

When data is received from partners or collected in the field, the **original data** (including corrections)[[44]](#footnote-120) should be immediately placed in a secure permanent storage system. Before analytical work begins, you should create a “for-publication” copy of the original dataset by removing potentially identifying information. This will become the raw data, and must be placed in an archival repository where it can be cited.[[45]](#footnote-121) This can initially be done under embargo or with limited release, in order to protect your data and future work. This type of data depositing or archiving precedes publishing or releasing any data: data at this stage may still need to be embargoed or have other, potentially permanent, access restrictions, so you can instruct the archive to formally release the data later. If your planned analysis requires the use of unpublishable data, that data should always remain encrypted and be stored separately so it is clear what portions of the code will work with and without obtaining a license to the needed restricted-access data.

Some project funders provide specific repositories in which they require the deposit of data they funded,[[46]](#footnote-122) and you should take advantage of these when possible. If this is not provided, you must be aware of privacy issues with directly identifying data and questions of data ownership before uploading raw data to any third-party server, whether public or not; this is a legal question for your home organization. If data that is required for analysis must be placed under restricted use or restricted access, including data that can never be distributed directly by you to third parties, develop a plan for storing that data separately from publishable information. This will allow you to maximize transparency by having a clear release package as well as providing instructions or developing a protocol for allowing access in the future for replicators or reviewers under appropriate access agreements.[[47]](#footnote-124) Regardless of these consideration, all data repositories, such as DIME’s standard, the World Bank Microdata Library[[48]](#footnote-126) and the World Bank Data Catalog,[[49]](#footnote-128) should create a record of the data’s existence and provide instructions on how access might be obtained by another researcher. For more on the steps required to prepare and publish a de-identified dataset, you can refer to Chapter 6 and Chapter 7 of this book. Data publication should create a data citation and a **digital object identifier (DOI)**,[[50]](#footnote-130) or some other persistent index that you can use in your future work to unambiguously indicate the location of your data. This data publication should also include the methodological documentation as well as complete human-readable codebooks for all the variables there.

## Analyzing data reproducibly

Reproducible research makes it easy for others to apply your techniques to new data or to implement a similar research design in a different context. Development research is rapidly moving in the direction of requiring adherence to specific reproducibility guidelines.[[51]](#footnote-132) Major publishers and funders, most notably the American Economic Association, have taken steps to require that code and data are accurately reported, cited, and preserved as research outputs that can be accessed and verified by others. Making research reproducible in this way is a public good.[[52]](#footnote-133) It enables other researchers to re-use code and processes to do their own work more easily and effectively in the future. Regardless of what is formally required, your code should be written neatly with clear instructions. It should be easy to read and understand. The corresponding analysis data should also be made accessible to the greatest legal and ethical extent that it can be.[[53]](#footnote-134)

Common research standards from journals and funders feature both regulation and verification policies.[[54]](#footnote-135) Regulation policies require that authors provide reproducibility packages before publication which are then reviewed by the journal for completeness.[[55]](#footnote-136) Verification policies require that authors make certain materials available to the public, but their completeness is not a precondition for publication. Other journals have adopted guidance that offer checklists for reporting on whether and how various practices were implemented, without specifically requiring any.[[56]](#footnote-137) If you are personally or professionally motivated by citations, producing these kinds of resources can lead to that as well. Even if privacy considerations mean you will not be publishing some or all data or results, these practices are still valuable for project organization.

Our recommendation, regardless of external requirements, is that your should prepare to release all data that can be published When data cannot be published, you should try to publish as much metadata as allowed, including information on how the data was obtained, what fields the data contains and aggregations or descriptive statistics. Even if the data cannot be published, it is rare for code files to contain restricted information, so the code should still be made available with clear instructions for obtaining usable data. Additionally, we recommend that reproducibility efforts be considered when designing the IRB and data licensing agreement for sensitive data, to establish acceptable conditions (such as a secure transfer or cold room) under which representatives from journals or other academics could access data may access the data for the purpose of independently reproducing results.

### Preparing a reproducibility package

At DIME, all research outputs are required to satisfy **computational reproducibility**,[[57]](#footnote-139) which is an increasingly common requirement for publication.[[58]](#footnote-140) Before releasing a working paper, the research team submits a **reproducibility package** with de-identified data, and DIME Analytics verifies that the package produces exactly the same results that appear in the paper.[[59]](#footnote-141) The team also comments on whether the package includes sufficient documentation. The Analytics team organizes frequent peer code review for works in progress, and our general recommendation is to ensure that projects are *always* externally reproducible instead of waiting until the final stages to prepare this material. Once the computational reproducibility check is complete, the team receives a completed reproducibility certificate that also lists any publicly available materials to accompany the package, for use as an appendix to the publication. The team also organizes regular peer code review for works in progress, and our general recommendation is to ensure that projects are *always* externally reproducible instead of waiting until the final stages to prepare this material. In this way, code is continuously maintained with clear documentation, and should be easy to read and understand in terms of structure, style, and syntax.

For research to be reproducible, all code files for data cleaning, construction and analysis should be public, unless they contain confidential information. Nobody should have to guess what exactly comprises a given index, or what controls are included in your main regression, or whether or not you clustered standard errors correctly. That is, as a purely technical matter, nobody should have to “just trust you”, nor should they have to bother you to find out what would happen if any or all of these things were to be done slightly differently.[[60]](#footnote-143) Letting people play around with your data and code is a great way to have new questions asked and answered based on the valuable work you have already done.[[61]](#footnote-144)

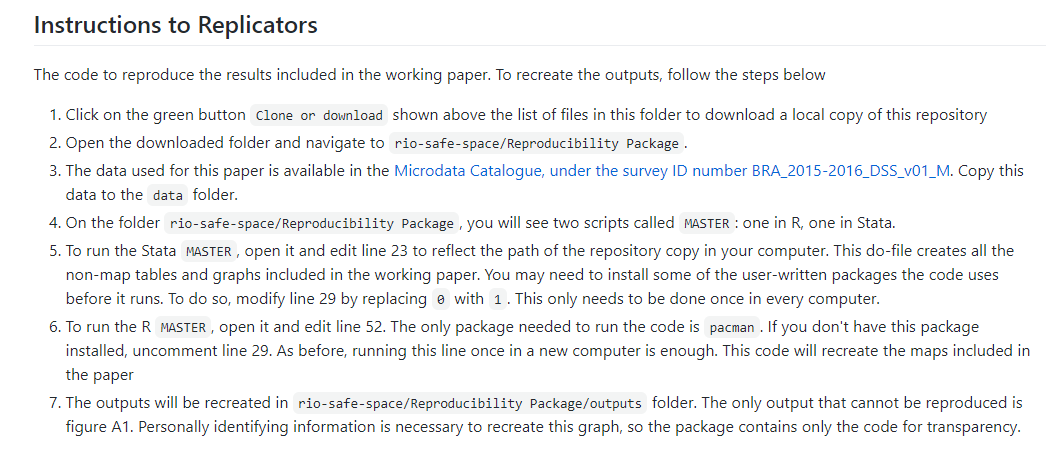
A reproducibility package should include the complete materials needed to exactly re-create your final analysis, and be accessible and well-documented so that others can identify and adjust potential decision points that they are interested in. They should be able to easily identify: what data was used and how that data can be accessed; what code generates each table, figure and in-text number; how key outcomes are constructed; and how all project results can be reproduced. This is important to plan ahead for, so that you can make sure you obtain the proper documentations and permissions for all data, code, and materials you use throughout the project. A well-organized reproducibility package usually takes the form of a complete directory structure, including documentation and a master script, that leads the reader through the process and rationale for the code behind each of the outputs when considered in combination with the corresponding publication.

### Demand for Safe Spaces Case Study: Preparing a Reproducibility Package

The *Demand for Safe Spaces* team published all final study materials to a repository on the World Bank’s GitHub account. The repository holds the abstract of the paper, ungated access to the most recent version of the full paper, an online appendix including robustness checks and supplemental material, and the project’s reproducibility package.

The data for this project is published in the Microdata Catalog, under the survey ID number BRA\_2015-2016\_DSS\_v01\_M. The Microdata catalog entry includes metadata on the study, documentation such as survey instruments and technical reports, terms of use for the data, and access to downloadable data files. Both the crowdsourced data and the platform survey data are accessible through the Microdata Catalog.

The “Reproducibility Package” folder on GitHub contains all the instructions for executing the code. Among other things, it provides licensing information for the materials, software and hardware requirements including time needed to run, and instructions for replicators (which are included below). Finally, it has a detailed list of the code files that will run, their data inputs, and the outputs of each process.



The Demand for Safe Space GitHub repository can be viewed at : <https://github.com/worldbank/rio-safe-space>

The Microdata Catalog entry for the study is available at <https://microdata.worldbank.org/index.php/catalog/3745>

## Looking ahead

With the ongoing rise of empirical research and increased public scrutiny of scientific evidence, making analysis code and data available is necessary but not sufficient to guarantee that findings will be credible. Even if your methods are highly precise, your evidence is only as good as your data – and there are plenty of mistakes that can be made between establishing a design and generating final results that would compromise its conclusions. That is why transparency is key for research credibility. It allows other researchers, and research consumers, to verify the steps to a conclusion by themselves, and decide whether their standards for accepting a finding as evidence are met. Every investment you make in documentation and transparency up front protects your project down the line, particularly as these standards continue to tighten. With these principles in mind, the approach we take to the development, structure, and documentation of data work provides a system to implementing these ideas in everyday work. In the next chapter, we will discuss the workspace you need in order to work reproducibly in an efficient, organized, and secure manner.

# Setting the stage for effective and efficient collaboration

In order to do effective data work in a team environment, you need to structure your workflow in advance. Preparation for collaborative data work begins long before you acquire any data, and involves planning both software tools and collaboration platforms for your team. This means knowing what types of data you’ll acquire, whether the data will require special handling due to size or privacy considerations, which datasets and outputs you will need at the end of the process, and how all data files and versions will stay organized throughout. It’s important to plan data workflows in advance because changing software or protocols halfway through a project is costly and time-consuming. Seemingly small decisions such as file-sharing services, folder structures, and filenames can be extremely painful to alter down the line in any project.

This chapter will guide you in setting up an effective environment for collaborative data work, structuring your data work to be well-organized and clearly documented, and setting up processes to handle confidential data securely. The first section outlines how to set up your working environment to effectively collaborate on technical tasks with others, and how to document tasks and decisions. The second section discusses how to organize your code and data so that others will be able to understand and interact with it easily. The third section provides guidelines for ensuring privacy and security when working with confidential data.

### Summary: Setting the stage for effective and efficient collaboration Summary

The technical environment for your data work needs to be established at the start of a research project. Agreeing with the team on software choices, standard code and data structure, and clear data security protocols will prepare you to successfully, safely, and efficiently implement technical tasks throughout the project lifecycle. Consider:

1. **The technical collaboration environment.** No matter the hardware and software your team plans to use, you should ensure now that it is standardized or interoperable across the team. This includes:

* Secure all **physical computing hardware** through encryption and password-protection. If specialized or more powerful hardware is required, initiate access requests, purchase orders, or other processes now
* Agree on tools for **collaboration** and documentation, such that key conversations and decisions are archived and organized outside of instant-message and email conversation
* Decide the **programming languages and environments** the team will use. Take time to set up a comfortable and modern digital work environment

1. The **organization of code and data**. The team should agree on where and how code files and databases will be stored, down to the level of the folder structure. This involves setting up:

* A standardized and scalable **folder structure** so all documents have an unambiguous location, and the location and naming of files describes their purpose and function and is intuitive to all team members
* A **backup and version control system** appropriate for each file type, to ensure information cannot be lost and that all team members understand how to interoperate and collaborate
* **Master script files** that will structure and execute the code base of the

1. **Information security measures and ethical frameworks**. These include:

* Formally **request and obtain approval** from legal entities governing research in all relevant jurisdictions
* Understand **how to respect the rights and dignity of research subjects** and plan for how to establish **informed consent** from individuals or groups participating in the research
* Adopt standardized **digital security practices** including proper encryption of all confidential information, at rest and in transit, both among your team and with external partners

#### Takeaways

**TTLs/PIs will:**

* Support the acquisition and maintenance of required computing hardware and software, liaising with procurement, information security and information technology teams as necessary
* Make final decisions regarding code languages and environments
* Discuss and agree upon an appropriate project-wide digital organization strategy
* Institute and communicate best practices in accordance with legal, ethical, and security obligations

**RAs will:**

* Communicate technical needs clearly with TTLs/PIs and relevant service providers
* Consistently implement digital organization strategy and flag issues with task management, documentation, or materials storage if they arise
* Support project compliance with ethical, legal, and security obligations and flag concerns to TTLs/PIs

#### Key Resources

* DIME Research Ethics Standards: Pillar 1 of the DIME Research Standards <https://github.com/worldbank/dime-standards>
* DIME GitHub Resources: <https://github.com/worldbank/dime-github-trainings>
* DIME Data Security Standards: Pillar 4 of the DIME Research Standards <https://github.com/worldbank/dime-standards>
* DIME Data Publication Standards: Pillar 5 of the DIME Research Standards <https://github.com/worldbank/dime-standards>

## Preparing a collaborative work environment

This section introduces core concepts and tools for organizing data work in an efficient, collaborative and reproducible manner. Some of these skills may seem elementary, but thinking about simple things from a workflow perspective can help you make marginal improvements every day you work; those add up to substantial gains over the course of multiple years and projects. Together, these processes form a collaborative workflow that will greatly accelerate your team’s ability to get tasks done on all your projects.

Teams often develop workflows in an ad hoc fashion, solving new challenges as they arise. Adaptation is good, of course. But it is important to recognize that there are a number of tasks that exist in common for every project, and it is more efficient to agree on the corresponding workflows in advance. For example, every project requires research documentation, organized file naming, directory organization, coding standards, version control, and code review. These tasks are common to almost every project, and their solutions translate well between projects. Therefore, there are large efficiency gains to thinking in advance about the best way to do these tasks, instead of throwing together a solution when the task arises. This section outlines the main points to discuss within the team, and suggests best practice solutions for these tasks.

### Demand for Safe Spaces Case Study: Preparing a Collaborative Work Environment

Here are a few highlights of how the **Demand for Safe Spaces** team chose to organize their work environment for effective collaboration:

* The data work for the project was done through a private GitHub repository housed in the World Bank organization account.
* GitHub issues were used to document research decisions and to provide feedback. Even the PIs for the study, who did not directly participate in coding, used Github issues to review code and outputs and to create a record of broader discussions.
* Stata was adopted as the primary software for data analysis, as that is the software all team members had in common at the start of the project. At a later stage of the project, R code was developed specifically to create maps. The R portion of the code was developed independently, as it used different datasets and created separate outputs. The team used two separate master scripts, one for the Stata code base and one for the R code.
* The team members shared a synchronized folder (using Dropbox), which included the de-identified data and project documentation such as survey instruments and enumerator training manuals.

### Setting up your computer

First things first: almost all your data work will be done on your computer, so make sure it’s set up for success. The operating system should be fully updated, it should be in good working order, and you should have a **password-protected** login. However, password-protection is not sufficient if your computer stores data that is not public. You would need to use encryption for sufficient protection, which will be covered later in this chapter. Make sure your computer is backed up to prevent information loss. Follow the **3-2-1 rule**: maintain 3 copies of all original or irreplaceable data, on at least 2 different hardware devices you have access to, with 1 offsite storage method.[[62]](#footnote-158) Chapter 4 provides a protocol for implementing this.

Ensure you know how to get the **absolute file path** for any given file. On MacOS this will be something like “/users/username/git/project/...” and “C:/users/username/git/project/...” on Windows. Absolute file paths will be an obstacle to collaboration unless they are **dynamic absolute file paths**. In a dynamic absolute file path the relative project path, “/git/project/...” in the examples above, is added to the user-specific root path for each user, “/users/username” or “C:/users/username” in the examples above, generating an absolute file path unique to each user. Master scripts introduced later in this chapter will show how this can be seamlessly implemented. Dynamic absolute file paths, starting from the file system root, is the best way to ensure that files are read and written correctly when multiple users work on the same project across many different platforms, operative systems and devices. There are contexts, for example some cloud environments, where relative file paths must be used, but in all other contexts we recommend you to always use dynamic absolute file paths.

Use forward slashes (/) in file paths for folders, and whenever possible use only the 26 English characters, numbers, dashes (-), and underscores (\_) in folder names and filenames.[[63]](#footnote-160) For emphasis: *always* use forward slashes (/) in file paths in code, just like in internet addresses. Do this even if you are using a Windows machine where both forward and backward slashes are allowed, as your code will otherwise break if anyone tries to run it on a Mac or Linux machine. Making the structure of your directories a core part of your workflow is very important, since otherwise you will not be able to reliably transfer the instructions for replicating or carrying out your analytical work.

When you are working with others, you will most likely be using some kind of **file sharing** software. The exact services you use will depend on your tasks, but in general, there are several approaches to file sharing, and the three discussed here are the most common. **File syncing** is the most familiar method, and is implemented by software like OneDrive, Dropbox, or Box. Sync forces everyone to have the same version of every file at the same time, which makes simultaneous editing difficult but other tasks easier. **Distributed version control** is another method, commonly implemented through systems like GitHub, GitLab, and Bitbucket that interact with Git.[[64]](#footnote-162) Distributed version control allows everyone to access different versions of files at the same time. It is only optimized for specific types of files (for example, any type of code files). Finally, **server storage** is the least-common method, because there is only one version of the materials, and simultaneous access must be carefully regulated. Server storage ensures that everyone has access to exactly the same files and environment, and it also enables high-powered computing processes for large and complex data.

All three file sharing methods are used for collaborative workflows, and you should review the types of data work that you will be doing, and plan which types of files will live in which types of sharing services. It is important to note that they are, in general, not interoperable, meaning you should not have version-controlled files inside a syncing service, or vice versa, without setting up complex workarounds, and you cannot shift files between them without losing historical information. Therefore, choosing the correct sharing service for each of your team’s needs at the outset is essential. At DIME we typically use file syncing for all project administrative files and data, version control in Git for code, and server storage for backup and/or large scale computations when needed.

### Establishing effective documentation practices

Once your technical and sharing workspace is set up, you need to decide how you are going to communicate with your team. The first habit that many teams need to break is using instant communication for management and documentation. Email is, simply put, not a system. It is not a system for anything. Neither is instant messaging apps like WhatsApp. Instant messaging tools are developed for communicating “now” and that is what they do well. They are not structured to manage group membership or to present the same information across a group of people, or to remind you when old information becomes relevant. They are not structured to allow people to collaborate over a long time or to review old discussions. It is therefore easy to miss or lose communications from the past when they have relevance in the present. Everything with future relevance that is communicated over email or any other instant medium – such as, for example, decisions about research design – should immediately be recorded in a system that is designed to keep permanent records. We call these systems collaboration tools, and there are several that are very useful.

Good collaboration tools are workflow-oriented systems that allow the team to create and assign tasks, carry out discussions related to single tasks, track task progress across time, and quickly see the overall project status. They are web-based so that everyone on your team can access them simultaneously and have ongoing discussions about tasks and processes. Such systems link communications to specific tasks so that related decisions are permanently recorded and easy to find in the future when questions about that task come up. Choosing the right tool for your team’s needs is essential to designing an effective workflow. What is important is that your team chooses a system and commits to using it, so that decisions, discussions, and tasks are easily reviewable long after they are completed.

Some popular and free collaboration tools that meet these criteria are GitHub and Dropbox Paper. Any specific list of software will quickly be outdated; we mention these as examples that have worked for our team. Different collaboration tools can be used different types of tasks. Our team, for example, uses GitHub for code-related tasks, and Dropbox Paper for more managerial tasks. GitHub creates incentives for writing down why changes were made in response to specific discussions as they are completed, creating naturally documented code. It is useful also because tasks in GitHub Issues can clearly be tied to file versions. On the other hand, Dropbox Paper provides a clean interface with task notifications, assignments, and deadlines, and is very intuitive for people with non-technical backgrounds. Therefore, it is a useful tool for managing non-code-related tasks.

### Setting up your code environment

Taking time early in your project to choose a programming language to work in and setting up a productive code environment for that language will make your work significantly easier. Setting up a productive code environment means to make sure that the programming language and all other software your code requires will run smoothly on all the hardware you need it to run on. It also means that you have a productive way of interacting with code, and that the code has a seamless method to access your data.

It is difficult and costly to switch programming languages halfway through a project, so think ahead about the various software your team will use. Take into account the technical abilities of team members, what type of internet access the software will need, the type of data you will need to access, and the level of security required. Big datasets require additional infrastructure and may overburden the tools commonly used for small datasets. Also consider the cost of licenses, the time to learn new tools, and the stability of the tools. There are few strictly right or wrong choices for software, but what is important is that you plan in advance and understand how the chosen tools will interact with your workflows.

One option is to hold your code environment constant over the lifecycle of a single project. While this means you will inevitably have different projects with different code environments, each successive project will be better than the last, and you will avoid the costly process of migrating an ongoing project into a new code environment. Code environments should be documented as precisely as possible. The specific version number of the programming languages and the individual packages you use should be referenced or maintained so that they can be reproduced going forward, even if different releases contain changes that would break your code or change your results. DIME Analytics developed the command ieboilstart in the ietoolkit package to support version and settings stability in Stata.[[65]](#footnote-165) If your project requires more than one programming language, for example if you analyze your data in one language but visualize your results in another, then make sure to make an as clear division between the two as possible. This means that you first complete all tasks done in one language, before the completing the rest of the tasks in the other language. Frequently swapping back and forth between languages is a reproducibility nightmare.

Next, think about how and where you write and execute code. This book is intended to be agnostic to the size or origin of your data, but we are going to broadly assume that you are using one of the two most popular statistical software packages: R or Stata. (If you are using another language, like Python, many of the same principles apply but the specifics will be different.) Most of your code work will be done in a code editor. If you are working in R, **RStudio** is the typical choice.[[66]](#footnote-167) For Stata, the built-in do-file editor is the most widely adopted code editor. You might also consider using an external editor for your R or Stata code.[[67]](#footnote-169) These editors offer great accessibility and quality features. For example, they can access an entire directory – rather than a single file – which gives you directory-level views and file management actions, such as folder management, Git integration, and simultaneous work with other types of files, without leaving the editor. Using an external editor can also be preferable since your editor will not crash if the execution of your code causes your statistical software to crash. Finally, you can often use the same editor for all programming languages you use, so any customization you do in your code editor of choice will improve your productivity across all your coding work.

## Organizing code and data for replicable research

We assume you are going to do your analytical work through code, and that you want all your processes to be documented and replicable. Though it is possible to interact with some statistical software through the user interface without writing any code, we strongly advise against it. Writing code creates a record of every task you performed. It also prevents direct interaction with the data files that could lead to non-reproducible steps. You may do some exploratory tasks by point-and-click or typing directly into the console, but anything that is included in a research output must be coded in an organized fashion so that you can release the exact code that produces your final results – up to and including individual statistics in text. Still, organizing code and data into files and folders is not a trivial task. What is intuitive to one person rarely comes naturally to another, and searching for files and folders is everybody’s least favorite task. As often as not, you come up with the wrong one, and then it becomes very easy to create problems that require complex resolutions later. This section provides basic tips on managing the folder that stores your project’s data work.

Maintaining an organized file structure for data work is the best way to ensure that you, your teammates, and others are able to easily edit and replicate your work in the future. It also ensures that automated processes from code and scripting tools are able to interact well with your work, whether they are yours or those of others. File organization makes data work easier as well as more transparent, and facilitates integration with tools like version control systems that aim to cut down on the amount of repeated tasks you have to perform. It is worth thinking in advance about how to store, name, and organize the different types of files you will be working with, so that there is no confusion down the line and everyone has the same expectations.

### Organizing files and folders

Once you start a research project, the number of scripts, datasets, and outputs that you have to manage will grow very quickly. This can get out of hand just as quickly, so it’s important to organize your data work and follow best practices from the beginning. You should agree with your team on a specific directory structure, and set it up at the beginning of the project. You should also agree on a file naming convention. This will help you to easily find project files and ensure that all team members can easily run the same code.

To support consistent folder organization at DIME, DIME Analytics created iefolder as a part of our ietoolkit package.[[68]](#footnote-173) This Stata command sets up a pre-standardized folder structure for what we call the DataWork folder.[[69]](#footnote-175) The DataWork folder includes folders for all the steps of a typical project. Since each project will always have unique needs, we have tried to make the structure easy to adapt. Having a universally standardized folder structure across the entire portfolio of projects means that everyone can easily move between projects without having to reorient on file and folder organization.

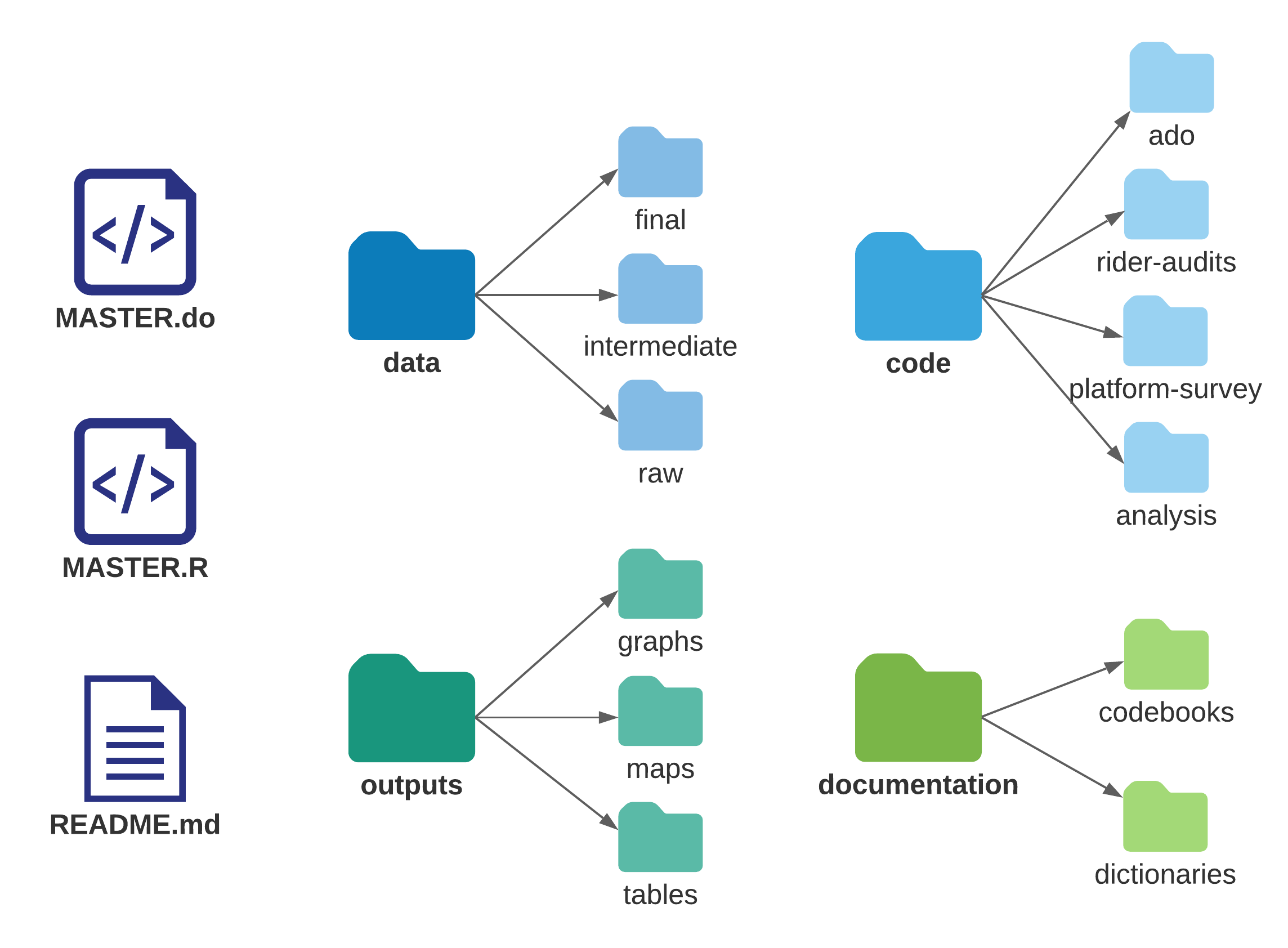
If you do not already have a standard file structure across projects, iefolder is an easy template to start from. The command creates a DataWork folder at the project level, and within that folder, creates standardized directory structures for each data source or survey round. Within each subdirectory, iefolder creates folders for raw encrypted data, raw deidentified data, cleaned data, final data, outputs, and documentation. It creates a parallel folder structure for the code files that move the data through this progression, and for the final analytical outputs. The ietoolkit package also includes the iegitaddmd command, which can place README.md placeholder files in your folders so that your folder structure can be shared using Git.[[70]](#footnote-177) Since these placeholder files are written in a plaintext language called **Markdown**, they also provide an easy way to document the contents of every folder in the structure.

The DataWork folder may be created either inside an existing project folder, or it may be created as a separate root folder. We advise keeping project management materials (such as contracts, terms of reference, briefs and other administrative or management work) separate from the DataWork folder structure. It is useful to maintain project management materials in a sync system like Dropbox, whereas the code folder should be maintained in a version-control system like Git. However, a version-controlled folder should *never* be stored in a synced folder that is shared with other people. Combining these two types of collaboration tools almost always creates undesired functionalities.

### Demand for Safe Spaces Case Study: Organizing Files and Folders

The diagram below illustrates the folder structure for the *Demand for Safe Spaces* data work. Note that this project started before the iefolder structured was created, and so differs from the current guidelines. Changing folder organization midway through a project is never recommended. The key feature of folder organization is planning ahead and agreeing with the whole team on the best structure to follow.

The root folder for the study’s data work included two master scripts (one for Stata and one R, that can be run independently), a readme, and 4 subfolders: data, code, documentation and outputs. The folder structure was identical in GitHub and DropBox, but the files present in each of them differed. All code, raw outputs (such as TeX table fragments and PNG images) and plain text documentation were stored in GitHub. All the datasets, PDF outputs and documentation in Word and Excel were stored in DropBox. When data processing was completed, binary files in the Documentation folder that were accessed by the code, such as iecodebook and ieduplicates spreadsheets, were moved to GitHub to ensure completeness of the repository.



### Establishing common file formats

Each task in the research workflow has specific inputs and outputs which feed into one another. It is common, particularly when different tasks are performed by different people inside a team, for incompatibilities to be created. For example, if the Principal Investigators are writing a paper using {}, exporting tables from statistical software into a .csv format will break the workflow. Therefore, it’s important to agree with your team on what tools will be used for what tasks, and where inputs and outputs will be stored, before you start creating them. Take into account ease of use for different team members, and keep in mind that learning how to use a new tool may require some time investment upfront that will be paid off as your project advances.

Knowing how code outputs will be used will help you decide the best format to export them. You can typically use the same software to save figures into various formats, such as .eps},.png,.pdfor.jpg`. However, the decision between using Office Suite software such as Word and PowerPoint versus {} and other plain text formats may influence how you write your code, as this choice often implicates in the use of a particular format. This decision will also affect the version control systems that your team can use.

### Using version control

We recommend using a **version control system** to maintain control of file history and functionality. A good version control system tracks who edited each file and when, allows you to revert to previous versions, and provides a protocol for ensuring that conflicting versions are avoided. This is important, for example, for your team to be able to find the version of a presentation that you delivered to a donor, or to understand why the significance level of your estimates has changed. Everyone who has ever encountered a file named something like final\_report\_v5\_LJK\_KLE\_jun15.docx can appreciate how useful such a system can be.

Most syncing services offer some kind of rudimentary version control; these are usually enough to manage changes to binary files (such as office documents) without needing to rely on dreaded filename-based versioning conventions. For code files, however, a more detailed version control system is usually desirable. We recommend using Git for version-control of all data work. Git documents changes to all **plaintext** files. Plaintext files include all code files, most raw outputs, and written outputs that use code languages, such as {} files and many dynamic documents. Git tracks all the changes you make to each plaintext file, and allows you to go back to previous versions without losing the information on changes made. It also makes it possible to work on multiple parallel versions of a file, so you don’t risk breaking code for other team members as you try something new.

### Writing code that others can read

Good code is written in a way that is easily understood and run by others. Below we discuss a few crucial steps to code organization. They all come from the principle that code is an output by itself, not just a means to an end, and should be written thinking of how easy it will be for someone to read it later. At the end of this section, we include a template for a master script do-file in Stata, to provide a concrete example of the required elements and structure. Throughout this section, we refer to lines of this example do-file to give concrete examples of the required code elements, organization and structure.

To be readable, code must be well-documented. Start by adding a code header to every file. A code header is a long **comment**[[71]](#footnote-184) that details the functionality of the entire script; refer to lines 5-10 in the example do-file. This should include simple things such as the purpose of the script and the name of the person who wrote it. If you are using a version control software, the last time a modification was made and the person who made it will be recorded by that software. Otherwise, you should include it in the header. You should always track the inputs and outputs of the script, as well as the uniquely identifying variable; refer to lines 52-54 in the example do-file. When you are trying to track down which code creates which dataset, this will be very helpful. While there are other ways to document decisions related to creating code, the information that is relevant to understand the code should always be written in the code file.

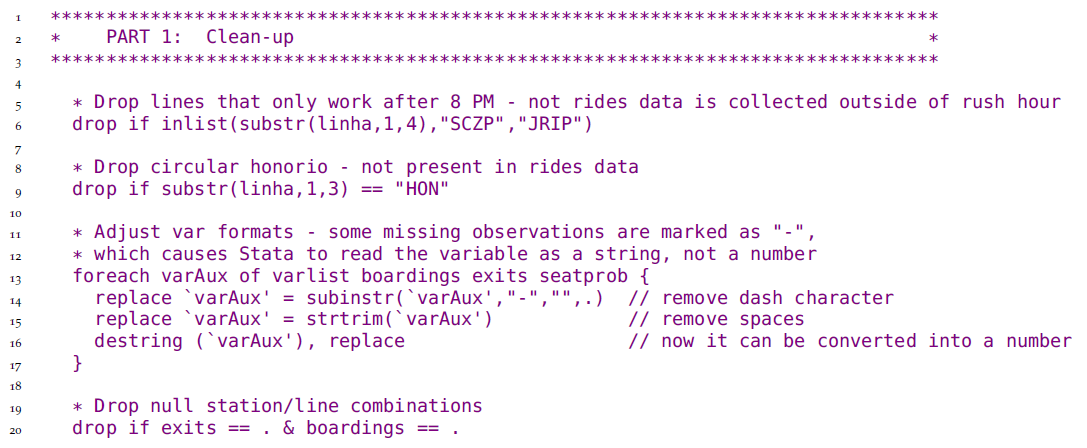
Two types of comments should be included in the script itself. The first type of comment describes *what* is being done. This might be easy to understand from the code itself if you know the language well enough and the code is clear, but often it is still a great deal of work to reverse-engineer the code’s intent. Describing the task in plain English (or whichever language you use to communicate with your team) will make it easier for everyone to read and understand the code’s purpose. It can also help you organize your own work and ensure you are following logical steps. The second type of comment explains *why* the code is performing a task in a particular way. As you are writing code, you are making a series of decisions that (hopefully) make perfect sense to you at the time. These are often highly specialized and may exploit a functionality that is not obvious or has not been seen by others before. Well-commented code is in itself a great way to document your data work which someone can follow to understand anything from data cleaning decisions that make the published data differ from the original data to decisions on how indicators are constructed. Even you will probably not remember the exact choices that were made in a couple of weeks. Therefore, you must document your precise processes in your code.

Code files should be stored in an easy-to-find location and named in a meaningful way. Breaking your code into independently readable “chunks” is good practice for code organization. You should write each functional element as a chunk that can run completely on its own. This ensures that each code component is independent; it does not rely on a complex program state created by other code chunks that are not obvious from the immediate context. One way to do this is by creating sections in your script to identify where a specific task is completed. For example, if you want to find the line in your code where the directory is set, you can go straight to “PART 2: Prepare folder paths and define programs”, instead of reading line by line through the entire code.

RStudio makes it very easy to create sections, and it compiles them into an interactive script index for you. In Stata, you can use comments to create section headers (see line 27 of the example do-file), though they’re just there to make the reading easier and don’t have functionality. Since an index is not automated, create this manually in the code header by copying and pasting section titles (see lines 8-10 in the example do-file). You can then add and navigate through them using the find functionality. Since Stata code is harder to navigate, as you will need to scroll through the document, it’s particularly important to avoid writing very long scripts. Therefore, in Stata at least, we recommend breaking code tasks down into separate do-files, since there is no limit on how many you can have, how detailed their names can be, and no advantage to writing longer files. One reasonable rule of thumb is to not write do-files that have more than 200 lines. This is an arbitrary limit, just like the common practice of limiting code lines to 80 characters: it seems to be “enough but not too much” for most purposes.

### Demand for Safe Spaces Case Study: Writing Code That Others Can Read

To ensure that all team members were able to easily read and understand data work, *Demand for Safe Spaces* code files were extensively commented. Comments typically took the form of “what – why”: what is this section of code doing, and why is it necessary. The below snippet from a do-file cleaning one of the raw data files illustrates the use of comments:



The full code file is available at <https://git.io/Jtgev>

### Writing code that others can run

To bring all these smaller code files together, you must maintain a master script.[[72]](#footnote-189) A master script is the map of all your project’s data work which serves as a table of contents for the instructions that you code. Anyone should be able to follow and reproduce all your work from raw data to all outputs by simply running this single script. By follow, we mean someone external to the project who has the master script and all the input data can (i) run all the code and recreate all outputs, (ii) have a general understanding of what is being done at every step, and (iii) see how codes and outputs are related. The master script is also where all the settings are established, such as versions, folder paths, functions, and constants used throughout the project.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* TEMPLATE MASTER DO-FILE \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* \*  
\* PURPOSE: Reproduce all data work, map inputs and outputs, \*  
\* facilitate collaboration \*  
\* \*  
\* OUTLINE: PART 1: Set standard settings and install packages \*  
\* PART 2: Prepare folder paths and define programs \*  
\* PART 3: Run do-files \*  
\* \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 PART 1: Install user-written packages and harmonize settings  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
 local user\_commands ietoolkit iefieldkit //Add required user-written commands  
 foreach command of local user\_commands {  
 cap which `command'  
 if \_rc == 111 ssc install `command'  
 }  
  
 \*Harmonize settings across users as much as possible  
 ieboilstart, v(13.1)  
 `r(version)'  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 PART 2: Prepare folder paths and define programs  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
 \* Research Assistant folder paths  
 if "`c(username)'" == "ResearchAssistant" {  
 global github "C:/Users/RA/Documents/GitHub/d4di/DataWork"  
 global dropbox "C:/Users/RA/Dropbox/d4di/DataWork"  
 global encrypted "M:/DataWork/EncryptedData"  
 }  
  
 \* Baseline folder globals  
 global bl\_encrypt "${encrypted}/Round Baseline Encrypted"  
 global bl\_dt "${dropbox}/Baseline/DataSets"  
 global bl\_doc "${dropbox}/Baseline/Documentation"  
 global bl\_do "${github}/Baseline/Dofiles"  
 global bl\_out "${github}/Baseline/Output"  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 PART 3: Run do-files  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
/\*------------------------------------------------------------------------------  
 PART 3.1: De-identify baseline data  
 REQUIRES: ${bl\_encrypt}/Raw Identified Data/D4DI\_baseline\_raw\_identified.dta  
 CREATES: ${bl\_dt}/Raw Deidentified/D4DI\_baseline\_raw\_deidentified.dta  
 IDS VAR: hhid  
----------------------------------------------------------------------------- \*/  
 \*Change the 0 to 1 to run the baseline de-identification dofile  
 if (0) do "${bl\_do}/Cleaning/deidentify.do"  
  
/\*------------------------------------------------------------------------------  
 PART 3.2: Clean baseline data  
 REQUIRES: ${bl\_dt}/Raw Deidentified/D4DI\_baseline\_raw\_deidentified.dta  
 CREATES: ${bl\_dt}/Final/D4DI\_baseline\_clean.dta  
 ${bl\_doc}/Codebook baseline.xlsx  
 IDS VAR: hhid  
----------------------------------------------------------------------------- \*/  
 \*Change the 0 to 1 to run the baseline cleaning dofile  
 if (0) do "${bl\_do}/Cleaning/cleaning.do"  
  
/\*-----------------------------------------------------------------------------  
 PART 3.3: Construct income indicators  
 REQUIRES: ${bl\_dt}/Final/D4DI\_baseline\_clean.dta  
 CREATES: ${bl\_out}/Raw/D4DI\_baseline\_income\_distribution.png  
 ${bl\_dt}/Intermediate/D4DI\_baseline\_constructed\_income.dta  
 IDS VAR: hhid  
----------------------------------------------------------------------------- \*/  
 \*Change the 0 to 1 to run the baseline variable construction dofile  
 if (0) do "${bl\_do}/Construct/construct\_income.do"

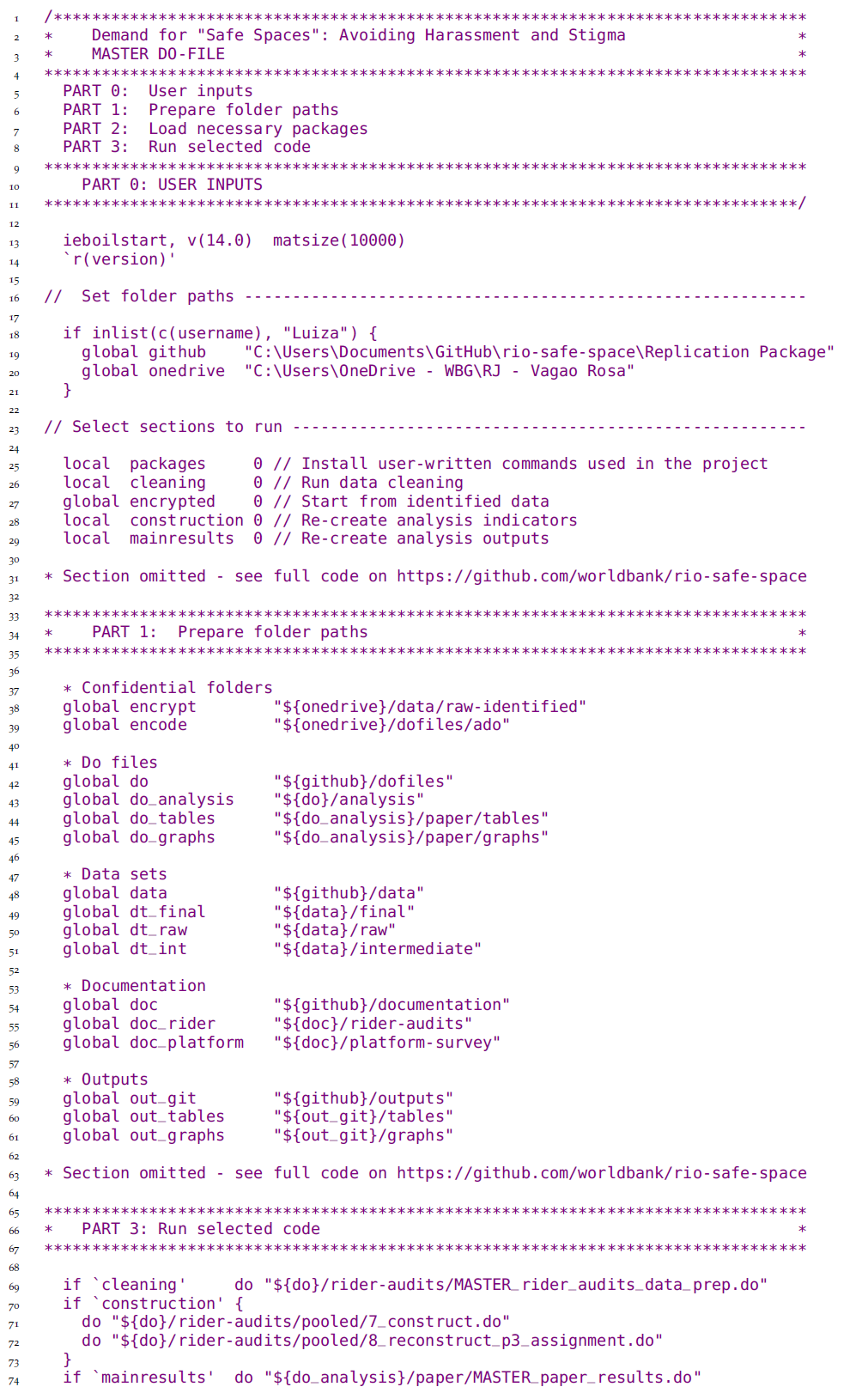
Try to create the habit of running your code from the master script. Creating “section switches” using macros or objects to run only the codes related to a certain task should always be preferred to manually open different scripts to run them in a certain order (see the if (0) switches in Part 3 of stata-master-dofile.do for one way of how to do this). Furthermore, running all scripts related to a particular task through the master whenever one of them is edited helps you identify unintended consequences of the changes you made. Say, for example, that you changed the name of a variable created in one script. This may break another script that refers to this variable. But unless you run both of them when the change is made, it may take time for that to happen, and when it does, it may take time for you to understand what’s causing an error. The same applies to changes in datasets and results.

To link code, data and outputs, the master script reflects the structure of the DataWork folder in code through globals (in Stata) or string scalars (in R); refer to lines 38-43 of the example do-file. These coding shortcuts can refer to subfolders, so that those folders can be referenced without repeatedly writing out their absolute file paths. Because the DataWork folder is shared by the whole team, its structure is the same in each team member’s computer. The only difference between machines should be the path to the project root folders, i.e. the highest-level shared folder. Depending on your software environment you may have multiple root folders. In a typical DIME project we have one Git root folder for our code, one sync software root folder for our de-identified data, and a third for our encrypted data. This is reflected in the master script in such a way that the only change necessary to run the entire code for a new team member is to change the path to the project root folders to reflect the file system and username; refer to lines 30-35 of the example do-file. The code in stata-master-dofile.do shows how folder structure is reflected in a master do-file. Because writing and maintaining a master script can be challenging as a project grows, an important feature of the iefolder is to write sub-master do-files and add to them whenever new subfolders are created in the DataWork folder.

In order to maintain well-documented and organized code, you should agree with your team on a plan to review code as it is written. Reading other people’s code is the best way to improve your coding skills. And having another set of eyes on your code will make you more comfortable with the results you find. It’s normal (and common) to make mistakes as you write your code. Reading it again to organize and comment it as you prepare it to be reviewed will help you identify them. Try to have a code review scheduled frequently, every time you finish writing a piece of code, or complete a small task. If you wait for a long time to have your code reviewed, and it gets too complex, preparation and code review will require more time and work, and that is usually the reason why this step is skipped. One other important advantage of code review is that making sure that the code is running properly on other machines, and that other people can read and understand the code easily, is the easiest way to be prepared in advance for a smooth project handover or for release of the code to the general public.

### Demand for Safe Spaces Case Study: Writing Code That Others Can Run

All code for the *Demand for Safe Spaces* study was organized to run from two master scripts, one Stata Master and one R Master. The master scripts were written such that any team member could run all project code by simply changing the top-level directory. Below is a snippet of the Stata Master:



The complete Stata master script can be found at <https://git.io/JtgeT>, and the R master at <https://git.io/JtgeY>.

## Preparing to handle confidential data ethically

Anytime you are working with original data in a development research project, you are almost certainly handling data that include **personally-identifying information (PII)**.[[73]](#footnote-196) PII is information which can, without any transformation or linkage, be used to identify individual people, households, firms, (or other units) in your data. Some examples of PII variables include names, addresses, and geolocations, email addresses, phone numbers, and bank accounts or other financial details. If you are working in a context or population that is either small, specific, or has extensive linkable data sources available to others, information like someone’s age and gender may be sufficient to disclose their identify, even though those variables would not be considered PII in general.

In a collaborative project, you will sometimes need to transfer and work with PII information, and sometimes you will prefer to remove or mask PII before transferring or working with data. There is no one-size-fits-all solution to determine what is PII, research teams have to use careful judgment in each case to avoid statistical disclosure.[[74]](#footnote-198) It is important to keep in mind that data privacy principles apply not only for the respondent giving you the information but also for their household members or other individuals who are included in the data.

In all cases where confidential information is involved, you must make sure that you adhere to several core principles. These include ethical approval, participant consent, data security, and participant privacy.[[75]](#footnote-199) If you are a US-based researcher, you will become familiar with a set of governance standards known as “The Common Rule.”[[76]](#footnote-200) If you interact with European institutions or persons, you will also become familiar with the General Data Protection Regulation (GDPR),[[77]](#footnote-201) a set of regulations governing **data ownership** and privacy standards. No matter who you are or what exact legal requirements you face, the core principles and practices you need to consider will always be similar.

### Seeking ethical approval

Most of the field research done in development involves human subjects.[[78]](#footnote-204) As a researcher, you are asking people to trust you with personal information about themselves: where they live, how rich they are, whether they have committed or been victims of crimes, their names, their national identity numbers, and all sorts of other data. PII data carries strict expectations about data storage and handling, and it is the responsibility of the research team to satisfy these expectations.[[79]](#footnote-205) Your donor or employer will most likely require you to hold a certification from a respected source.

For almost all such data collection and research activities, you will be required to complete some form of **Institutional Review Board (IRB)** process.[[80]](#footnote-207) Most commonly this consists of a formal application for approval of a specific protocol for consent, data collection, and data handling.[[81]](#footnote-208) Which IRB has authority over your project is not always apparent, particularly if some institutions do not have their own. It is customary to obtain an approval from a university IRB where at least one PI is affiliated, and if work is being done in an international setting, approval is often also required from an appropriate local institution subject to the laws of the country where data originates.

IRB approval should be obtained well before any data is acquired. IRBs may have infrequent meeting schedules or require several rounds of review for an application to be approved. If there are any deviations from an approved plan or expected adjustments, report these as early as possible so that you can update or revise the protocol. IRBs have the authority to retroactively deny the right to use data which was not acquired in accordance with an approved plan. This is extremely rare, but shows the seriousness of these considerations since the institution itself may face legal penalties if its IRB is unable to enforce them. As always, as long as you work in good faith, you should not have any issues complying with these regulations.

### Demand for Safe Spaces Example – Seeking Ethical Approval

The Duke University IRB reviewed and approved the protocol for all components of fieldwork for the *Demand for Safe Spaces* study (IRB number D0190). As one of the PIs was at Duke, and the World Bank does not have an IRB, Duke was the relevant institution in this case. The study was registered with the Duke IRB on September 2015. It was amended 4 times to reflect additions to the study design, such as an implicit association test (IAT) and a new survey. The IRB approval was renewed twice, in 2016 and 2017. Highlights from the study IRB protocols:

* *Voluntary study participation:* The study intervention was done through a smartphone application. Through the app, users were offered payment to complete a set of tasks while using the metro. The tasks involved answering questions at different moments of the trip (before boarding the train, during the ride, after leaving the train). At the start of each task, participants could review what it comprised and the exact payment value, then decide whether to accept the task or not. There was no obligation to complete any task.
* *Survey instruments:* translated drafts of all survey instruments were shared with the IRB
* *Privacy protection:* the intervention was done specialized mobile application developed by a partner technology company, which recruited users through social media. User data from the app was encrypted and stored in AWS. As per the user agreement, access to the raw data was restricted to employees of the tech company, using VPN and encrypted laptops. The tech company processed the raw data and released non-identifying data to the researchers, plus household coordinates (study participants provided informed consent to share this data).
* *Risk:* Participants were tasked with riding the public transport system in Rio De Janeiro. There is some general risk inherent in travel around Rio de Janeiro. However, the public transport system is widely used, and participants are expected to be those who regularly travel on the train. The assigned task may cause them to travel on a different route or at a different time than usual. Tasks are assigned around rush hour, so stations and trains will be crowded, which is expected to reduce risks. Note that half the riders on the system are women, and only a small fraction of the cars are reserved for women, so the task of riding the regular carriage will not require users to go into an all-male environment.
* *Ethical obligation:* when completing an assigned task, participants were asked whether they experienced any harassment. If harassment was reported, the app directed the participant to the platform guards to whom she could report harassment incidences (the guards are trained to respond to harassment reports), as well as to other resources available in the Rio area.

Appendix C in the working paper discusses the ethics aspects of the study, including participant recruitment, informed consent and how reports of harassment were addressed: <https://openknowledge.worldbank.org/handle/10986/33853>

### Obtaining informed consent

One primary consideration of IRBs is the protection of the people about whom information is being collected and whose lives may be affected by the research design. Some jurisdictions (especially those who governed by EU law) view all personal data as intrinsically owned by the persons who they describe. This means that those persons have the right to refuse to participate in data collection before it happens, as it is happening, or after it has already happened. It also means that they must explicitly and affirmatively consent to the collection, storage, and use of their information for any purpose.

The development of appropriate consent processes is of primary importance. All survey instruments must include a module in which the sampled respondent grants informed consent to participate. Research participants must be informed of the purpose of the research, what their participation will entail in terms of duration and any procedures, any foreseeable benefits or risks, and how their identity will be protected.[[82]](#footnote-212) There are special additional protections in place for vulnerable populations, such as minors, prisoners, and people with disabilities, and these should be confirmed with relevant authorities if your research includes them.

### Demand for Safe Spaces Case Study: Obtaining Informed Consent

Participation in both the study intervention (assignments to take a specific Supervia ride) and the platform survey were fully voluntary, and both included informed consent. Per the informed consent protocols for the study intervention, participation in each assigned task was voluntary, and participants were paid for each ride they completed shortly after completion, regardless of the total number of rides they completed. Thus, participants could choose to stop participating at any time if they felt uncomfortable.

The consent statement participants were shown was the following:

“If you choose to do this task, you will have to go to a Supervia station, ride the train, and answer questions about your experience on the train. You will have to ride the train for \_\_\_ minutes, and answering the questions will take about ten minutes. You will be paid at least \_\_\_\_ reais for the task, and possibly more. You will be able to review the payment for each task and option before deciding whether to do that task. You can choose during the task whether to ride either the women’s-only or the mixed carriage on the Supervia. Your responses to the task will not be identified in any way with you personally. The anonymous data will be shared with researchers at Duke University in the United States. You can choose to stop the task at any time. To get paid for this task, however, you have to finish the task.”

The sentence in italics was removed for the portion of the data collection in which participants were assigned to ride in a particular type of car; the rest of the consent statement applied to all tasks.

### Ensuring research subject privacy

In order to safeguard PII data and protect respondent privacy, you must set up a data protection protocol from the outset of a project. Secure data storage and transfer are ultimately your personal responsibility.[[83]](#footnote-216) Later chapters will discuss how to properly protect your data depending on which method you are using to acquire, store or share data. This section will only cover the computer setup you will need for any project. There are several components to this.

First, you need a system for managing strong and unique passwords for all accounts – including personal accounts like computer logins and email. This means that all your passwords should be long, not use common words and should not be reused for multiple accounts. The only way to make that practically feasible is to use a password manager.[[84]](#footnote-218) Most password managers also allow you to securely share passwords for shared accounts with your colleagues. Multi-factor authentication (sometimes called 2-step verification) is a secure alternative to passwords when available.

Second, machines that stores confidential data should not be connected to insecure physical or wireless networks, and when it is necessary to do so, they should use a VPN[[85]](#footnote-219) to connect to the internet. Furthermore, USB drives and other devices connecting over USB cables should not be connected to machines storing confidential data unless you know where the USB device came from and who has used it before you. Machines with confidential data should also be stored in a secure location when not in use.

Third, all confidential data must be encrypted at all times.[[86]](#footnote-220) When files are properly encrypted, the information they contain will be completely unreadable and unusable even if they were to be intercepted by a malicious “intruder” (an information-security term for any unauthorized information access) or accidentally made public. We will discuss implementations of encryption in more detail specific to different stages of data work in Chapter 4, but when setting up a software environment for you and your team, make sure that you have a solution for all parts of your workflow on all team members computers. You can encrypt your data at the disk (hard drive) level, called **full disk encryption** (FDE), or at the individual file or folder level, called **file system encryption** (FSE). When possible both should be applied, but our recommendation is that project teams set up data protection protocols using file system encryption as the main type of protection, and require all members that handle confidential data to use it.

Most implementations of FDE use your computer login password to prevent your files from ever being read by anyone but you. It is important to note that password protection alone is not sufficient. Password protection makes it more difficult for someone else to gain access to your files, but only encryption properly protects your data if someone manages to access your files anyway. Whenever FDE is available in your operating system it should always be enabled. FDE protects your data when encrypted just as well as FSE, but we recommend FSE due to the following disadvantages of FDE. First, FDE is implemented differently in different operating systems making it difficult to create useful instructions for all computer set-ups team members may use. Second, when FDE decrypts your disk, then all data on that disk is decrypted, even files you do not need to use at that time. Since most FDE systems automatically decrypt your full disk each time you log in, a malicious intruder that has gained access to your computer would have access to all your files. Third and perhaps most important, FDE cannot be used when sharing confidential data over insecure channels like file syncing services or email. So FDE would anyways have to be complimented with some other type of protection during collaboration.

When using FSE, instead of encrypting a full disk or drive, you create encrypted folders in which you can securely store confidential data. These encrypted folders protect your data when they are stored on your computer but can also be used to securely transfer data over insecure channels like file syncing services and email. Encrypted folders in FSE are only decrypted when you need that specific folder to be decrypted. That means that a malicious intruder that has gained access to your computer only gains access to the folders you have decrypted while your computer was compromised. Folders you rarely or never decrypt therefore remain protected even if someone gains access to your computer. DIME uses VeraCrypt for FSE, and our protocols are available as part of the DIME Research Standards.[[87]](#footnote-222) VeraCrypt is free of charge and available for Windows, MacOS, and Linux. While some details are different across platforms, the encryption will be implemented the same way for all team members.

Regardless of whether you use full disk encryption or file system encryption, it is important to remember that encryption provides no protection when your data is decrypted. Therefore, you should always log out from your computer when you are not using it, and only keep folders with confidential data decrypted when you need exactly those files. The latter is only possible when you are using FSE.

Handling confidential data properly will always add to your workload. The easiest way to reduce that work load is to handle it as rarely as possible. Whenever our research design allows us to, we should not work with confidential data at all, and not collect it or ask for it in the first place. Even when we do need confidential data for some aspect of our research, it is almost never the case that we need it for *all* aspects of our data. It is often very simple to conduct planning and analytical work using a subset of the data that does not include this type of information. Therefore we recommend that all projects plan a workflow with a version of the data where confidential data has been removed and always use that dataset when possible.

Note that it is in practice impossible to **anonymize** data. There is always some statistical chance that an individual’s identity will be re-linked to the data collected about them – even if that data has had all directly identifying information removed – by using some other data that becomes identifying when analyzed together. For this reason, we recommend de-identification in two stages. The **initial de-identification** process strips the data of direct identifiers as early in the process as possible, to create a working de-identified dataset that can be shared *within the research team* without the need for encryption. This data set should always be used when possible. The **final de-identification** process involves making a decision about the trade-off between risk of disclosure and utility of the data before publicly releasing a dataset.[[88]](#footnote-223)

Finally, it is essential to have an end-of-life plan for data even before it is acquired.[[89]](#footnote-225) This includes plans for how to transfer access and control to a new person joining the team, and how to revoke that access when someone is leaving the team. It should also include a plan for how the confidential data should be deleted. Every project should have a clear data retention and destruction plan. After a project is completed and its de-identified data has been made available as a part of data publication, research teams should not retain confidential data indefinitely.

### Demand for Safe Spaces Example: Ensuring Research Subject Privacy

The *Demand for Safe Spaces* team adopted the following data security protocols:

* All confidential data was stored in a World Bank OneDrive folder The World Bank One Drive has been set up by WB IT to be more secure than regular OneDrive and is the recommended institutional solution for storing confidential data.
* Access to the confidential data was limited to the Research Analyst and the Research Assistant working on the data cleaning.
* All de-identified data used for the analysis was stored in the synchronized folder shared by the full research team (in this case, using Dropbox).
* Indirect identifiers such as demographic variables and labels to train lines and stations were removed from the data before it was published to the Microdata Catalog.

## Looking ahead

With your code environment established, you will have a firm idea about how you are going to handle the data and code that you receive and create throughout the research process. This structure should prepare you to work collaboratively, to share code and data across machines and among team members, and to document your work as a group. With an organization plan and plans to to version-control and back up files, you are ready to handle materials ethically and securely. You should also have secured the approvals needed for any planned work. You are now ready to translate your project’s research design into a measurement framework to answer your research questions. In the next chapter, we will outline how to prepare the essential elements of research data. You will learn how to map out your project’s data needs according to both the research design and the planned creation and use of data across the project timeline.

# Bibliography

# DEVELOPMENT RESEARCH IN PRACTICE: THE DIME ANALYTICS DATA HANDBOOK

# Acknowledgments and notes

Placeholder

## How to read this book

## The DIME Wiki: A complementary resource

## Standardizing data work

## Standardizing coding practices

## The team behind this book

## Looking ahead

# Conducting reproducible, transparent, and credible research

Placeholder

## Developing a credible research project

### Registering research studies

### Writing pre-analysis plans

### Publishing registered reports

## Conducting research transparently

### Documenting data acquisition and analysis

### Cataloging and archiving data

## Analyzing data reproducibly

### Preparing a reproducibility package

## Looking ahead

# Setting the stage for effective and efficient collaboration

In order to do effective data work in a team environment, you need to structure your workflow in advance. Preparation for collaborative data work begins long before you acquire any data, and involves planning both software tools and collaboration platforms for your team. This means knowing what types of data you’ll acquire, whether the data will require special handling due to size or privacy considerations, which datasets and outputs you will need at the end of the process, and how all data files and versions will stay organized throughout. It’s important to plan data workflows in advance because changing software or protocols halfway through a project is costly and time-consuming. Seemingly small decisions such as file-sharing services, folder structures, and filenames can be extremely painful to alter down the line in any project.

This chapter will guide you in setting up an effective environment for collaborative data work, structuring your data work to be well-organized and clearly documented, and setting up processes to handle confidential data securely. The first section outlines how to set up your working environment to effectively collaborate on technical tasks with others, and how to document tasks and decisions. The second section discusses how to organize your code and data so that others will be able to understand and interact with it easily. The third section provides guidelines for ensuring privacy and security when working with confidential data.

### Summary: Setting the stage for effective and efficient collaboration Summary

The technical environment for your data work needs to be established at the start of a research project. Agreeing with the team on software choices, standard code and data structure, and clear data security protocols will prepare you to successfully, safely, and efficiently implement technical tasks throughout the project lifecycle. Consider:

1. **The technical collaboration environment.** No matter the hardware and software your team plans to use, you should ensure now that it is standardized or interoperable across the team. This includes:

* Secure all **physical computing hardware** through encryption and password-protection. If specialized or more powerful hardware is required, initiate access requests, purchase orders, or other processes now
* Agree on tools for **collaboration** and documentation, such that key conversations and decisions are archived and organized outside of instant-message and email conversation
* Decide the **programming languages and environments** the team will use. Take time to set up a comfortable and modern digital work environment

1. The **organization of code and data**. The team should agree on where and how code files and databases will be stored, down to the level of the folder structure. This involves setting up:

* A standardized and scalable **folder structure** so all documents have an unambiguous location, and the location and naming of files describes their purpose and function and is intuitive to all team members
* A **backup and version control system** appropriate for each file type, to ensure information cannot be lost and that all team members understand how to interoperate and collaborate
* **Master script files** that will structure and execute the code base of the

1. **Information security measures and ethical frameworks**. These include:

* Formally **request and obtain approval** from legal entities governing research in all relevant jurisdictions
* Understand **how to respect the rights and dignity of research subjects** and plan for how to establish **informed consent** from individuals or groups participating in the research
* Adopt standardized **digital security practices** including proper encryption of all confidential information, at rest and in transit, both among your team and with external partners

#### Takeaways

**TTLs/PIs will:**

* Support the acquisition and maintenance of required computing hardware and software, liaising with procurement, information security and information technology teams as necessary
* Make final decisions regarding code languages and environments
* Discuss and agree upon an appropriate project-wide digital organization strategy
* Institute and communicate best practices in accordance with legal, ethical, and security obligations

**RAs will:**

* Communicate technical needs clearly with TTLs/PIs and relevant service providers
* Consistently implement digital organization strategy and flag issues with task management, documentation, or materials storage if they arise
* Support project compliance with ethical, legal, and security obligations and flag concerns to TTLs/PIs

#### Key Resources

* DIME Research Ethics Standards: Pillar 1 of the DIME Research Standards <https://github.com/worldbank/dime-standards>
* DIME GitHub Resources: <https://github.com/worldbank/dime-github-trainings>
* DIME Data Security Standards: Pillar 4 of the DIME Research Standards <https://github.com/worldbank/dime-standards>
* DIME Data Publication Standards: Pillar 5 of the DIME Research Standards <https://github.com/worldbank/dime-standards>

## Preparing a collaborative work environment

This section introduces core concepts and tools for organizing data work in an efficient, collaborative and reproducible manner. Some of these skills may seem elementary, but thinking about simple things from a workflow perspective can help you make marginal improvements every day you work; those add up to substantial gains over the course of multiple years and projects. Together, these processes form a collaborative workflow that will greatly accelerate your team’s ability to get tasks done on all your projects.

Teams often develop workflows in an ad hoc fashion, solving new challenges as they arise. Adaptation is good, of course. But it is important to recognize that there are a number of tasks that exist in common for every project, and it is more efficient to agree on the corresponding workflows in advance. For example, every project requires research documentation, organized file naming, directory organization, coding standards, version control, and code review. These tasks are common to almost every project, and their solutions translate well between projects. Therefore, there are large efficiency gains to thinking in advance about the best way to do these tasks, instead of throwing together a solution when the task arises. This section outlines the main points to discuss within the team, and suggests best practice solutions for these tasks.

### Demand for Safe Spaces Case Study: Preparing a Collaborative Work Environment

Here are a few highlights of how the **Demand for Safe Spaces** team chose to organize their work environment for effective collaboration:

* The data work for the project was done through a private GitHub repository housed in the World Bank organization account.
* GitHub issues were used to document research decisions and to provide feedback. Even the PIs for the study, who did not directly participate in coding, used Github issues to review code and outputs and to create a record of broader discussions.
* Stata was adopted as the primary software for data analysis, as that is the software all team members had in common at the start of the project. At a later stage of the project, R code was developed specifically to create maps. The R portion of the code was developed independently, as it used different datasets and created separate outputs. The team used two separate master scripts, one for the Stata code base and one for the R code.
* The team members shared a synchronized folder (using Dropbox), which included the de-identified data and project documentation such as survey instruments and enumerator training manuals.

### Setting up your computer

First things first: almost all your data work will be done on your computer, so make sure it’s set up for success. The operating system should be fully updated, it should be in good working order, and you should have a **password-protected** login. However, password-protection is not sufficient if your computer stores data that is not public. You would need to use encryption for sufficient protection, which will be covered later in this chapter. Make sure your computer is backed up to prevent information loss. Follow the **3-2-1 rule**: maintain 3 copies of all original or irreplaceable data, on at least 2 different hardware devices you have access to, with 1 offsite storage method.[[90]](#footnote-255) Chapter 4 provides a protocol for implementing this.

Ensure you know how to get the **absolute file path** for any given file. On MacOS this will be something like “/users/username/git/project/...” and “C:/users/username/git/project/...” on Windows. Absolute file paths will be an obstacle to collaboration unless they are **dynamic absolute file paths**. In a dynamic absolute file path the relative project path, “/git/project/...” in the examples above, is added to the user-specific root path for each user, “/users/username” or “C:/users/username” in the examples above, generating an absolute file path unique to each user. Master scripts introduced later in this chapter will show how this can be seamlessly implemented. Dynamic absolute file paths, starting from the file system root, is the best way to ensure that files are read and written correctly when multiple users work on the same project across many different platforms, operative systems and devices. There are contexts, for example some cloud environments, where relative file paths must be used, but in all other contexts we recommend you to always use dynamic absolute file paths.

Use forward slashes (/) in file paths for folders, and whenever possible use only the 26 English characters, numbers, dashes (-), and underscores (\_) in folder names and filenames.[[91]](#footnote-256) For emphasis: *always* use forward slashes (/) in file paths in code, just like in internet addresses. Do this even if you are using a Windows machine where both forward and backward slashes are allowed, as your code will otherwise break if anyone tries to run it on a Mac or Linux machine. Making the structure of your directories a core part of your workflow is very important, since otherwise you will not be able to reliably transfer the instructions for replicating or carrying out your analytical work.

When you are working with others, you will most likely be using some kind of **file sharing** software. The exact services you use will depend on your tasks, but in general, there are several approaches to file sharing, and the three discussed here are the most common. **File syncing** is the most familiar method, and is implemented by software like OneDrive, Dropbox, or Box. Sync forces everyone to have the same version of every file at the same time, which makes simultaneous editing difficult but other tasks easier. **Distributed version control** is another method, commonly implemented through systems like GitHub, GitLab, and Bitbucket that interact with Git.[[92]](#footnote-257) Distributed version control allows everyone to access different versions of files at the same time. It is only optimized for specific types of files (for example, any type of code files). Finally, **server storage** is the least-common method, because there is only one version of the materials, and simultaneous access must be carefully regulated. Server storage ensures that everyone has access to exactly the same files and environment, and it also enables high-powered computing processes for large and complex data.

All three file sharing methods are used for collaborative workflows, and you should review the types of data work that you will be doing, and plan which types of files will live in which types of sharing services. It is important to note that they are, in general, not interoperable, meaning you should not have version-controlled files inside a syncing service, or vice versa, without setting up complex workarounds, and you cannot shift files between them without losing historical information. Therefore, choosing the correct sharing service for each of your team’s needs at the outset is essential. At DIME we typically use file syncing for all project administrative files and data, version control in Git for code, and server storage for backup and/or large scale computations when needed.

### Establishing effective documentation practices

Once your technical and sharing workspace is set up, you need to decide how you are going to communicate with your team. The first habit that many teams need to break is using instant communication for management and documentation. Email is, simply put, not a system. It is not a system for anything. Neither is instant messaging apps like WhatsApp. Instant messaging tools are developed for communicating “now” and that is what they do well. They are not structured to manage group membership or to present the same information across a group of people, or to remind you when old information becomes relevant. They are not structured to allow people to collaborate over a long time or to review old discussions. It is therefore easy to miss or lose communications from the past when they have relevance in the present. Everything with future relevance that is communicated over email or any other instant medium – such as, for example, decisions about research design – should immediately be recorded in a system that is designed to keep permanent records. We call these systems collaboration tools, and there are several that are very useful.

Good collaboration tools are workflow-oriented systems that allow the team to create and assign tasks, carry out discussions related to single tasks, track task progress across time, and quickly see the overall project status. They are web-based so that everyone on your team can access them simultaneously and have ongoing discussions about tasks and processes. Such systems link communications to specific tasks so that related decisions are permanently recorded and easy to find in the future when questions about that task come up. Choosing the right tool for your team’s needs is essential to designing an effective workflow. What is important is that your team chooses a system and commits to using it, so that decisions, discussions, and tasks are easily reviewable long after they are completed.

Some popular and free collaboration tools that meet these criteria are GitHub and Dropbox Paper. Any specific list of software will quickly be outdated; we mention these as examples that have worked for our team. Different collaboration tools can be used different types of tasks. Our team, for example, uses GitHub for code-related tasks, and Dropbox Paper for more managerial tasks. GitHub creates incentives for writing down why changes were made in response to specific discussions as they are completed, creating naturally documented code. It is useful also because tasks in GitHub Issues can clearly be tied to file versions. On the other hand, Dropbox Paper provides a clean interface with task notifications, assignments, and deadlines, and is very intuitive for people with non-technical backgrounds. Therefore, it is a useful tool for managing non-code-related tasks.

### Setting up your code environment

Taking time early in your project to choose a programming language to work in and setting up a productive code environment for that language will make your work significantly easier. Setting up a productive code environment means to make sure that the programming language and all other software your code requires will run smoothly on all the hardware you need it to run on. It also means that you have a productive way of interacting with code, and that the code has a seamless method to access your data.

It is difficult and costly to switch programming languages halfway through a project, so think ahead about the various software your team will use. Take into account the technical abilities of team members, what type of internet access the software will need, the type of data you will need to access, and the level of security required. Big datasets require additional infrastructure and may overburden the tools commonly used for small datasets. Also consider the cost of licenses, the time to learn new tools, and the stability of the tools. There are few strictly right or wrong choices for software, but what is important is that you plan in advance and understand how the chosen tools will interact with your workflows.

One option is to hold your code environment constant over the lifecycle of a single project. While this means you will inevitably have different projects with different code environments, each successive project will be better than the last, and you will avoid the costly process of migrating an ongoing project into a new code environment. Code environments should be documented as precisely as possible. The specific version number of the programming languages and the individual packages you use should be referenced or maintained so that they can be reproduced going forward, even if different releases contain changes that would break your code or change your results. DIME Analytics developed the command ieboilstart in the ietoolkit package to support version and settings stability in Stata.[[93]](#footnote-260) If your project requires more than one programming language, for example if you analyze your data in one language but visualize your results in another, then make sure to make an as clear division between the two as possible. This means that you first complete all tasks done in one language, before the completing the rest of the tasks in the other language. Frequently swapping back and forth between languages is a reproducibility nightmare.

Next, think about how and where you write and execute code. This book is intended to be agnostic to the size or origin of your data, but we are going to broadly assume that you are using one of the two most popular statistical software packages: R or Stata. (If you are using another language, like Python, many of the same principles apply but the specifics will be different.) Most of your code work will be done in a code editor. If you are working in R, **RStudio** is the typical choice.[[94]](#footnote-261) For Stata, the built-in do-file editor is the most widely adopted code editor. You might also consider using an external editor for your R or Stata code.[[95]](#footnote-262) These editors offer great accessibility and quality features. For example, they can access an entire directory – rather than a single file – which gives you directory-level views and file management actions, such as folder management, Git integration, and simultaneous work with other types of files, without leaving the editor. Using an external editor can also be preferable since your editor will not crash if the execution of your code causes your statistical software to crash. Finally, you can often use the same editor for all programming languages you use, so any customization you do in your code editor of choice will improve your productivity across all your coding work.

## Organizing code and data for replicable research

We assume you are going to do your analytical work through code, and that you want all your processes to be documented and replicable. Though it is possible to interact with some statistical software through the user interface without writing any code, we strongly advise against it. Writing code creates a record of every task you performed. It also prevents direct interaction with the data files that could lead to non-reproducible steps. You may do some exploratory tasks by point-and-click or typing directly into the console, but anything that is included in a research output must be coded in an organized fashion so that you can release the exact code that produces your final results – up to and including individual statistics in text. Still, organizing code and data into files and folders is not a trivial task. What is intuitive to one person rarely comes naturally to another, and searching for files and folders is everybody’s least favorite task. As often as not, you come up with the wrong one, and then it becomes very easy to create problems that require complex resolutions later. This section provides basic tips on managing the folder that stores your project’s data work.

Maintaining an organized file structure for data work is the best way to ensure that you, your teammates, and others are able to easily edit and replicate your work in the future. It also ensures that automated processes from code and scripting tools are able to interact well with your work, whether they are yours or those of others. File organization makes data work easier as well as more transparent, and facilitates integration with tools like version control systems that aim to cut down on the amount of repeated tasks you have to perform. It is worth thinking in advance about how to store, name, and organize the different types of files you will be working with, so that there is no confusion down the line and everyone has the same expectations.

### Organizing files and folders

Once you start a research project, the number of scripts, datasets, and outputs that you have to manage will grow very quickly. This can get out of hand just as quickly, so it’s important to organize your data work and follow best practices from the beginning. You should agree with your team on a specific directory structure, and set it up at the beginning of the project. You should also agree on a file naming convention. This will help you to easily find project files and ensure that all team members can easily run the same code.

To support consistent folder organization at DIME, DIME Analytics created iefolder as a part of our ietoolkit package.[[96]](#footnote-265) This Stata command sets up a pre-standardized folder structure for what we call the DataWork folder.[[97]](#footnote-266) The DataWork folder includes folders for all the steps of a typical project. Since each project will always have unique needs, we have tried to make the structure easy to adapt. Having a universally standardized folder structure across the entire portfolio of projects means that everyone can easily move between projects without having to reorient on file and folder organization.

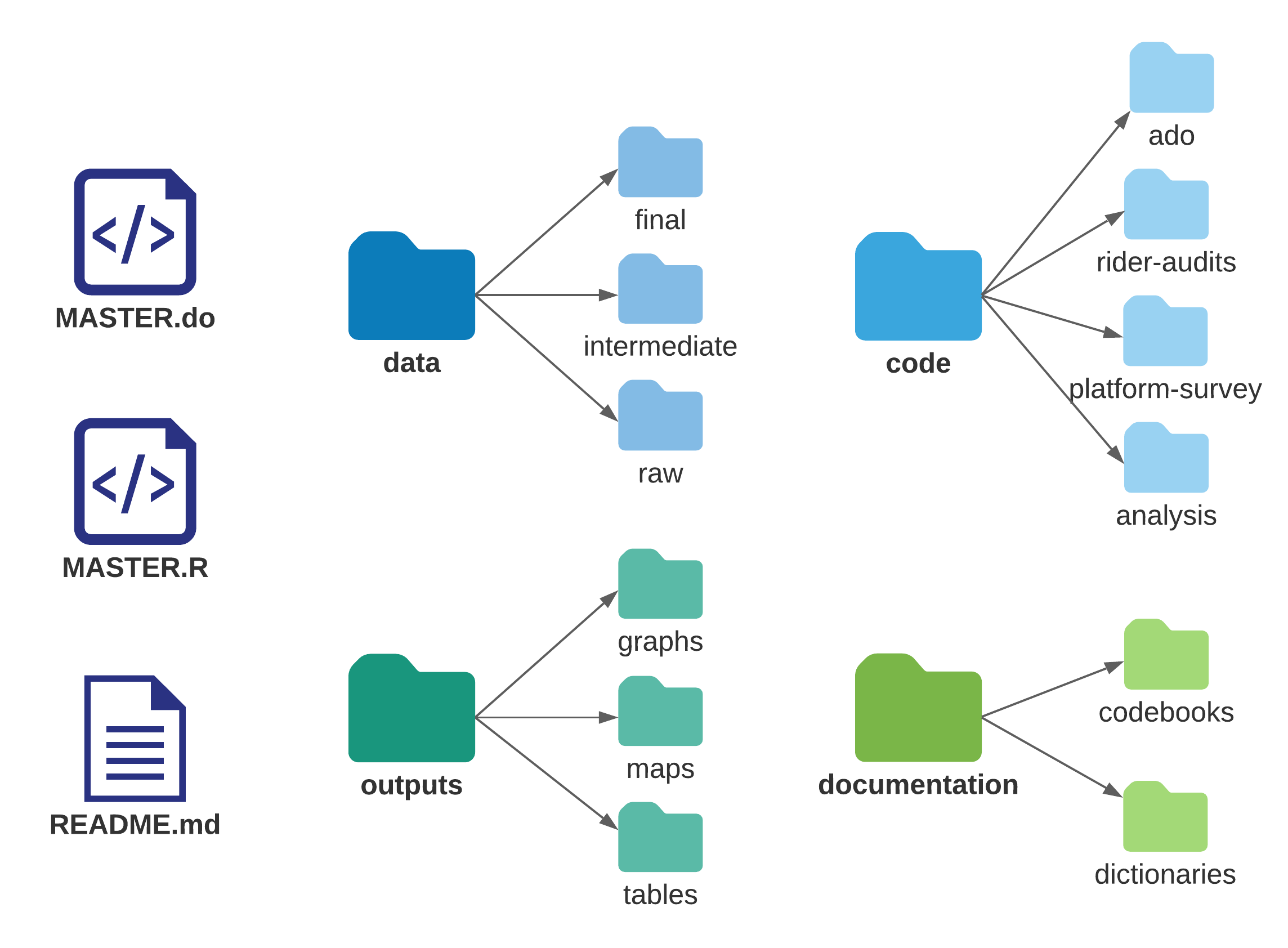
If you do not already have a standard file structure across projects, iefolder is an easy template to start from. The command creates a DataWork folder at the project level, and within that folder, creates standardized directory structures for each data source or survey round. Within each subdirectory, iefolder creates folders for raw encrypted data, raw deidentified data, cleaned data, final data, outputs, and documentation. It creates a parallel folder structure for the code files that move the data through this progression, and for the final analytical outputs. The ietoolkit package also includes the iegitaddmd command, which can place README.md placeholder files in your folders so that your folder structure can be shared using Git.[[98]](#footnote-267) Since these placeholder files are written in a plaintext language called **Markdown**, they also provide an easy way to document the contents of every folder in the structure.

The DataWork folder may be created either inside an existing project folder, or it may be created as a separate root folder. We advise keeping project management materials (such as contracts, terms of reference, briefs and other administrative or management work) separate from the DataWork folder structure. It is useful to maintain project management materials in a sync system like Dropbox, whereas the code folder should be maintained in a version-control system like Git. However, a version-controlled folder should *never* be stored in a synced folder that is shared with other people. Combining these two types of collaboration tools almost always creates undesired functionalities.

### Demand for Safe Spaces Case Study: Organizing Files and Folders

The diagram below illustrates the folder structure for the *Demand for Safe Spaces* data work. Note that this project started before the iefolder structured was created, and so differs from the current guidelines. Changing folder organization midway through a project is never recommended. The key feature of folder organization is planning ahead and agreeing with the whole team on the best structure to follow.

The root folder for the study’s data work included two master scripts (one for Stata and one R, that can be run independently), a readme, and 4 subfolders: data, code, documentation and outputs. The folder structure was identical in GitHub and DropBox, but the files present in each of them differed. All code, raw outputs (such as TeX table fragments and PNG images) and plain text documentation were stored in GitHub. All the datasets, PDF outputs and documentation in Word and Excel were stored in DropBox. When data processing was completed, binary files in the Documentation folder that were accessed by the code, such as iecodebook and ieduplicates spreadsheets, were moved to GitHub to ensure completeness of the repository.



### Establishing common file formats

Each task in the research workflow has specific inputs and outputs which feed into one another. It is common, particularly when different tasks are performed by different people inside a team, for incompatibilities to be created. For example, if the Principal Investigators are writing a paper using {}, exporting tables from statistical software into a .csv format will break the workflow. Therefore, it’s important to agree with your team on what tools will be used for what tasks, and where inputs and outputs will be stored, before you start creating them. Take into account ease of use for different team members, and keep in mind that learning how to use a new tool may require some time investment upfront that will be paid off as your project advances.

Knowing how code outputs will be used will help you decide the best format to export them. You can typically use the same software to save figures into various formats, such as .eps},.png,.pdfor.jpg`. However, the decision between using Office Suite software such as Word and PowerPoint versus {} and other plain text formats may influence how you write your code, as this choice often implicates in the use of a particular format. This decision will also affect the version control systems that your team can use.

### Using version control

We recommend using a **version control system** to maintain control of file history and functionality. A good version control system tracks who edited each file and when, allows you to revert to previous versions, and provides a protocol for ensuring that conflicting versions are avoided. This is important, for example, for your team to be able to find the version of a presentation that you delivered to a donor, or to understand why the significance level of your estimates has changed. Everyone who has ever encountered a file named something like final\_report\_v5\_LJK\_KLE\_jun15.docx can appreciate how useful such a system can be.

Most syncing services offer some kind of rudimentary version control; these are usually enough to manage changes to binary files (such as office documents) without needing to rely on dreaded filename-based versioning conventions. For code files, however, a more detailed version control system is usually desirable. We recommend using Git for version-control of all data work. Git documents changes to all **plaintext** files. Plaintext files include all code files, most raw outputs, and written outputs that use code languages, such as {} files and many dynamic documents. Git tracks all the changes you make to each plaintext file, and allows you to go back to previous versions without losing the information on changes made. It also makes it possible to work on multiple parallel versions of a file, so you don’t risk breaking code for other team members as you try something new.

### Writing code that others can read

Good code is written in a way that is easily understood and run by others. Below we discuss a few crucial steps to code organization. They all come from the principle that code is an output by itself, not just a means to an end, and should be written thinking of how easy it will be for someone to read it later. At the end of this section, we include a template for a master script do-file in Stata, to provide a concrete example of the required elements and structure. Throughout this section, we refer to lines of this example do-file to give concrete examples of the required code elements, organization and structure.

To be readable, code must be well-documented. Start by adding a code header to every file. A code header is a long **comment**[[99]](#footnote-272) that details the functionality of the entire script; refer to lines 5-10 in the example do-file. This should include simple things such as the purpose of the script and the name of the person who wrote it. If you are using a version control software, the last time a modification was made and the person who made it will be recorded by that software. Otherwise, you should include it in the header. You should always track the inputs and outputs of the script, as well as the uniquely identifying variable; refer to lines 52-54 in the example do-file. When you are trying to track down which code creates which dataset, this will be very helpful. While there are other ways to document decisions related to creating code, the information that is relevant to understand the code should always be written in the code file.

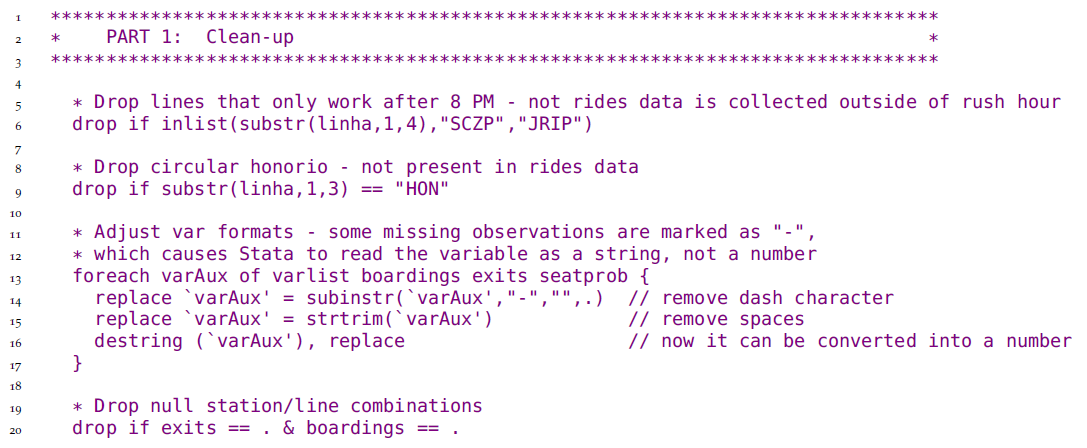
Two types of comments should be included in the script itself. The first type of comment describes *what* is being done. This might be easy to understand from the code itself if you know the language well enough and the code is clear, but often it is still a great deal of work to reverse-engineer the code’s intent. Describing the task in plain English (or whichever language you use to communicate with your team) will make it easier for everyone to read and understand the code’s purpose. It can also help you organize your own work and ensure you are following logical steps. The second type of comment explains *why* the code is performing a task in a particular way. As you are writing code, you are making a series of decisions that (hopefully) make perfect sense to you at the time. These are often highly specialized and may exploit a functionality that is not obvious or has not been seen by others before. Well-commented code is in itself a great way to document your data work which someone can follow to understand anything from data cleaning decisions that make the published data differ from the original data to decisions on how indicators are constructed. Even you will probably not remember the exact choices that were made in a couple of weeks. Therefore, you must document your precise processes in your code.

Code files should be stored in an easy-to-find location and named in a meaningful way. Breaking your code into independently readable “chunks” is good practice for code organization. You should write each functional element as a chunk that can run completely on its own. This ensures that each code component is independent; it does not rely on a complex program state created by other code chunks that are not obvious from the immediate context. One way to do this is by creating sections in your script to identify where a specific task is completed. For example, if you want to find the line in your code where the directory is set, you can go straight to “PART 2: Prepare folder paths and define programs”, instead of reading line by line through the entire code.

RStudio makes it very easy to create sections, and it compiles them into an interactive script index for you. In Stata, you can use comments to create section headers (see line 27 of the example do-file), though they’re just there to make the reading easier and don’t have functionality. Since an index is not automated, create this manually in the code header by copying and pasting section titles (see lines 8-10 in the example do-file). You can then add and navigate through them using the find functionality. Since Stata code is harder to navigate, as you will need to scroll through the document, it’s particularly important to avoid writing very long scripts. Therefore, in Stata at least, we recommend breaking code tasks down into separate do-files, since there is no limit on how many you can have, how detailed their names can be, and no advantage to writing longer files. One reasonable rule of thumb is to not write do-files that have more than 200 lines. This is an arbitrary limit, just like the common practice of limiting code lines to 80 characters: it seems to be “enough but not too much” for most purposes.

### Demand for Safe Spaces Case Study: Writing Code That Others Can Read

To ensure that all team members were able to easily read and understand data work, *Demand for Safe Spaces* code files were extensively commented. Comments typically took the form of “what – why”: what is this section of code doing, and why is it necessary. The below snippet from a do-file cleaning one of the raw data files illustrates the use of comments:



The full code file is available at <https://git.io/Jtgev>

### Writing code that others can run

To bring all these smaller code files together, you must maintain a master script.[[100]](#footnote-275) A master script is the map of all your project’s data work which serves as a table of contents for the instructions that you code. Anyone should be able to follow and reproduce all your work from raw data to all outputs by simply running this single script. By follow, we mean someone external to the project who has the master script and all the input data can (i) run all the code and recreate all outputs, (ii) have a general understanding of what is being done at every step, and (iii) see how codes and outputs are related. The master script is also where all the settings are established, such as versions, folder paths, functions, and constants used throughout the project.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* TEMPLATE MASTER DO-FILE \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* \*  
\* PURPOSE: Reproduce all data work, map inputs and outputs, \*  
\* facilitate collaboration \*  
\* \*  
\* OUTLINE: PART 1: Set standard settings and install packages \*  
\* PART 2: Prepare folder paths and define programs \*  
\* PART 3: Run do-files \*  
\* \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 PART 1: Install user-written packages and harmonize settings  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
 local user\_commands ietoolkit iefieldkit //Add required user-written commands  
 foreach command of local user\_commands {  
 cap which `command'  
 if \_rc == 111 ssc install `command'  
 }  
  
 \*Harmonize settings across users as much as possible  
 ieboilstart, v(13.1)  
 `r(version)'  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 PART 2: Prepare folder paths and define programs  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
 \* Research Assistant folder paths  
 if "`c(username)'" == "ResearchAssistant" {  
 global github "C:/Users/RA/Documents/GitHub/d4di/DataWork"  
 global dropbox "C:/Users/RA/Dropbox/d4di/DataWork"  
 global encrypted "M:/DataWork/EncryptedData"  
 }  
  
 \* Baseline folder globals  
 global bl\_encrypt "${encrypted}/Round Baseline Encrypted"  
 global bl\_dt "${dropbox}/Baseline/DataSets"  
 global bl\_doc "${dropbox}/Baseline/Documentation"  
 global bl\_do "${github}/Baseline/Dofiles"  
 global bl\_out "${github}/Baseline/Output"  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 PART 3: Run do-files  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
/\*------------------------------------------------------------------------------  
 PART 3.1: De-identify baseline data  
 REQUIRES: ${bl\_encrypt}/Raw Identified Data/D4DI\_baseline\_raw\_identified.dta  
 CREATES: ${bl\_dt}/Raw Deidentified/D4DI\_baseline\_raw\_deidentified.dta  
 IDS VAR: hhid  
----------------------------------------------------------------------------- \*/  
 \*Change the 0 to 1 to run the baseline de-identification dofile  
 if (0) do "${bl\_do}/Cleaning/deidentify.do"  
  
/\*------------------------------------------------------------------------------  
 PART 3.2: Clean baseline data  
 REQUIRES: ${bl\_dt}/Raw Deidentified/D4DI\_baseline\_raw\_deidentified.dta  
 CREATES: ${bl\_dt}/Final/D4DI\_baseline\_clean.dta  
 ${bl\_doc}/Codebook baseline.xlsx  
 IDS VAR: hhid  
----------------------------------------------------------------------------- \*/  
 \*Change the 0 to 1 to run the baseline cleaning dofile  
 if (0) do "${bl\_do}/Cleaning/cleaning.do"  
  
/\*-----------------------------------------------------------------------------  
 PART 3.3: Construct income indicators  
 REQUIRES: ${bl\_dt}/Final/D4DI\_baseline\_clean.dta  
 CREATES: ${bl\_out}/Raw/D4DI\_baseline\_income\_distribution.png  
 ${bl\_dt}/Intermediate/D4DI\_baseline\_constructed\_income.dta  
 IDS VAR: hhid  
----------------------------------------------------------------------------- \*/  
 \*Change the 0 to 1 to run the baseline variable construction dofile  
 if (0) do "${bl\_do}/Construct/construct\_income.do"

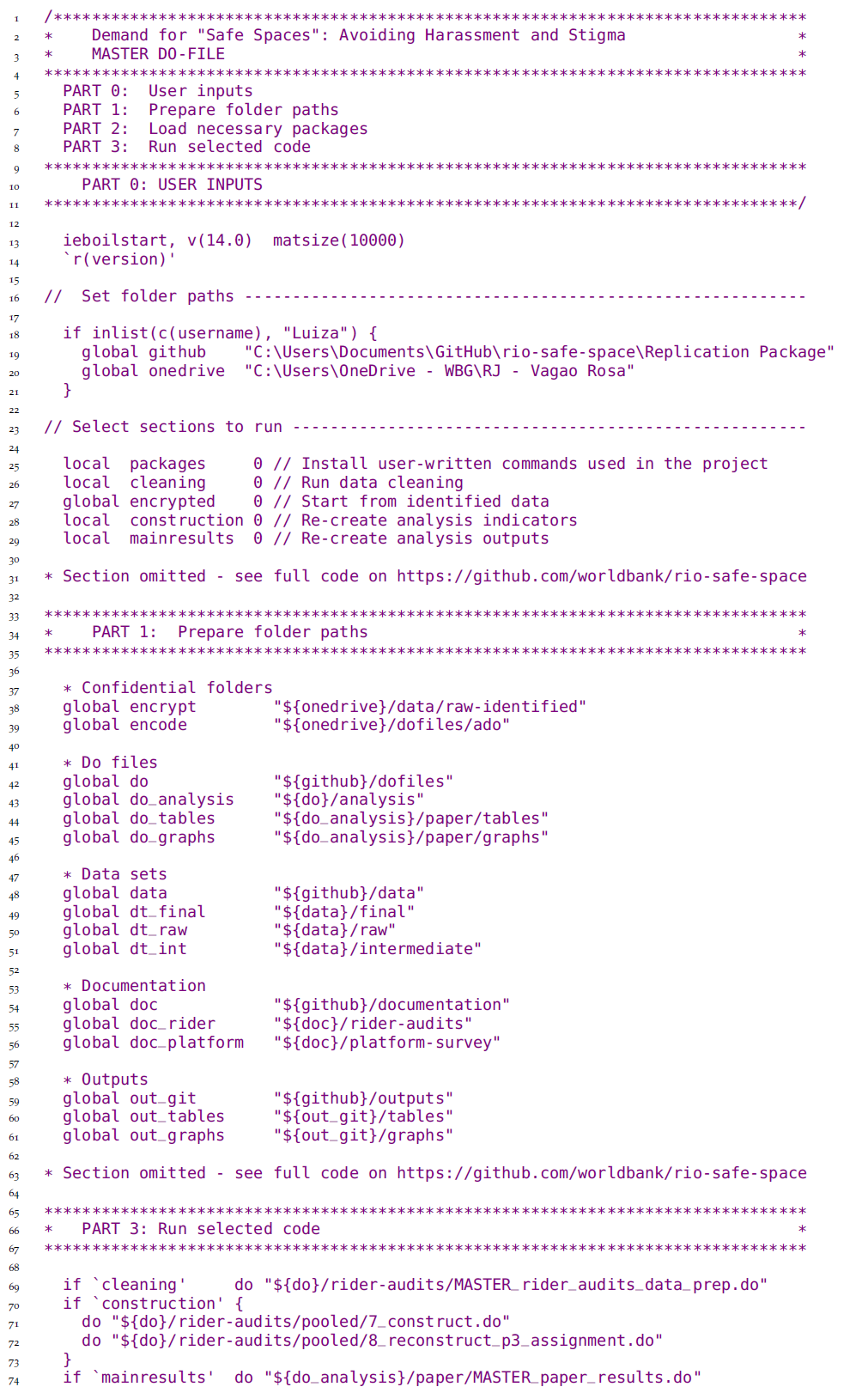
Try to create the habit of running your code from the master script. Creating “section switches” using macros or objects to run only the codes related to a certain task should always be preferred to manually open different scripts to run them in a certain order (see the if (0) switches in Part 3 of stata-master-dofile.do for one way of how to do this). Furthermore, running all scripts related to a particular task through the master whenever one of them is edited helps you identify unintended consequences of the changes you made. Say, for example, that you changed the name of a variable created in one script. This may break another script that refers to this variable. But unless you run both of them when the change is made, it may take time for that to happen, and when it does, it may take time for you to understand what’s causing an error. The same applies to changes in datasets and results.

To link code, data and outputs, the master script reflects the structure of the DataWork folder in code through globals (in Stata) or string scalars (in R); refer to lines 38-43 of the example do-file. These coding shortcuts can refer to subfolders, so that those folders can be referenced without repeatedly writing out their absolute file paths. Because the DataWork folder is shared by the whole team, its structure is the same in each team member’s computer. The only difference between machines should be the path to the project root folders, i.e. the highest-level shared folder. Depending on your software environment you may have multiple root folders. In a typical DIME project we have one Git root folder for our code, one sync software root folder for our de-identified data, and a third for our encrypted data. This is reflected in the master script in such a way that the only change necessary to run the entire code for a new team member is to change the path to the project root folders to reflect the file system and username; refer to lines 30-35 of the example do-file. The code in stata-master-dofile.do shows how folder structure is reflected in a master do-file. Because writing and maintaining a master script can be challenging as a project grows, an important feature of the iefolder is to write sub-master do-files and add to them whenever new subfolders are created in the DataWork folder.

In order to maintain well-documented and organized code, you should agree with your team on a plan to review code as it is written. Reading other people’s code is the best way to improve your coding skills. And having another set of eyes on your code will make you more comfortable with the results you find. It’s normal (and common) to make mistakes as you write your code. Reading it again to organize and comment it as you prepare it to be reviewed will help you identify them. Try to have a code review scheduled frequently, every time you finish writing a piece of code, or complete a small task. If you wait for a long time to have your code reviewed, and it gets too complex, preparation and code review will require more time and work, and that is usually the reason why this step is skipped. One other important advantage of code review is that making sure that the code is running properly on other machines, and that other people can read and understand the code easily, is the easiest way to be prepared in advance for a smooth project handover or for release of the code to the general public.

### Demand for Safe Spaces Case Study: Writing Code That Others Can Run

All code for the *Demand for Safe Spaces* study was organized to run from two master scripts, one Stata Master and one R Master. The master scripts were written such that any team member could run all project code by simply changing the top-level directory. Below is a snippet of the Stata Master:



The complete Stata master script can be found at <https://git.io/JtgeT>, and the R master at <https://git.io/JtgeY>.

## Preparing to handle confidential data ethically

Anytime you are working with original data in a development research project, you are almost certainly handling data that include **personally-identifying information (PII)**.[[101]](#footnote-278) PII is information which can, without any transformation or linkage, be used to identify individual people, households, firms, (or other units) in your data. Some examples of PII variables include names, addresses, and geolocations, email addresses, phone numbers, and bank accounts or other financial details. If you are working in a context or population that is either small, specific, or has extensive linkable data sources available to others, information like someone’s age and gender may be sufficient to disclose their identify, even though those variables would not be considered PII in general.

In a collaborative project, you will sometimes need to transfer and work with PII information, and sometimes you will prefer to remove or mask PII before transferring or working with data. There is no one-size-fits-all solution to determine what is PII, research teams have to use careful judgment in each case to avoid statistical disclosure.[[102]](#footnote-279) It is important to keep in mind that data privacy principles apply not only for the respondent giving you the information but also for their household members or other individuals who are included in the data.

In all cases where confidential information is involved, you must make sure that you adhere to several core principles. These include ethical approval, participant consent, data security, and participant privacy.[[103]](#footnote-280) If you are a US-based researcher, you will become familiar with a set of governance standards known as “The Common Rule.”[[104]](#footnote-281) If you interact with European institutions or persons, you will also become familiar with the General Data Protection Regulation (GDPR),[[105]](#footnote-282) a set of regulations governing **data ownership** and privacy standards. No matter who you are or what exact legal requirements you face, the core principles and practices you need to consider will always be similar.

### Seeking ethical approval

Most of the field research done in development involves human subjects.[[106]](#footnote-284) As a researcher, you are asking people to trust you with personal information about themselves: where they live, how rich they are, whether they have committed or been victims of crimes, their names, their national identity numbers, and all sorts of other data. PII data carries strict expectations about data storage and handling, and it is the responsibility of the research team to satisfy these expectations.[[107]](#footnote-285) Your donor or employer will most likely require you to hold a certification from a respected source.

For almost all such data collection and research activities, you will be required to complete some form of **Institutional Review Board (IRB)** process.[[108]](#footnote-286) Most commonly this consists of a formal application for approval of a specific protocol for consent, data collection, and data handling.[[109]](#footnote-287) Which IRB has authority over your project is not always apparent, particularly if some institutions do not have their own. It is customary to obtain an approval from a university IRB where at least one PI is affiliated, and if work is being done in an international setting, approval is often also required from an appropriate local institution subject to the laws of the country where data originates.

IRB approval should be obtained well before any data is acquired. IRBs may have infrequent meeting schedules or require several rounds of review for an application to be approved. If there are any deviations from an approved plan or expected adjustments, report these as early as possible so that you can update or revise the protocol. IRBs have the authority to retroactively deny the right to use data which was not acquired in accordance with an approved plan. This is extremely rare, but shows the seriousness of these considerations since the institution itself may face legal penalties if its IRB is unable to enforce them. As always, as long as you work in good faith, you should not have any issues complying with these regulations.

### Demand for Safe Spaces Example – Seeking Ethical Approval

The Duke University IRB reviewed and approved the protocol for all components of fieldwork for the *Demand for Safe Spaces* study (IRB number D0190). As one of the PIs was at Duke, and the World Bank does not have an IRB, Duke was the relevant institution in this case. The study was registered with the Duke IRB on September 2015. It was amended 4 times to reflect additions to the study design, such as an implicit association test (IAT) and a new survey. The IRB approval was renewed twice, in 2016 and 2017. Highlights from the study IRB protocols:

* *Voluntary study participation:* The study intervention was done through a smartphone application. Through the app, users were offered payment to complete a set of tasks while using the metro. The tasks involved answering questions at different moments of the trip (before boarding the train, during the ride, after leaving the train). At the start of each task, participants could review what it comprised and the exact payment value, then decide whether to accept the task or not. There was no obligation to complete any task.
* *Survey instruments:* translated drafts of all survey instruments were shared with the IRB
* *Privacy protection:* the intervention was done specialized mobile application developed by a partner technology company, which recruited users through social media. User data from the app was encrypted and stored in AWS. As per the user agreement, access to the raw data was restricted to employees of the tech company, using VPN and encrypted laptops. The tech company processed the raw data and released non-identifying data to the researchers, plus household coordinates (study participants provided informed consent to share this data).
* *Risk:* Participants were tasked with riding the public transport system in Rio De Janeiro. There is some general risk inherent in travel around Rio de Janeiro. However, the public transport system is widely used, and participants are expected to be those who regularly travel on the train. The assigned task may cause them to travel on a different route or at a different time than usual. Tasks are assigned around rush hour, so stations and trains will be crowded, which is expected to reduce risks. Note that half the riders on the system are women, and only a small fraction of the cars are reserved for women, so the task of riding the regular carriage will not require users to go into an all-male environment.
* *Ethical obligation:* when completing an assigned task, participants were asked whether they experienced any harassment. If harassment was reported, the app directed the participant to the platform guards to whom she could report harassment incidences (the guards are trained to respond to harassment reports), as well as to other resources available in the Rio area.

Appendix C in the working paper discusses the ethics aspects of the study, including participant recruitment, informed consent and how reports of harassment were addressed: <https://openknowledge.worldbank.org/handle/10986/33853>

### Obtaining informed consent

One primary consideration of IRBs is the protection of the people about whom information is being collected and whose lives may be affected by the research design. Some jurisdictions (especially those who governed by EU law) view all personal data as intrinsically owned by the persons who they describe. This means that those persons have the right to refuse to participate in data collection before it happens, as it is happening, or after it has already happened. It also means that they must explicitly and affirmatively consent to the collection, storage, and use of their information for any purpose.

The development of appropriate consent processes is of primary importance. All survey instruments must include a module in which the sampled respondent grants informed consent to participate. Research participants must be informed of the purpose of the research, what their participation will entail in terms of duration and any procedures, any foreseeable benefits or risks, and how their identity will be protected.[[110]](#footnote-290) There are special additional protections in place for vulnerable populations, such as minors, prisoners, and people with disabilities, and these should be confirmed with relevant authorities if your research includes them.

### Demand for Safe Spaces Case Study: Obtaining Informed Consent

Participation in both the study intervention (assignments to take a specific Supervia ride) and the platform survey were fully voluntary, and both included informed consent. Per the informed consent protocols for the study intervention, participation in each assigned task was voluntary, and participants were paid for each ride they completed shortly after completion, regardless of the total number of rides they completed. Thus, participants could choose to stop participating at any time if they felt uncomfortable.

The consent statement participants were shown was the following:

“If you choose to do this task, you will have to go to a Supervia station, ride the train, and answer questions about your experience on the train. You will have to ride the train for \_\_\_ minutes, and answering the questions will take about ten minutes. You will be paid at least \_\_\_\_ reais for the task, and possibly more. You will be able to review the payment for each task and option before deciding whether to do that task. You can choose during the task whether to ride either the women’s-only or the mixed carriage on the Supervia. Your responses to the task will not be identified in any way with you personally. The anonymous data will be shared with researchers at Duke University in the United States. You can choose to stop the task at any time. To get paid for this task, however, you have to finish the task.”

The sentence in italics was removed for the portion of the data collection in which participants were assigned to ride in a particular type of car; the rest of the consent statement applied to all tasks.

### Ensuring research subject privacy

In order to safeguard PII data and protect respondent privacy, you must set up a data protection protocol from the outset of a project. Secure data storage and transfer are ultimately your personal responsibility.[[111]](#footnote-293) Later chapters will discuss how to properly protect your data depending on which method you are using to acquire, store or share data. This section will only cover the computer setup you will need for any project. There are several components to this.

First, you need a system for managing strong and unique passwords for all accounts – including personal accounts like computer logins and email. This means that all your passwords should be long, not use common words and should not be reused for multiple accounts. The only way to make that practically feasible is to use a password manager.[[112]](#footnote-294) Most password managers also allow you to securely share passwords for shared accounts with your colleagues. Multi-factor authentication (sometimes called 2-step verification) is a secure alternative to passwords when available.

Second, machines that stores confidential data should not be connected to insecure physical or wireless networks, and when it is necessary to do so, they should use a VPN[[113]](#footnote-295) to connect to the internet. Furthermore, USB drives and other devices connecting over USB cables should not be connected to machines storing confidential data unless you know where the USB device came from and who has used it before you. Machines with confidential data should also be stored in a secure location when not in use.

Third, all confidential data must be encrypted at all times.[[114]](#footnote-296) When files are properly encrypted, the information they contain will be completely unreadable and unusable even if they were to be intercepted by a malicious “intruder” (an information-security term for any unauthorized information access) or accidentally made public. We will discuss implementations of encryption in more detail specific to different stages of data work in Chapter 4, but when setting up a software environment for you and your team, make sure that you have a solution for all parts of your workflow on all team members computers. You can encrypt your data at the disk (hard drive) level, called **full disk encryption** (FDE), or at the individual file or folder level, called **file system encryption** (FSE). When possible both should be applied, but our recommendation is that project teams set up data protection protocols using file system encryption as the main type of protection, and require all members that handle confidential data to use it.

Most implementations of FDE use your computer login password to prevent your files from ever being read by anyone but you. It is important to note that password protection alone is not sufficient. Password protection makes it more difficult for someone else to gain access to your files, but only encryption properly protects your data if someone manages to access your files anyway. Whenever FDE is available in your operating system it should always be enabled. FDE protects your data when encrypted just as well as FSE, but we recommend FSE due to the following disadvantages of FDE. First, FDE is implemented differently in different operating systems making it difficult to create useful instructions for all computer set-ups team members may use. Second, when FDE decrypts your disk, then all data on that disk is decrypted, even files you do not need to use at that time. Since most FDE systems automatically decrypt your full disk each time you log in, a malicious intruder that has gained access to your computer would have access to all your files. Third and perhaps most important, FDE cannot be used when sharing confidential data over insecure channels like file syncing services or email. So FDE would anyways have to be complimented with some other type of protection during collaboration.

When using FSE, instead of encrypting a full disk or drive, you create encrypted folders in which you can securely store confidential data. These encrypted folders protect your data when they are stored on your computer but can also be used to securely transfer data over insecure channels like file syncing services and email. Encrypted folders in FSE are only decrypted when you need that specific folder to be decrypted. That means that a malicious intruder that has gained access to your computer only gains access to the folders you have decrypted while your computer was compromised. Folders you rarely or never decrypt therefore remain protected even if someone gains access to your computer. DIME uses VeraCrypt for FSE, and our protocols are available as part of the DIME Research Standards.[[115]](#footnote-297) VeraCrypt is free of charge and available for Windows, MacOS, and Linux. While some details are different across platforms, the encryption will be implemented the same way for all team members.

Regardless of whether you use full disk encryption or file system encryption, it is important to remember that encryption provides no protection when your data is decrypted. Therefore, you should always log out from your computer when you are not using it, and only keep folders with confidential data decrypted when you need exactly those files. The latter is only possible when you are using FSE.

Handling confidential data properly will always add to your workload. The easiest way to reduce that work load is to handle it as rarely as possible. Whenever our research design allows us to, we should not work with confidential data at all, and not collect it or ask for it in the first place. Even when we do need confidential data for some aspect of our research, it is almost never the case that we need it for *all* aspects of our data. It is often very simple to conduct planning and analytical work using a subset of the data that does not include this type of information. Therefore we recommend that all projects plan a workflow with a version of the data where confidential data has been removed and always use that dataset when possible.

Note that it is in practice impossible to **anonymize** data. There is always some statistical chance that an individual’s identity will be re-linked to the data collected about them – even if that data has had all directly identifying information removed – by using some other data that becomes identifying when analyzed together. For this reason, we recommend de-identification in two stages. The **initial de-identification** process strips the data of direct identifiers as early in the process as possible, to create a working de-identified dataset that can be shared *within the research team* without the need for encryption. This data set should always be used when possible. The **final de-identification** process involves making a decision about the trade-off between risk of disclosure and utility of the data before publicly releasing a dataset.[[116]](#footnote-298)

Finally, it is essential to have an end-of-life plan for data even before it is acquired.[[117]](#footnote-299) This includes plans for how to transfer access and control to a new person joining the team, and how to revoke that access when someone is leaving the team. It should also include a plan for how the confidential data should be deleted. Every project should have a clear data retention and destruction plan. After a project is completed and its de-identified data has been made available as a part of data publication, research teams should not retain confidential data indefinitely.

### Demand for Safe Spaces Example: Ensuring Research Subject Privacy

The *Demand for Safe Spaces* team adopted the following data security protocols:

* All confidential data was stored in a World Bank OneDrive folder The World Bank One Drive has been set up by WB IT to be more secure than regular OneDrive and is the recommended institutional solution for storing confidential data.
* Access to the confidential data was limited to the Research Analyst and the Research Assistant working on the data cleaning.
* All de-identified data used for the analysis was stored in the synchronized folder shared by the full research team (in this case, using Dropbox).
* Indirect identifiers such as demographic variables and labels to train lines and stations were removed from the data before it was published to the Microdata Catalog.

## Looking ahead

With your code environment established, you will have a firm idea about how you are going to handle the data and code that you receive and create throughout the research process. This structure should prepare you to work collaboratively, to share code and data across machines and among team members, and to document your work as a group. With an organization plan and plans to to version-control and back up files, you are ready to handle materials ethically and securely. You should also have secured the approvals needed for any planned work. You are now ready to translate your project’s research design into a measurement framework to answer your research questions. In the next chapter, we will outline how to prepare the essential elements of research data. You will learn how to map out your project’s data needs according to both the research design and the planned creation and use of data across the project timeline.

# Bibliography

Angrist, Joshua D, and Jörn-Steffen Pischke. 2010. “The Credibility Revolution in Empirical Economics: How Better Research Design Is Taking the Con Out of Econometrics.” *Journal of Economic Perspectives* 24 (2): 3–30.

Baldwin, Kate, Shylock Muyengwa, and Eric Mvukiyehe. 2017. *Reforming Village-Level Governance via Horizontal Pressure: Evidence from an Experiment in Zimbabwe*. The World Bank.

Baldwin, Kate, and Eric Mvukiyehe. 2015. “Elections and Collective Action: Evidence from Changes in Traditional Institutions in Liberia.” *World Politics* 67 (4): 690–725.

Bedoya, Guadalupe, Aidan Coville, Johannes Haushofer, Mohammad Razaq Isaqzadeh, and Jeremy Shapiro. 2019. *No Household Left Behind: Afghanistan Targeting the Ultra Poor Impact Evaluation*. The World Bank.

Bierer, Barbara E, Mark Barnes, and Holly Fernandez Lynch. 2017. “Revised ‘Common Rule’ Shapes Protections for Research Participants.”

Camerer, Colin F, Anna Dreber, Eskil Forsell, Teck-Hua Ho, Jürgen Huber, Magnus Johannesson, Michael Kirchler, et al. 2016. “Evaluating Replicability of Laboratory Experiments in Economics.” *Science* 351 (6280): 1433–6.

Chang, Andrew C, and Phillip Li. 2015. “Is Economics Research Replicable? Sixty Published Papers from Thirteen Journals Say ’Usually Not’.” *Available at SSRN 2669564*.

Christensen, Garret, and Edward Miguel. 2018. “Transparency, Reproducibility, and the Credibility of Economics Research.” *Journal of Economic Literature* 56 (3): 920–80.

Coville, Aidan, Vincenzo Di Maro, Felipe Alexander Dunsch, and Siegfried Zottel. 2019. *The Nollywood Nudge: An Entertaining Approach to Saving*. The World Bank.

Cusolito, Ana Paula, Ernest Dautovic, and David McKenzie. 2018. *Can Government Intervention Make Firms More Investment-Ready? A Randomized Experiment in the Western Balkans*. The World Bank.

Duflo, Esther, Abhijit Banerjee, Amy Finkelstein, Lawrence F Katz, Benjamin A Olken, and Anja Sautmann. 2020. “In Praise of Moderation: Suggestions for the Scope and Use of Pre-Analysis Plans for RCTs in Economics.” National Bureau of Economic Research.

Duvendack, Maren, Richard Palmer-Jones, and W Robert Reed. 2017. “What Is Meant by ‘Replication’ and Why Does It Encounter Resistance in Economics?” *American Economic Review* 107 (5): 46–51.

Gelman, Andrew, and Eric Loken. 2013. “The Garden of Forking Paths: Why Multiple Comparisons Can Be a Problem, Even When There Is No ‘Fishing Expedition’ or ‘P-Hacking’ and the Research Hypothesis Was Posited Ahead of Time.” *Department of Statistics, Columbia University*.

Hamermesh, Daniel S. 2007. “Replication in Economics.” *Canadian Journal of Economics/Revue Canadienne d’économique* 40 (3): 715–33.

Ioannidis, John PA. 2005. “Why Most Published Research Findings Are False.” *PLoS Medicine* 2 (8): e124.

Ioannidis, John PA, Tom D Stanley, and Hristos Doucouliagos. 2017. “The Power of Bias in Economics Research.” *The Economic Journal*.

Kerr, Norbert L. 1998. “HARKing: Hypothesizing After the Results Are Known.” *Personality and Social Psychology Review* 2 (3): 196–217.

Legovini, Arianna, Guigonan Serge Adjognon, Guadalupe Bedoya Arguelles, Theophile Bougna Lonla, Kayleigh Bierman Campbell, Paul J. Christian, Aidan Coville, et al. 2019. “Science for Impact: Better Evidence for Better Decisions -– the DIME Experience.” World Bank Group. <http://documents.worldbank.org/curated/en/942491550779087507/Science-for-Impact-Better-Evidence-for-Better-Decisions-The-Dime-Experience>.

Legovini, Arianna, Vincenzo Di Maro, and Caio Piza. 2015. *Impact Evaluation Helps Deliver Development Projects*. The World Bank.

McCullough, Bruce D, Kerry Anne McGeary, and Teresa D Harrison. 2008. “Do Economics Journal Archives Promote Replicable Research?” *Canadian Journal of Economics/Revue Canadienne d’économique* 41 (4): 1406–20.

Nosek, Brian A, George Alter, George C Banks, Denny Borsboom, Sara D Bowman, Steven J Breckler, Stuart Buck, et al. 2015. “Promoting an Open Research Culture.” *Science* 348 (6242): 1422–5.

Nosek, Brian A, Charles R Ebersole, Alexander C DeHaven, and David T Mellor. 2018. “The Preregistration Revolution.” *Proceedings of the National Academy of Sciences* 115 (11): 2600–2606.

Olken, Benjamin A. 2015. “Promises and Perils of Pre-Analysis Plans.” *Journal of Economic Perspectives* 29 (3): 61–80.

Ozier, Owen. 2019. *Replication Redux: The Reproducibility Crisis and the Case of Deworming*. The World Bank.

Simmons, Joseph P, Leif D Nelson, and Uri Simonsohn. 2011. “False-Positive Psychology: Undisclosed Flexibility in Data Collection and Analysis Allows Presenting Anything as Significant.” *Psychological Science* 22 (11): 1359–66.

Simonsohn, Uri, Leif D Nelson, and Joseph P Simmons. 2014. “P-Curve: A Key to the File-Drawer.” *Journal of Experimental Psychology: General* 143 (2): 534.

Simonsohn, Uri, Joseph P Simmons, and Leif D Nelson. 2015. “Specification Curve: Descriptive and Inferential Statistics on All Reasonable Specifications.” *Available at SSRN 2694998*.

StataCorp, LLC. 2019. “Stata Statistical Software: Release 16. College Station, TX.”

Stodden, Victoria, Peixuan Guo, and Zhaokun Ma. 2013. “Toward Reproducible Computational Research: An Empirical Analysis of Data and Code Policy Adoption by Journals.” *PLoS One* 8 (6): e67111.

Swanson, Nicholas, Garret Christensen, Rebecca Littman, David Birke, Edward Miguel, Elizabeth Levy Paluck, and Zenan Wang. 2020. “Research Transparency Is on the Rise in Economics.” In *AEA Papers and Proceedings*, 110:61–65.

Vilhuber, Lars. 2020. “Implementing Increased Transparency and Reproducibility in Economics.” Zenodo. <https://doi.org/10.5281/zenodo.3911311>.

Vilhuber, Lars, James Turrito, and Keesler Welch. 2020. “Report by the AEA Data Editor.” In *AEA Papers and Proceedings*, 110:764–75.

Wicherts, Jelte M, Coosje LS Veldkamp, Hilde EM Augusteijn, Marjan Bakker, Robbie Van Aert, and Marcel ALM Van Assen. 2016. “Degrees of Freedom in Planning, Running, Analyzing, and Reporting Psychological Studies: A Checklist to Avoid P-Hacking.” *Frontiers in Psychology* 7: 1832.

Yishay, Ariel Ben, Maria Jones, Florence Kondylis, and Ahmed Mushfiq. 2016. *Are Gender Differences in Performance Innate or Socially Mediated?* The World Bank.

1. Swanson et al. ([2020](#ref-swanson2020research)) [↑](#footnote-ref-37)
2. Vilhuber ([2020](#ref-vilhuber_lars_2020_3911311)) [↑](#footnote-ref-38)
3. McCullough, McGeary, and Harrison ([2008](#ref-mccullough2008economics)) [↑](#footnote-ref-39)
4. <https://blogs.worldbank.org/impactevaluations/more-replication-economics> [↑](#footnote-ref-40)
5. See Baldwin and Mvukiyehe ([2015](#ref-baldwin2015elections)) for an example. [↑](#footnote-ref-42)
6. Camerer et al. ([2016](#ref-camerer2016evaluating)) [↑](#footnote-ref-43)
7. **Process standardization:** Agreement within a research team about how all tasks of a specific type will be approached. [↑](#footnote-ref-44)
8. Hamermesh ([2007](#ref-hamermesh2007replication)) [↑](#footnote-ref-46)
9. Ozier ([2019](#ref-ozier2019replication)) [↑](#footnote-ref-47)
10. Chang and Li ([2015](#ref-chang2015economics)) [↑](#footnote-ref-48)
11. StataCorp, LLC ([2019](#ref-statacorp2019stata)) [↑](#footnote-ref-49)
12. See DIME Analytics Coding Standards: <https://github.com/worldbank/dime-standards> [↑](#footnote-ref-50)
13. <https://www.worldbank.org/en/research/dime> [↑](#footnote-ref-53)
14. Legovini, Di Maro, and Piza ([2015](#ref-legovini2015impact)) [↑](#footnote-ref-55)
15. Legovini et al. ([2019](#ref-legovini2019)) [↑](#footnote-ref-56)
16. Angrist and Pischke ([2010](#ref-angrist2010credibility)), Ioannidis ([2005](#ref-ioannidis2005most)) [↑](#footnote-ref-69)
17. Ioannidis, Stanley, and Doucouliagos ([2017](#ref-ioannidis2017power)) [↑](#footnote-ref-70)
18. Simmons, Nelson, and Simonsohn ([2011](#ref-simmons2011false)) [↑](#footnote-ref-71)
19. Wicherts et al. ([2016](#ref-wicherts2016degrees)) [↑](#footnote-ref-72)
20. Vilhuber, Turrito, and Welch ([2020](#ref-vilhuber2020report)) [↑](#footnote-ref-74)
21. More details on study registrations and links to additional resources can be found on the DIME Wiki: <https://dimewiki.worldbank.org/Study_Registration> [↑](#footnote-ref-75)
22. Nosek et al. ([2018](#ref-nosek2018preregistration)) [↑](#footnote-ref-80)
23. More details on how to pre-register your study and links to other resources can be found on the DIME Wiki: <https://dimewiki.worldbank.org/Pre-Registration> [↑](#footnote-ref-81)
24. Kerr ([1998](#ref-kerr1998harking)) [↑](#footnote-ref-86)
25. We recommend this checklist: <https://blogs.worldbank.org/impactevaluations/a-pre-analysis-plan-checklist> [↑](#footnote-ref-87)
26. Gelman and Loken ([2013](#ref-gelman2013garden)) [↑](#footnote-ref-89)
27. More details on how to prepare a pre-analysis plans and links to additional resources can be found on the DIME Wiki: <https://dimewiki.worldbank.org/Pre-Analysis_Plan> [↑](#footnote-ref-90)
28. See Cusolito, Dautovic, and McKenzie ([2018](#ref-cusolito2018can)) for an example. [↑](#footnote-ref-91)
29. Olken ([2015](#ref-olken2015promises)) [↑](#footnote-ref-92)
30. <https://blogs.worldbank.org/impactevaluations/pre-analysis-plans-and-registered-reports-what-new-opinion-piece-does-and-doesnt> [↑](#footnote-ref-93)
31. See Bedoya et al. ([2019](#ref-bedoya2019no)) for an example. [↑](#footnote-ref-95)
32. Duflo et al. ([2020](#ref-duflo2020praise)) [↑](#footnote-ref-96)
33. More details on registered reports and links to additional resources can be found on the DIME Wiki: <https://dimewiki.worldbank.org/Registered_Reports> [↑](#footnote-ref-100)
34. <https://blogs.worldbank.org/impactevaluations/registered-reports-piloting-pre-results-review-process-journal-development-economics> [↑](#footnote-ref-102)
35. Simonsohn, Nelson, and Simmons ([2014](#ref-simonsohn2014p)) [↑](#footnote-ref-104)
36. See Coville et al. ([2019](#ref-coville2019nollywood)) for an example. [↑](#footnote-ref-105)
37. More details on research documentation and links to additional resources can be found on the DIME Wiki: <https://dimewiki.worldbank.org/Research_Documentation> [↑](#footnote-ref-107)
38. Duvendack, Palmer-Jones, and Reed ([2017](#ref-duvendack2017meant)) [↑](#footnote-ref-109)
39. **Archival repository:** A third-party service for information storage that guarantees the permanent availability of current and prior versions of materials. [↑](#footnote-ref-110)
40. <https://osf.io> [↑](#footnote-ref-112)
41. <https://github.com> [↑](#footnote-ref-114)
42. More details on how to use Git and GitHub and links to all DIME Analytics resources on best practices and how to get started can be found on the DIME Wiki: <https://dimewiki.worldbank.org/Getting_started_with_GitHub> [↑](#footnote-ref-116)
43. See Yishay et al. ([2016](#ref-yishay2016gender)) for an example. [↑](#footnote-ref-119)
44. **Original data:** A new dataset, as obtained and corrected, that becomes the functional basis for research work. [↑](#footnote-ref-120)
45. Vilhuber, Turrito, and Welch ([2020](#ref-vilhuber2020report)) [↑](#footnote-ref-121)
46. For example, <https://data.usaid.gov> [↑](#footnote-ref-122)
47. Details on how to document this type of material can be found at <https://doi.org/10.5281/zenodo.4319999>. [↑](#footnote-ref-124)
48. <https://microdata.worldbank.org> [↑](#footnote-ref-126)
49. <https:/datacatalog.worldbank.org> [↑](#footnote-ref-128)
50. **Digital object identifier (DOI):** A permanent reference for electronic information that persistently updates to a new URL or other locations if the information is relocated. [↑](#footnote-ref-130)
51. Christensen and Miguel ([2018](#ref-christensen2018transparency)) [↑](#footnote-ref-132)
52. More details and links to additional resources on how to make your research reproducible and prepare a reproducibility package can be found on the DIME Wiki: <https://dimewiki.worldbank.org/Reproducible_Research>. More details can also be found under Pillar 3 in the DIME Research Standards: <https://github.com/worldbank/dime-standards>. [↑](#footnote-ref-133)
53. More details and links to best practices on topics related to data publication, such as de-identification and how to license published data, can be found on the DIME Wiki: <https://dimewiki.worldbank.org/Publishing_Data>. More details can also be found under Pillar 5 in the DIME Research Standards: <https://github.com/worldbank/dime-standards> [↑](#footnote-ref-134)
54. Stodden, Guo, and Ma ([2013](#ref-stodden2013toward)) [↑](#footnote-ref-135)
55. The DIME Analytics reproducibility checklist can be found in Pillar 3 of the DIME Research Standards at <https://github.com/worldbank/dime-standards>. [↑](#footnote-ref-136)
56. Nosek et al. ([2015](#ref-nosek2015promoting)) [↑](#footnote-ref-137)
57. **Computational reproducibility:** The ability of another individual to reuse the same code and data and obtain the exact same results as yours. [↑](#footnote-ref-139)
58. More details and links to additional resources on how to make your research reproducible and prepare a reproducibility package can be found on the DIME Wiki: <https://dimewiki.worldbank.org/Reproducible_Research>. More details can also be found under Pillar 3 in the DIME Research Standards: <https://github.com/worldbank/dime-standards>. [↑](#footnote-ref-140)
59. <https://blogs.worldbank.org/impactevaluations/what-development-economists-talk-about-when-they-talk-about-reproducibility> [↑](#footnote-ref-141)
60. Simonsohn, Simmons, and Nelson ([2015](#ref-simonsohn2015specification)) [↑](#footnote-ref-143)
61. <https://blogs.worldbank.org/opendata/making-analytics-reusable> [↑](#footnote-ref-144)
62. Read more details about back up strategies and other aspects of data storage on the DIME Wiki: <https://dimewiki.worldbank.org/Data_Storage>. [↑](#footnote-ref-158)
63. More naming conventions and links to more resources can be found on the DIME Wiki: <https://dimewiki.worldbank.org/Naming_Conventions>. [↑](#footnote-ref-160)
64. **Git:** A distributed version control system for collaborating on and tracking changes to code as it is written. [↑](#footnote-ref-162)
65. Read more about how to install and use ieboilstart and how it can help you harmonize settings across users as much as possible in Stata on the DIME Wiki: <https://dimewiki.worldbank.org/ieboilstart>. [↑](#footnote-ref-165)
66. <https://www.rstudio.com> [↑](#footnote-ref-167)
67. More details on the benefits of external code editors and links to popular code editors can be found on the DIME Wiki: <https://dimewiki.worldbank.org/Code_Editors>. [↑](#footnote-ref-169)
68. Read more about how to install and use iefolder, as well as find links to resources explaining the best practices this command build upon on the DIME Wiki: <https://dimewiki.worldbank.org/iefolder>. [↑](#footnote-ref-173)
69. More details on DIME’s suggested data work folder structure and explanations of the best practices it is based on can be found on the DIME Wiki: <https://dimewiki.worldbank.org/DataWork_Folder>. [↑](#footnote-ref-175)
70. Git only tracks files, so empty folders – which most folders are in the beginning of a project – are ignored if placeholder files are not used, leading to only parts of the folder structure being shared across the team. Read more about how to install and use iegitaddmd and how it can help you track empty folder in Git on the DIME Wiki: <https://dimewiki.worldbank.org/Iegitaddmd>. [↑](#footnote-ref-177)
71. **Comments:** Code components that have no function to the computer, but describe in plain language for humans to read what the code is supposed to do. [↑](#footnote-ref-184)
72. More details and description of each section of our template master do-file can be found on the DIME Wiki: <https://dimewiki.worldbank.org/Master_Do-files>. [↑](#footnote-ref-189)
73. **Personally-identifying information:** any piece or set of information that can be used to identify an individual research subject. Read more about what extra consideration you must take into account when working with PII data on the DIME Wiki: <https://dimewiki.worldbank.org/Protecting_Human_Research_Subjects>. [↑](#footnote-ref-196)
74. More details on statistical disclosure and links to best practices related to data publication, can be found on the DIME Wiki: <https://dimewiki.worldbank.org/Publishing_Data>. [↑](#footnote-ref-198)
75. See Baldwin, Muyengwa, and Mvukiyehe ([2017](#ref-baldwin2017reforming)) for an example. [↑](#footnote-ref-199)
76. Bierer, Barnes, and Lynch ([2017](#ref-bierer2017revised)) [↑](#footnote-ref-200)
77. <https://gdpr-info.eu> [↑](#footnote-ref-201)
78. Read more about what extra consideration you must take into account when working with human subjects on the DIME Wiki: <https://dimewiki.worldbank.org/Protecting_Human_Research_Subjects> [↑](#footnote-ref-204)
79. More details on research ethics as well as links to tools and other resources related can be found on the DIME Wiki: <https://dimewiki.worldbank.org/Research_Ethics>. More details can also be found under Pillar 1 in the DIME Research Standards: <https://github.com/worldbank/dime-standards> [↑](#footnote-ref-205)
80. **Institutional Review Board (IRB):** An institution formally responsible for ensuring that research meets ethical standards. [↑](#footnote-ref-207)
81. More details and best practices for how to submit a project for an IRB approval can be found on the DIME Wiki: <https://dimewiki.worldbank.org/IRB_Approval>. [↑](#footnote-ref-208)
82. More details on best practices when obtaining informed consent and links to additional resources can be found on the DIME Wiki: <https://dimewiki.worldbank.org/Informed_Consent>. [↑](#footnote-ref-212)
83. Read more about data security and the options you have to protect your data either on the DIME Wiki: <https://dimewiki.worldbank.org/Data_Security>, or under Pillar 4 in the DIME Research Standards <https://github.com/worldbank/dime-standards>. [↑](#footnote-ref-216)
84. Read our step-by-step guide for how to get started with password managers under Pillar 4 in the DIME Research Standards: <https://github.com/worldbank/dime-standards>. [↑](#footnote-ref-218)
85. **Virtual Private Network (VPN):** Allows you to securely connect to a network you trust over an insecure network. With the VPN you can securely communicate with other devices on the trusted network and make your traffic to the internet inaccessible to the host of the insecure network. [↑](#footnote-ref-219)
86. **Encryption:** Methods which ensure that files are unreadable even if laptops are stolen, databases are hacked, or any other type of unauthorized access is obtained. Read more about these methods on the DIME Wiki: <https://dimewiki.worldbank.org/Encryption>. [↑](#footnote-ref-220)
87. Read our step-by-step guide for how to get started with Veracrypt under Pillar 4 in the DIME Research Standards: <https://github.com/worldbank/dime-standards>. [↑](#footnote-ref-222)
88. More details and best practices related to de-identification as well as tools that can help you assess disclosure risks can be found on the DIME Wiki: <https://dimewiki.worldbank.org/De-identification>. [↑](#footnote-ref-223)
89. Read more details about end-of-life plans for data and other aspects of data storage on the DIME Wiki: <https://dimewiki.worldbank.org/Data_Storage>. [↑](#footnote-ref-225)
90. Read more details about back up strategies and other aspects of data storage on the DIME Wiki: <https://dimewiki.worldbank.org/Data_Storage>. [↑](#footnote-ref-255)
91. More naming conventions and links to more resources can be found on the DIME Wiki: <https://dimewiki.worldbank.org/Naming_Conventions>. [↑](#footnote-ref-256)
92. **Git:** A distributed version control system for collaborating on and tracking changes to code as it is written. [↑](#footnote-ref-257)
93. Read more about how to install and use ieboilstart and how it can help you harmonize settings across users as much as possible in Stata on the DIME Wiki: <https://dimewiki.worldbank.org/ieboilstart>. [↑](#footnote-ref-260)
94. <https://www.rstudio.com> [↑](#footnote-ref-261)
95. More details on the benefits of external code editors and links to popular code editors can be found on the DIME Wiki: <https://dimewiki.worldbank.org/Code_Editors>. [↑](#footnote-ref-262)
96. Read more about how to install and use iefolder, as well as find links to resources explaining the best practices this command build upon on the DIME Wiki: <https://dimewiki.worldbank.org/iefolder>. [↑](#footnote-ref-265)
97. More details on DIME’s suggested data work folder structure and explanations of the best practices it is based on can be found on the DIME Wiki: <https://dimewiki.worldbank.org/DataWork_Folder>. [↑](#footnote-ref-266)
98. Git only tracks files, so empty folders – which most folders are in the beginning of a project – are ignored if placeholder files are not used, leading to only parts of the folder structure being shared across the team. Read more about how to install and use iegitaddmd and how it can help you track empty folder in Git on the DIME Wiki: <https://dimewiki.worldbank.org/Iegitaddmd>. [↑](#footnote-ref-267)
99. **Comments:** Code components that have no function to the computer, but describe in plain language for humans to read what the code is supposed to do. [↑](#footnote-ref-272)
100. More details and description of each section of our template master do-file can be found on the DIME Wiki: <https://dimewiki.worldbank.org/Master_Do-files>. [↑](#footnote-ref-275)
101. **Personally-identifying information:** any piece or set of information that can be used to identify an individual research subject. Read more about what extra consideration you must take into account when working with PII data on the DIME Wiki: <https://dimewiki.worldbank.org/Protecting_Human_Research_Subjects>. [↑](#footnote-ref-278)
102. More details on statistical disclosure and links to best practices related to data publication, can be found on the DIME Wiki: <https://dimewiki.worldbank.org/Publishing_Data>. [↑](#footnote-ref-279)
103. See Baldwin, Muyengwa, and Mvukiyehe ([2017](#ref-baldwin2017reforming)) for an example. [↑](#footnote-ref-280)
104. Bierer, Barnes, and Lynch ([2017](#ref-bierer2017revised)) [↑](#footnote-ref-281)
105. <https://gdpr-info.eu> [↑](#footnote-ref-282)
106. Read more about what extra consideration you must take into account when working with human subjects on the DIME Wiki: <https://dimewiki.worldbank.org/Protecting_Human_Research_Subjects> [↑](#footnote-ref-284)
107. More details on research ethics as well as links to tools and other resources related can be found on the DIME Wiki: <https://dimewiki.worldbank.org/Research_Ethics>. More details can also be found under Pillar 1 in the DIME Research Standards: <https://github.com/worldbank/dime-standards> [↑](#footnote-ref-285)
108. **Institutional Review Board (IRB):** An institution formally responsible for ensuring that research meets ethical standards. [↑](#footnote-ref-286)
109. More details and best practices for how to submit a project for an IRB approval can be found on the DIME Wiki: <https://dimewiki.worldbank.org/IRB_Approval>. [↑](#footnote-ref-287)
110. More details on best practices when obtaining informed consent and links to additional resources can be found on the DIME Wiki: <https://dimewiki.worldbank.org/Informed_Consent>. [↑](#footnote-ref-290)
111. Read more about data security and the options you have to protect your data either on the DIME Wiki: <https://dimewiki.worldbank.org/Data_Security>, or under Pillar 4 in the DIME Research Standards <https://github.com/worldbank/dime-standards>. [↑](#footnote-ref-293)
112. Read our step-by-step guide for how to get started with password managers under Pillar 4 in the DIME Research Standards: <https://github.com/worldbank/dime-standards>. [↑](#footnote-ref-294)
113. **Virtual Private Network (VPN):** Allows you to securely connect to a network you trust over an insecure network. With the VPN you can securely communicate with other devices on the trusted network and make your traffic to the internet inaccessible to the host of the insecure network. [↑](#footnote-ref-295)
114. **Encryption:** Methods which ensure that files are unreadable even if laptops are stolen, databases are hacked, or any other type of unauthorized access is obtained. Read more about these methods on the DIME Wiki: <https://dimewiki.worldbank.org/Encryption>. [↑](#footnote-ref-296)
115. Read our step-by-step guide for how to get started with Veracrypt under Pillar 4 in the DIME Research Standards: <https://github.com/worldbank/dime-standards>. [↑](#footnote-ref-297)
116. More details and best practices related to de-identification as well as tools that can help you assess disclosure risks can be found on the DIME Wiki: <https://dimewiki.worldbank.org/De-identification>. [↑](#footnote-ref-298)
117. Read more details about end-of-life plans for data and other aspects of data storage on the DIME Wiki: <https://dimewiki.worldbank.org/Data_Storage>. [↑](#footnote-ref-299)