BROOKE ANDERSON, MICHAEL LYONS, MERCEDES GONZALEZ-JUARRERO, MARCELA HENAO-TAMAYO, AND GREGORY ROBERTSON

IMPROVING THE REPRODUCIBILITY OF EXPERIMENTAL DATA RECORDING AND PRE-PROCESSING

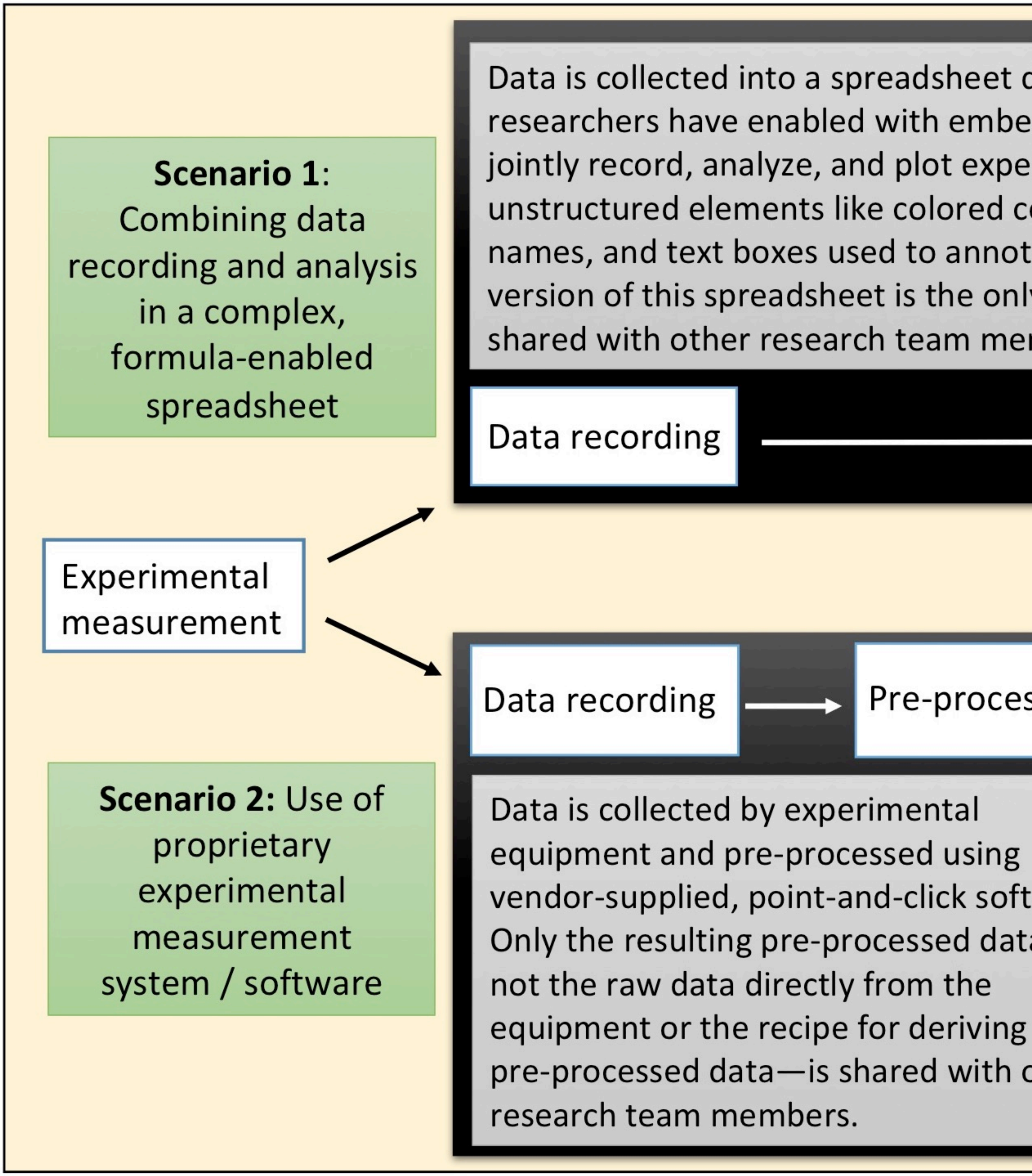
Contents

1

Overview

The recent NIH-Wide Strategic Plan (U.S. Department of Health and Human Services, National Institutes of Health, 2016) describes an integrative view of biology and human health that includes translational medicine, team science, and the importance of capitalizing on an exponentially growing and increasingly complex data ecosystem (U.S. Department of Health and Human Services, National Institutes of Health, 2018). Underlying this view is the need to use, share, and re-use biomedical data generated from widely varying experimental systems and researchers. Basic sources of biomedical data range from relatively small sets of measurements, such as animal body weights and bacterial cell counts that may be recorded by hand, to thousands or millions of instrument-generated data points from various imaging, -omic, and flow cytometry experiments. In either case, there is a generally common workflow that proceeds from measurement to data recording, pre-processing, analysis, and interpretation. However, in practice the distinct actions of data recording, data pre-processing, and data analysis are often merged or combined as a single entity by the researcher using commercial or open source spreadsheets, or as part of an often proprietary experimental measurement system / software combination (Figure 1.1), resulting in key failure points for reproducibility at the stages of data recording and pre-processing.

It is widely known and discussed among data scientists, mathematical modelers, and statisticians (Broman and Woo, 2018; Krishnan et al., 2016) that there is frequently a need to discard, transform, and reformat various elements of the data shared with them by laboratory-based researchers, and that data is often shared in an unstructured format, increasing the risks of introducing errors through reformatting before applying more advanced computational methods. Instead, a critical need for reproducibility is for the transparent and clear sharing across research teams of: (1) raw data, directly from hand-recording or directly output from experimental equipment; (2) data that has been pre-processed as necessary (e.g., gating for flow cytometry data, feature identification for metabolomics data), saved in a consistent, structured format, and (3) a clear and repeatable description of how the pre-processed data was


Scenario 1: Combining data recording and analysis in a complex, formula-enabled spreadsheet
Experimental measurement
Data recording
Scenario 2: Use of proprietary experimental measurement system / software
Data recording
Pre-proces

Data is collected into a spreadsheet c
researchers have enabled with embe
jointly record, analyze, and plot expe
unstructured elements like colored c
names, and text boxes used to annot
version of this spreadsheet is the only
shared with other research team me

Data is collected by experimental
equipment and pre-processed using
vendor-supplied, point-and-click soft
Only the resulting pre-processed dat
not the raw data directly from the
equipment or the recipe for deriving
pre-processed data—is shared with o
research team members.

Figure 1.1: Two scenarios where 'black boxes' of non-transparent, non-reproducible data handling exist in research data workflows at the stages of data recording and pre-processing. These create potential points of failure for reproducible research. Red arrows indicate where data is passed to other research team members, including statisticians / data analysts, often within complex or unstructured spreadsheet files.

generated from the raw data (Broman and Woo, 2018; Ellis and Leek, 2018).

To enhance data reproducibility, it is critical to create a clear separation among data recording, data pre-processing, and data analysis—breaking up commonly existing "black boxes" in data handling across the research process. Such a rigorous demarcation requires some change in the conventional understanding and use of spreadsheets and a recognition by biomedical researchers that recent advances in computer programming languages, especially the R programming language, provide user-friendly and accessible tools and concepts that can be used to extend a transparent and reproducible data workflow to the steps of data recording and pre-processing. Among our team, we have found that there are many common existing practices—including use of spreadsheets with embedded formulas that concurrently record and analyze experimental data, problematic management of project files, and reliance on proprietary, vendor-supplied point-and-click software for data pre-processing—that can interfere with the transparency, reproducibility, and efficiency of laboratory-based biomedical research projects, problems that have also been identified by others as key barriers to research reproducibility (Broman and Woo, 2018; Bryan, 2018; Ellis and Leek, 2018, Marwick et al. (2018)). In these training modules, we have choosen topics that tackle barriers to reproducibility that have straightforward, easy-to-teach solutions, but which are still very common in biomedical laboratory-based research programs.

1.1 License

Click on the **Next** button (or navigate using the links at the top of the page) to continue.

2
Experimental Data Recording

2.1 Separating data recording and analysis

Many biomedical laboratories use spreadsheets, with embedded formulas, to both record and analyze experimental data. This practice impedes transparency and reproducibility of both data recording and data analysis. In this module, we will describe this common practice and will outline alternative approaches that separate the steps of data recording and data analysis.

Objectives. After this module, the trainee will be able to:

- Explain the difference between data recording and data analysis
- Understand why collecting data on spreadsheets with embedded formulas impedes reproducibility
- List alternative approaches to improve reproducibility

2.1.1 Data recording versus data analysis

2.1.2 Common practices combining recording and analysis

2.1.3 Hazards of combining recording and analysis

2.1.4 Approaches to separate recording and analysis

2.1.5 Discussion questions

2.2 Principles and power of structured data formats

The format in which experimental data is recorded can have a large influence on how easy and likely it is to implement reproducibility tools in later stages of the research workflow. Recording data in a "structured" format brings many benefits. In this module, we will explain what makes a dataset "structured" and why this format is a powerful tool for reproducible research.

Objectives. After this module, the trainee will be able to:

- List the characteristics of a structured data format
- Describe benefits for research transparency and reproducibility
- Outline other benefits of using a structured format when recording data

2.2.1 Characteristics of a structured data format

2.2.2 Benefits of a structured data format

2.2.3 Applied exercise

2.3 The 'tidy' data format

The "tidy" data format is an implementation of a structured data format popular among statisticians and data scientists. By consistently using this data format, researchers can combine simple, generalizable tools to perform complex tasks in data processing, analysis, and visualization. We will explain what characteristics determine if a dataset is "tidy" and how use of the "tidy" implementation of a structure data format can improve the ease and efficiency of "Team Science".

Objectives. After this module, the trainee will be able to:

- List characteristics defining the "tidy" structured data format
- Explain the difference between the a structured data format (general concept) and the 'tidy' data format (one popular implementation)

2.3.1 The "tidy" data format

2.3.2 The "tidy" data format as a structured data format

2.3.3 Practice quiz

2.4 Designing templates for "tidy" data collection

This module will move from the principles of the "tidy" data format to the practical details of designing a "tidy" data format to use when collecting experimental data. We will describe common issues that prevent biomedical research datasets from being "tidy" and show how these issues can be avoided. We will also provide rubrics and a checklist to help determine if a data collection template complies with a "tidy" format.

Objectives. After this module, the trainee will be able to:

- Identify characteristics that keep a dataset from being 'tidy'
- Convert data from an "untidy" to a "tidy" format

2.4.1 Subsection 1

2.4.2 Applied exercise

2.5 Example: Creating a template for "tidy" data collection

We will walk through an example of creating a template to collect data in a "tidy" format for a laboratory-based research project, based on a research project on drug efficacy in murine tuberculosis models. We will show the initial "untidy" format for data recording and show how we converted it to a "tidy" format. Finally, we will show how the data can then easily be analyzed and visualized using reproducible tools.

Objectives. After this module, the trainee will be able to:

- Understand how the principles of "tidy" data can be applied for a real, complex research project;
- List advantages of the "tidy" data format for the example project

2.5.1 Subsection 1

2.5.2 Subsection 2

2.5.3 Discussion questions

2.6 Power of using a single structured 'Project' directory for storing and tracking research project files

To improve the computational reproducibility of a research project, researchers can use a single 'Project' directory to collectively store all research data, metadata, pre-processing code, and research products (e.g., paper drafts, figures). We will explain how this practice improves the reproducibility and list some of the common components and subdirectories to include in the structure of a 'Project' directory, including subdirectories for raw and pre-processed experimental data.

Objectives. After this module, the trainee will be able to:

- Describe a 'Project' directory, including common components and subdirectories
- List how a single 'Project' directory improves reproducibility

2.6.1 Subsection 1

2.6.2 Subsection 2

2.6.3 Practice quiz

2.7 Creating 'Project' templates

Researchers can use RStudio's 'Projects' can facilitate collecting research files in a single, structured directory, with the added benefit of easy use of version control. Researchers can gain even more benefits by consistently structuring all their 'Project' directories. We will demonstrate how to implement structured project directories through RStudio, as well as how RStudio enables the creation of a 'Project' for initializing consistently-structured directories for all of a research group's projects.

Objectives. After this module, the trainee will be able to:

- Be able to create a structured `Project` directory within RStudio
- Understand how RStudio can be used to create 'Project' templates

2.7.1 Subsection 1

2.7.2 Subsection 2

2.7.3 Discussion questions

2.8 Example: Creating a 'Project' template

We will walk through a real example, based on the experiences of one of our Co-Is, of establishing the format for a research group's 'Project' template, creating that template using RStudio, and initializing a new research project directory using the created template. This example will be from a laboratory-based research group that studies the efficacy of tuberculosis drugs in a murine model.

Objectives. After this module, the trainee will be able to:

- Create a 'Project' template in RStudio to initialize consistently-formatted 'Project' directories
- Initialize a new 'Project' directory using this template

2.8.1 Subsection 1

2.8.2 Subsection 2

2.8.3 Applied exercise

2.9 Harnessing version control for transparent data recording

As a research project progresses, a typical practice in many experimental research groups is to save new versions of files (e.g., 'draft1.doc', 'draft2.doc'), so that changes can be reverted. However, this practice leads to an explosion of files, and it becomes hard to track which files represent the 'current' state of a project. Version control allows researchers to edit and change research project files more cleanly, while maintaining the power to 'backtrack' to previous versions, messages included to explain changes. We will explain what version control is and how it can be used in research projects to improve the transparency and reproducibility of research, particularly for data recording.

Objectives. After this module, the trainee will be able to:

- Describe version control
- Explain how version control can be used to improve reproducibility for data recording

2.9.1 Subsection 1

2.9.2 Subsection 2

2.9.3 Discussion questions

2.10 Enhance the reproducibility of collaborative research with version control platforms

Once a researcher has learned to use *git* on their own computer for local version control, they can begin using version control platforms (e.g., *GitLab*, *GitHub*) to collaborate with others under version control. We will describe how a research team can benefit from using a version control platform to work collaboratively.

Objectives. After this module, the trainee will be able to:

- List benefits of using a version control platform to collaborate on research projects, particularly for reproducibility
- Describe the difference between version control (e.g., *git*) and a version control platform (e.g., *GitLab*)

2.10.1 Subsection 1

2.10.2 Subsection 2

2.10.3 Discussion questions

2.11 Using git and GitLab to implement version control

For many years, use of version control required use of the command line, limiting its accessibility to researchers with limited programming experience. However, graphical interfaces have removed this barrier, and RStudio has particularly user-friendly tools for implementing version control. In this module, we will show how to use *git* through RStudio's user-friendly interface and how to connect from a local computer to *GitLab* through RStudio.

Objectives. After this module, the trainee will be able to:

- Understand how to set up and use *git* through RStudio's interface
- Understand how to connect with *GitLab* through RStudio to collaborate on research projects while maintaining version control

2.11.1 Subsection 1

2.11.2 Subsection 2

2.11.3 Applied exercise

3

Experimental Data Preprocessing

3.1 Principles and benefits of scripted pre-processing of experimental data

The experimental data collected for biomedical research often requires pre-processing before it can be analyzed (e.g., gating of flow cytometry data, feature finding / quantification for mass spectrometry data). Use of point-and-click software can limit the transparency and reproducibility of this analysis stage and is time-consuming for repeated tasks. We will explain how scripted pre-processing, especially using open source software, can improve transparency and reproducibility.

Objectives. After this module, the trainee will be able to:

- Define 'pre-processing' of experimental data
- Describe an open source code script and explain how it can increase reproducibility of data pre-processing

3.1.1 Subsection 1

3.1.2 Subsection 2

3.1.3 Discussion questions

3.2 Introduction to scripted data pre-processing in R

We will show how to implement scripted pre-processing of experimental data through R scripts. We will demonstrate the difference between interactive coding and code scripts, using R for examples. We will then demonstrate how to create, save, and run an R code script for a simple data cleaning task.

Objectives. After this module, the trainee will be able to:

- Describe what an R code script is and how it differs from interactive coding in R
- Create and save an R script to perform a simple data pre-processing task
- Run an R script
- List some popular packages in R for pre-processing biomedical data

3.2.1 Subsection 1

3.2.2 Subsection 2

3.2.3 Applied exercise

3.3 Simplify scripted pre-processing through R's 'tidyverse' tools

The R programming language now includes a collection of 'tidyverse' extension packages that enable user-friendly yet powerful work with experimental data, including pre-processing and exploratory visualizations. The principle behind the 'tidyverse' is that a collection of simple, general tools can be joined together to solve complex problems, as long as a consistent format is used for the input and output of each tool (the 'tidy' data format taught in other modules). In this module, we will explain why this 'tidyverse' system is so powerful and how it can be leveraged within biomedical research, especially for reproducibly pre-processing experimental data.

Objectives. After this module, the trainee will be able to:

- Define R's 'tidyverse' system
- Explain how the 'tidyverse' collection of packages can be both user-friendly and powerful in solving many complex tasks with data
- Describe the difference between base R and R's 'tidyverse'.

3.3.1 Subsection 1

3.3.2 Subsection 2

3.3.3 Practice quiz

3.4 Complex data types in experimental data pre-processing

Raw data from many biomedical experiments, especially those that use high-throughput techniques, can be very large and complex. Because of the scale and complexity of these data, software for pre-processing the data in R often uses complex, 'untidy' data formats. While these formats are necessary for computational efficiency, they add a critical barrier for researchers wishing to implement reproducibility tools. In this module, we will explain why use of complex data formats is often necessary within open source pre-processing software and outline the hurdles created in reproducibility tool use among laboratory-based scientists.

Objectives. After this module, the trainee will be able to:

- Explain why R software for pre-processing biomedical data often stores data in complex, 'untidy' formats
- Describe how these complex data formats can create barriers to laboratory-based researchers seeking to use reproducibility tools for data pre-processing

3.4.1 Subsection 1

3.4.2 Subsection 2

3.4.3 Practice quiz

3.5 Complex data types in R and Bioconductor

Many R extension packages for pre-processing experimental data use complex (rather than 'tidy') data formats within their code, and many output data in complex formats. Very recently, the *broom* and *biobroom* R packages have been developed to extract a 'tidy' dataset from a complex data format. These tools create a clean, simple connection between the complex data formats often used in pre-processing experimental data and the 'tidy' format required to use the 'tidyverse' tools now taught in many introductory R courses. In this module, we will describe the 'list' data structure, the common backbone for complex data structures in R and provide tips on how to explore and extract data stored in R in this format, including through the *broom* and *biobroom* packages.

Objectives. After this module, the trainee will be able to:

- Describe the structure of R's 'list' data format
- Take basic steps to explore and extract data stored in R's complex, list-based structures
- Describe what the *broom* and *biobroom* R packages can do
- Explain how converting data to a 'tidy' format can improve reproducibility

3.5.1 Subsection 1

3.5.2 Subsection 2

3.5.3 Applied exercise

3.6 Example: Converting from complex to 'tidy' data formats

We will provide a detailed example of a case where data pre-processing in R results in a complex, 'untidy' data format. We will walk through an example of applying automated gating to flow cytometry data. We will demonstrate the complex initial format of this pre-processed data and then show trainees how a 'tidy' dataset can be extracted and used for further data analysis and visualization using the popular R 'tidyverse' tools. This example will use real experimental data from one of our Co-Is research on the immunology of tuberculosis.

Objectives. After this module, the trainee will be able to:

- Describe how tools like *biobroom* were used in this real research example to convert from the complex data format from pre-processing to a format better for further data analysis and visualization
- Understand how these tools would fit in their own research pipelines

3.6.1 Subsection 1

3.6.2 Subsection 2

3.6.3 Applied exercise

3.7 Introduction to reproducible data pre-processing protocols

Reproducibility tools can be used to create reproducible data pre-processing protocols—documents that combine code and text in a 'knitted' document, which can be re-used to ensure data pre-processing is consistent and reproducible across research projects. In this module, we will describe how reproducible data pre-processing protocols can improve reproducibility of pre-processing experimental data, as well as to ensure transparency, consistency, and reproducibility across the research projects conducted by a research team.

Objectives. After this module, the trainee will be able to:

- Define a 'reproducible data pre-processing protocol'
- Explain how such protocols improve reproducibility at the data pre-processing phase
- List other benefits, including improving efficiency and consistency of data pre-processing

3.7.1 Subsection 1

3.7.2 Subsection 2

3.7.3 Discussion questions

3.8 RMarkdown for creating reproducible data pre-processing protocols

The R extension package RMarkdown can be used to create documents that combine code and text in a 'knitted' document, and it has become a popular tool for improving the computational reproducibility and efficiency of the data analysis stage of research. This tool can also be used earlier in the research process, however, to improve reproducibility of pre-processing steps. In this module, we will provide detailed instructions on how to use RMarkdown in RStudio to create documents that combine code and text. We will show how an RMarkdown document describing a data pre-processing protocol can be used to efficiently apply the same data pre-processing steps to different sets of raw data.

Objectives. After this module, the trainee will be able to:

- Define RMarkdown and the documents it can create
- Explain how RMarkdown can be used to improve the reproducibility of research projects at the data pre-processing phase
- Create a document in RStudio using
- Apply it to several different datasets with the same format

3.8.1 Subsection 1

3.8.2 Subsection 2

3.8.3 Applied exercise

3.9 Example: Creating a reproducible data pre-processing protocol

We will walk through an example of creating a reproducible protocol for the automated gating of flow cytometry data for a project on the immunology of tuberculosis lead by one of our Co-Is. This data pre-processing protocol was created using RMarkdown and allows the efficient, transparent, and reproducible gating of flow cytometry data for all experiments in the research group. We will walk the trainees through how we developed the protocol initially, the final pre-processing protocol, how we apply this protocol to new experimental data.

Objectives. After this module, the trainee will be able to:

- Explain how a reproducible data pre-processing protocol can be integrated into a real research project
- Understand how to design and implement a data pre-processing protocol to replace manual or point-and-click data pre-processing tools

3.9.1 Subsection 1

3.9.2 Subsection 2

3.9.3 Practice quiz

4
Bibliography

Broman, K. W. and Woo, K. H. (2018). Data organization in spreadsheets. *The American Statistician*, 72(1):2–10.

Bryan, J. (2018). Excuse me, do you have a moment to talk about version control? *The American Statistician*, 72(1):20–27.

Ellis, S. E. and Leek, J. T. (2018). How to share data for collaboration. *The American Statistician*, 72(1):53–57.

Krishnan, S., Haas, D., Franklin, M. J., and Wu, E. (2016). Towards reliable interactive data cleaning: A user survey and recommendations. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, page 9. ACM.

Marwick, B., Boettiger, C., and Mullen, L. (2018). Packaging data analytical work reproducibly using R (and friends). *The American Statistician*, 72(1):80–88.

U.S. Department of Health and Human Services, National Institutes of Health (2016). NIH-Wide Strategic Plan, Fiscal Years 2016-2020: Turning Discovery Into Health. Accessed: 2018-06-24.

U.S. Department of Health and Human Services, National Institutes of Health (2018). NIH Strategic Plan for Data Science. Accessed: 2018-06-24.