Table 1: Modules for the first sequence, **'Improving the Reproducibility of Experimental Data Recording'**. The color of each module's title indicates whether the module focuses on **Principles** (blue), **Implementation** (red), or **Case study examples** (black). This table is continued over several pages.

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video length (min.)	Extra educational materials
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.	• 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	15	• Discussion questions about data recording approaches the trainee has previously used in research projects and the benefits and limitations for data transparency and reproducibility • Short audio recording of two Co-Is giving their answers
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.	• 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	10	• Applied exercise: For example datasets, specify whether each is in a structured data format and, if not, draft a structured version • Video walking trainees through solutions to the applied exercise
The 'tidy' data format: an implementation of a structured 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'.	• List characteristics defining the the 'tidy' structured data format • Explain the difference between the a structured data format (general concept) and the 'tidy' data format (one popular implementation)	15	• Quiz questions: For example datasets, correctly identify which of the 'tidy' data principles the dataset has or lacks • Video explaining quiz solutions
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.	• Identify characteristics that keep a dataset from being 'tidy' • Convert data from an 'untidy' to a 'tidy' format	20	• Applied exercise: For an 'untidy' dataset, explain why it is not 'tidy' and convert to a 'tidy' format • Video providing a detailed solution to the applied exercise

Table 1: Modules for the first sequence, **'Improving the Reproducibility of Experimental Data Recording'**. The color of each module's title indicates whether the module focuses on **Principles** (blue), **Implementation** (red), or **Case study examples** (black). This table is continued over several pages. *(continued)*

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video length (min.)	Extra educational materials
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.	• 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	15	• Discussion questions, including listing examples of experiences collecting data in an 'untidy' format • Short audio recording of two Co-Is giving their answers
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, meta-data, 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.	• Describe a 'Project' directory, including common components and subdirectories • List how a single 'Project' directory improves reproducibility	20	• Quiz questions: What is a structured 'Project' directory and what are its benefits to reproducibility • Video with detailed discussion of quiz solutions
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.	• Be able to create a structured 'Project' directory within RStudio • Understand how RStudio can be used to create 'Project' templates	25	• Discussion questions on how the trainee has saved and tracked research project files for previous research projects and related barriers to reproducibility • Short audio recording of two Co-Is discussing their answers
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.	• Create a 'Project' template in RStudio to initialize consistently-formatted 'Project' directories • Initialize a new 'Project' directory using this template	15	• Applied exercise: Create and save a 'Project' template that meets specifications provided for an example research group • Video demonstrating a detailed solution

Table 1: Modules for the first sequence, **'Improving the Reproducibility of Experimental Data Recording'**. The color of each module's title indicates whether the module focuses on **Principles** (blue), **Implementation** (red), or **Case study examples** (black). This table is continued over several pages. *(continued)*

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video length (min.)	Extra educational materials
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.	• Describe version control • Explain how version control can be used to improve reproducibility for data recording	10	• Discussion questions, including discussion of how the trainee has managed evolving research project files in previous projects and related barriers to reproducibility • Short audio recording of two Co-Is giving their own answers
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.	• 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*)	10	• Discussion questions: Describe how past research projects shared files without using version control • Short audio file with two Co-Is discussing their answers
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.	• 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	20	• Applied exercise: Use RStudio to initialize *git* version control for a directory and to make several tracked changes. Create a matching *GitLab* repository and use RStudio to push local changes to this *GitLab* version of the directory • Video walking trainees through a detailed solution

Table 2: Modules for the second sequence, **'Improving the Reproducibility of Experimental Data Pre-Processing'**. The color of each module's title indicates whether the module focuses on **Principles** (blue), **Implementation** (red), or **Case study examples** (black). This table is continued over several pages.

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video length (min.)	Extra educational materials
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.	• Define 'pre-processing' of experimental data • Describe an open source code script and explain how it can increase reproducibility of data pre-processing	15	• Discussion questions, including common pre-processing needs and practices • Short audio recording of two Co-Is giving their answers
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.	• 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	10	• Applied exercise: Given a simple example dataset and a data cleaning task, write and run an R script to perform the task. Then adapt that script to re-use it on a second dataset. Hints will be provided for those new to R • Video providing a detailed walk-through of a solution to the applied exercise
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.	• 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'.	15	• Quiz questions: What is R's 'tidyverse' and why is it a powerful yet user-friendly tool for improving the reproducibility of research projects • Video with detailed answers and explanations for the quiz questions • Links to free sources for developing more 'tidyverse' coding skills

Table 2: Modules for the second sequence, **'Improving the Reproducibility of Experimental Data Pre-Processing'**. The color of each module's title indicates whether the module focuses on **Principles** (blue), **Implementation** (red), or **Case study examples** (black). This table is continued over several pages. *(continued)*

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video length (min.)	Extra educational materials
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.	• 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	15	• Quiz questions: Why are complex data formats often used within steps of experimental data pre-processing in open-source software and how does their use complicate the use of reproducibility tools • Video providing detailed answers
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.	• 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	15	• Applied exercise: Starting with example data in a complex, list-based format, explore the data and extract specified elements, including with the *broom* and *biobroom* packages; • Video providing a detailed walk-through of the solution to this exercise
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.	• 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	20	• Applied exercise: With an example dataset in a complex, 'untidy' data format in R, convert it to a 'tidy' format and create simple plots with this 'tidy' dataset • Video demonstrating a detailed solution to the applied exercise

Table 2: Modules for the second sequence, **'Improving the Reproducibility of Experimental Data Pre-Processing'**. The color of each module's title indicates whether the module focuses on **Principles** (blue), **Implementation** (red), or **Case study examples** (black). This table is continued over several pages. *(continued)*

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video length (min.)	Extra educational materials
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.	• 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	15	• Discussion questions: How reproducible data pre-processing protocols can make biomedical research more reproducible at the data pre-processing stage in the trainee's research area • Short audio recording of two Co-Is giving their own answers to these discussion questions
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.	• 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 RMarkdown • Apply it to several different datasets with the same format	15	• Applied exercise: Create, save, and render their own RMarkdown document through RStudio • Video providing a detailed walk-through of a solution to the applied exercise
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.	• 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	20	• Quiz questions: Test understanding of how and why we created a reproducible data pre-processing protocol for this pre-processing step, and how this improves reproducibility for the research group; • Short video with a detailed discussion of quiz questions