

GRA Continuity Guide

Cannell Research Team

Table of contents

Welcome and Overview	4
Expectations and Advice	5
General Expectations	5
Advice for GRA Success	6
Onboarding	7
Getting Paid	7
IRB Requirements	8
Complete CITI training	8
Complete Conflict of Interest (COI) Disclosure	9
I Style Guide	10
Overview	11
Emphasizing Text	12
Application names	12
Keywords	12
Programming code	13
Clickable operations in applications	13
File paths	13
File and Folder names	14
Document Library Structure	15
Folder structure	15
Folder names	15
File structure	17
File names	17
Collaborating on Files	19
Appendix	21
Appendix A. Style Cheat Sheet	21

Appendix B. UTHHealth Brand Standards	22
Logos	22
Colors	22
Typefaces	23
 II GRA Tasks	 24
IRB Tasks	25
IRB Documents	25
iRIS	25
IRB Change Requests and Amendments	25
IRB Personnel Change Requests	27
 Data Management and Analysis Tasks	 29
Protected Health Information	29
Reproducible Research	30

Welcome and Overview

Welcome to the team!

We are excited to have you join us and want you to be successful. This guide is intended to help Graduate Research Assistants (GRAs) acclimate to the project, find resources, and gain a clear understanding of expectations.

Expectations and Advice

General Expectations

We believe that setting clear expectations upfront is kind and will help you to have a successful GRA experience. In that spirit, here are some of the general expectations we have for our team members:

Work Your Committed Hours

Nobody likes to be micromanaged, and we don't like micromanaging. In general, we prefer to have a relaxed work environment where people are free to work whenever and wherever they work best. However, at the end of the day, we are all getting paid to do a job, and the job needs to get done. If you committed to working 19.5 hours per week, the expectation is that the project will get a full 19.5 hours of your best effort each week if there is work that needs to be done.

Be Available

Try your best to be available when the team needs you. For example, be responsive to emails and attend team meetings. If you are sick, need to adjust your hours, or have a conflict, let the program manager know and we can arrange accommodation. We realize that emergencies, class commitments, and other situations arise.

Code Collaboratively through GitHub

All collaborative coding tasks will be tracked through GitHub. If you aren't familiar with GitHub, you can start by checking out the [chapters in R4Epi](#) on using Git and GitHub. When you complete your coding tasks, push them to GitHub so that the PI can review and approve them.

Be Flexible

Be ready for many kinds of work projects. Some weeks may be all coding, other weeks may have responsibilities that include working in iRIS, writing, or researching. All of these tasks have value and help move the project forward to success.

Advice for GRA Success

The most important trait you can master as a GRA is developing a habit of taking initiative. Of course, you will need some time at the beginning of your GRA experience to figure out what is going with the project you've been hired to work on and how each team member's role/personality should be best managed. However, after a few weeks to a month, you should start to take initiative on your own.

What does this mean? To us, it means:

Get the Big Picture

Make sure you have a solid understanding of the project's 'big picture.' What is the project trying to accomplish overall? What are some big milestones that need to be accomplished by the end of the year and/or the end of the project? If you don't know or aren't sure, ask!

Fill in the Details

Once you understand the 'big picture' goals for the project, really work on developing your ability to think through the individual steps that need to occur for those big picture goals to be achieved. As an example, let's say that you know that one of the project goals is to create a participant/patient recruiting dashboard so that we can track our recruitment progress. What steps need to happen? Well, we need data from somewhere, we typically need to clean that data, we usually need to do some exploratory analysis of the data, and then we need to start drafting useful metrics that we can add to our dashboard.

Make it Happen

Once you've thought through the steps that need to occur, don't wait for someone to assign you a task. This is where you take the initiative to complete a task that you know needs to be completed without someone telling you. Having said that, there's nothing wrong with running it by other team members first to make sure you aren't about to start on something that will be a waste of your time.

Onboarding

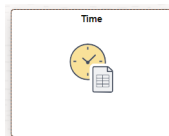
Getting Paid

We all want to get paid! To make sure you do, please follow the steps outlined below:

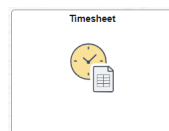
1. If you haven't already, set up your UTHealth Virtual Private Network (VPN) following these [instructions](#).
2. Log into the VPN using your UTHealth credentials.
3. Log into the [Employee Self-Service site](#).

The image shows a login screen for UTHealth Houston PeopleSoft HCM. At the top is the UTHealth Houston logo. Below it is the text "PeopleSoft HCM". There are three input fields: "User ID" (with a cursor), "Password", and "Select a Language" (with a dropdown menu showing "English"). Below these fields is a black "Sign In" button. At the bottom, there is a checkbox labeled "Enable Screen Reader Mode".

4. On the next screen, click on the Time button.



5. And then the Timesheet button.



- Finally, fill in your electronic timesheet. After your hours are entered, the program manager will approve them in the system, and you will get paid.

SPH Grad Res Assist - Master's
Emp# Record: 0
Earliest Change Date: 08/01/2024

Select Another Timesheet
*View By: Calendar Period
*Date: 08/16/2024
Previous Period Next Period

From Friday 08/16/2024 to Saturday 08/31/2024

Fri 8/16	Sat 8/17	Sun 8/18	Mon 8/19	Tue 8/20	Wed 8/21	Thu 8/22	Fri 8/23	Sat 8/24	Sun 8/25	Mon 8/26	Tue 8/27	Wed 8/28	Thu 8/29	Fri 8/30	Sat 8/31	Total Time Reporting Code	Type	Taskgroup		
																		PSNONCATSK		
																		PSNONCATSK		
																		PSNONCATSK		

Save for Later Submit

Reported Time Status Summary Exceptions Payable Time

Reported Time Status Personalize Find 1 of 1

Date	Total TRC	Description	Comments
	0.000000		

If you have issues with any step in this process, you may reach out to the IT help desk at [713-486-4848](tel:713-486-4848).

IRB Requirements

IRB stands for Institutional Review Board, and operating the IRB is one of the most important functions of our school's Committee for the Protection of Human Subjects. As you might have guessed from the name, the IRB exists to review all research the school plans to conduct and make sure that it is in compliance with all federal, state, and ethical guidelines intended to protect human research participants.

One of the first things we will need to do when you join the team is add you to our IRB protocols. If you haven't already done so, we will need you to complete two steps before we can add you:

Complete CITI training

CITI is a program that provides human subjects protection training to many universities across the country. To complete your CITI training:

- Follow the instructions on the [UTHealth IRB website](#) to log on to CITI and register.
- After registering, complete the course called, "GCP – Social and Behavioral Research Best Practices for Clinical Research."
- Download the Certificate of Completion to send to the study PI and program manager so that they may add you to the IRB.

Complete Conflict of Interest (COI) Disclosure

Every UTHealth student, staff, and faculty member who participates in research needs to complete a COI annually. Please fill out your COI on the [UTHealth COI website](#). You will also be able to look up frequently asked questions while you are there. Students very rarely have any financial conflicts of interest that they need to report, but feel free to reach out to [Dr. Cannell](#) if you have any concerns.

Part I

Style Guide

Overview

The ultimate goal of a style guide is to reduce cognitive load. Doing so should improve the quality and consistency of our work product and make our work lives easier and more efficient. How does a style guide do this?

- First, it reduces the number of choices we need to make as we are authoring content. For example, we don't need to come up with our own answers to questions like, "where should I put this document?" or "what should I name this folder?"
- Second, having the predetermined choices written down for reference reduces the amount of information we need to store in our intentional memory. For example, "OK, remember to always start these file names with the date written as yyyy-mm-dd". Although we don't need to *intentionally* memorize them, these predetermined choices may eventually bleed over into our memory by accident through repetition.
- Third, having uniformly styled content makes it easier for others – *including future us* – to find what we are looking for and use it. We can focus our brain power on the content instead of the style and/or organization of the content.

Emphasizing Text

Use the following conventions to emphasize keywords, concepts, code snippets, and other words or phrases that need to stand out or be emphasized.

Application names

Capitalize the names of applications.

- Do this: Microsoft Outlook, Outlook, Microsoft Planner, Planner.
- Do not do this: microsoft outlook, outlook, microsoft planner, planner.

Keywords

Underline a keyword or phrase if it is a keyword or phrase that we would want to define in the glossary of the document (whether the document actually has a glossary or not).

- **Do this:** A standard operating procedure (SOP) is...
- **Do not do this:** **A standard operating procedure (SOP)** is..., A standard operating procedure (SOP) is...

Bold a keyword or phrase that we want to call attention to, but it is not necessarily a keyword or phrase that we would want to define in the glossary of the document (whether the document actually has a glossary or not).

- **Do this:** ****Step 1.****, **** Do this:****.
- **Do not do this:** Step 1., Do this.

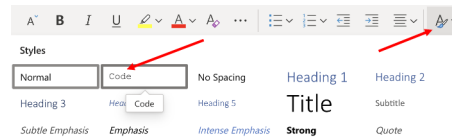
At times, we will also *Italicize* keywords or phrases that we want to call attention to, but are not necessarily keywords or phrases that we would want to define in the glossary of the document (whether the document actually has a glossary or not). In general, we will follow [standard English grammar rules for using the italics typeface](#) (pay special attention to bullet 14). Further, “avoid using italics with other stylized typefaces, such as bold and underline. Since all three are designed to make words stand out, only one at a time is necessary.” ([Ellis, 2022](#))

- **Do this:** We do not want to coerce participants into signing the consent form.

- **Do not do this:** We do NOT/~~not~~/~~not~~/~~not~~/not want to coerce participants into signing the consent form.

Programming code

Snippets of programming code should be written in the **Courier New** font with a light gray background. This matches the style convention used by many popular programming websites like GitHub and Stack Overflow. To make using this format easier, any document that was created by duplicating the New SOP Template will have a **code** option in the styles menu.



- **Do this:** `utcNow(), dplyr::select()`.
- **Do not do this:** `utcNow(), dplyr::select(), dplyr::select(), "dplyr::select()"`

Clickable operations in applications

In many of the applications we use, clickable steps or operations serve the same function as programming code. For example, in Microsoft word, we don't type "bold('Do this')". Instead, we highlight the phrase "Do this" with our mouse and then click the **B** button in the toolbar. Then, Microsoft Word takes care of the programming behind the scenes for us. Therefore, clickable operations that need to be performed in an application should be written in same style used to write code. To make using this format easier, any document that was created by duplicating the New SOP Template will have a **code** option in the styles menu.

- **Do this:** Click on the **Insert** tab...
- **Do not do this:** Click on the *Insert*/INSERT/**insert**/insert tab.

File paths

All file paths should be written in italic text with a light gray background. To make using this format easier, any document that was created by duplicating the New SOP Template will have a **File Paths** option in the styles menu. Additionally, the root directory (i.e., starting point) for all SharePoint/Teams documents is the Documents folder.

- **Do this:** *Documents/General/SOPs/02 Style Guide.docx*.
- **Do not do this:** "Documents/General/SOPs/02 Style Guide.docx".

File and Folder names

File and Folder names are essentially very short file paths. Therefore, file names and folder names should be written using the file paths guidelines from above. There is one exception to this rule. When creating a hyperlink to a file or folder, then the standard hyperlink style (i.e., link) should be used.

- **Do this:** The SOP folder.
- **Do not do this:** The “SOP” folder. The **SOP** folder.

Document Library Structure

Folder structure

- Each folder in the document library should generally be associated with a specific theme or topic, which may have sub-themes and subtopics.
 - **Do this:** IRB Documents or Budget Documents.
 - **Do not do this:** Budgets and IRB Documents.
- Folders should not generally be associated with specific people or positions. There are at least two reasons why we don't want to associate folders with people or positions. First, people and positions change over time. Topics can change too, but they tend to be more stable. Second, and perhaps more importantly, this is a shared document library. We shouldn't think of anything we keep in this library as belonging to any individual. As such, the names and purposes of documents should be clear to others who may need to use them. For example, it's going to be much easier for most people to reason about what is contained in the IRB folder than in the Brad folder.
 - **Do this:** IRB Documents or Budget Documents.
 - **Do not do this:** Brad's Documents or GRA Documents.
- Folders that can be logically nested inside an existing folder, or two similar folders that can logically be combined under a single parent folder, should be.
 - For example, a folder containing CITI training certificates should probably be nested in the IRB Documents folder.
 - A folder containing GRA applicant resumes, a folder containing GRA work schedules, and a folder containing job announcements should probably all be nested in a Hiring folder, which should probably be nested in a HR folder.

Folder names

- Folders for programming code repositories should be written in snake case. This is to help ensure that the folder name works easily with R, Git, Bash, and other software used in the data analysis pipeline.
 - **Do this:** `detect_public_repo`.

- **Do not do this:** DETECT Public Repo.
- All other folders should be written in title case.
 - **Do this:** Budget Documents.
 - **Do not do this:** budget documents.
- Folder names should be descriptive enough for most people to reasonably be able to figure out what the folder contains from the name.
 - **Do this:** Budget Documents.
 - **Do not do this:** Brad’s Stuff or MU-334-011.
- Folder names should be succinct.
 - **Do this:** Pre-award.
 - **Do not do this:** Narrative - Spec Aims - Budgets - and Other Documents Submitted to NIA in the Original Proposal.
- Folder names should only include letters, numbers, underscores, or dashes. Other characters can cause syncing failures and/or issues with other file systems.
 - **Do this:** Materials and Operations.
 - **Do not do this:** Materials & Operations.
- Folders that can be naturally arranged by dates should be. The folder name should begin with the date written in the **yyyy-mm-dd** format. This format is important because dates written in this format will naturally be arranged in correct chronological order across years. Some examples of folder topics that may have a natural chronological order include IRB documents, budgets, and meeting minutes.
 - **Do this:** 2023-02-17 Budget.
 - **Do not do this:** February Budget or 02-17-2023 Budget.
- Folders that can be naturally arranged in a sequential order should be. When doing so, the folder names should be sequentially numbered. Single digits should be prefixed with a zero (0). This format is important because single digits written in this format will naturally be arranged in correct sequential order when the number of folders in the sequence exceeds 9. Data analysis projects are an example where folder topics may have a natural sequential order.
 - **Do this:** 01_data_import, 02_data_clean, 03_table_01, 04_table_04.
 - **Do not do this:** 1_data_import, 2_data_clean, table_1, my_other_table.
- Folders for documents related to peer-reviewed manuscripts should begin with the first author’s last name.
 - **Do this:** Cannell - Protocol Paper.
 - **Do not do this:** Protocol Paper.

- When multiple folders have a similar theme/topic, begin the folder names with a description of the theme/topic. Doing so will ensure that all folders with that theme/topic are grouped together in the file list.
 - **Do this:** Reliance Agreement UAB, Reliance Agreement UCSF, Reliance Agreement JHU.
 - **Do not do this:** UAB Reliance Agreement, UCSF Reliance Agreement, JHU Reliance Agreement.

File structure

- Files should generally be grouped together and placed in a folder that describes the theme or topic they belong to.
- Folders that contain a single file should be rare.

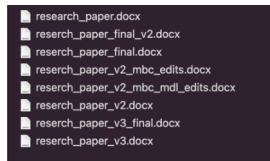
File names

- Programming code files should be written in snake case. This is to help ensure that the file name works easily with R, Git, Bash, and other software used in the data analysis pipeline.
 - **Do this:** 01_data_import.Rmd.
 - **Do not do this:** Data Import.Rmd.
- All other file names should be written in title case.
 - **Do this:** Cannell CITI Certificate.pdf.
 - **Do not do this:** cannell citi certificate.pdf.
- File names should be descriptive enough for most people to reasonably be able to figure out what the file contains from the name.
 - **Do this:** Cannell CITI Certificate.pdf.
 - **Do not do this:** CITI.pdf.
- File names should be succinct.
 - **Do this:** Approved Protocol.docx.
 - **Do not do this:** DETECT-RPC - Phase 1 - Focus Groups - Research Protocol Final Accepted Copy.docx.
- File names should only include letters, numbers, underscores, or dashes. Other characters can cause syncing failures and/or issues with other file systems.
 - **Do this:** Materials and Operations.xlsx.
 - **Do not do this:** Materials & Operations.xlsx.

- Files that can be naturally arranged by dates should be. The file name should begin with the date written in the **yyyy-mm-dd** format. This format is important because dates written in this format will naturally be arranged in correct chronological order across years. Some examples of files that may have a natural chronological order include IRB documents, budgets, and meeting minutes.
 - **Do this:** 2023-02-17 Budget.xlsx.
 - **Do not do this:** February Budget.xlsx or 02-17-2023 Budget.xlsx.
- Files that can be naturally arranged in a sequential order should be. When doing so, the files names should be sequentially numbered. Single digits should be prefixed with a zero (0). This format is important because single digits written in this format will naturally be arranged in correct sequential order when the number of files in the sequence exceeds 9. Data analysis projects are an example where files may have a natural sequential order.
 - **Do this:** 01_data_import.Rmd, 02_data_clean.Rmd, 03_table_01.Rmd, 04_table_04.Rmd.
 - **Do not do this:** 1_data_import.Rmd, 2_data_clean.Rmd, table_1.Rmd, my_other_table.Rmd.
- Begin file names with a description of a theme or topic when there are multiple files that will have a similar theme or topic. Doing so will ensure that all of the files with that theme are grouped together in the file list.
 - **Do this:** Reliance Agreement UAB.pdf, Reliance Agreement UCSF.pdf, Reliance Agreement JHU.pdf.
 - **Do not do this:** UAB Reliance Agreement.pdf, UCSF Reliance Agreement.pdf, JHU Reliance Agreement.pdf.

Collaborating on Files

Have you ever worked on a paper or report and had a folder on your computer that looked something like this?



Saving a bunch of different versions of a file like this is a real mess, and it tends to get worse the more collaborators we have. What is contained in each document again? What order were the documents created in? What are the differences between the documents? Does the version of the paper that Doug emailed this morning contain the edits that Jason made yesterday? Versioning helps us get around these problems, and luckily, we have some good tools for versioning. Therefore, we should generally not be emailing files to each other. Instead, we should strive to work on Microsoft Office Documents online and use Git and GitHub for collaborative computer programming. There are links to instructions for both below.

Please see the Using Word SOP for instructions on how to use Word's versioning system (Excel and PowerPoint have similar systems).

Please see for instructions for versioning other file types with Git and GitHub.

Appendix

Appendix A. Style Cheat Sheet

Theme: Default

The Title Written in Title Case

The Subtitle Written in Title Case

The normal style.

The default paragraph font style.

Level 1 header

Level 2 header

Level 3 header

Level 4 header

The subtle emphasis style.

The emphasis style.

The intense emphasis style.

The strong style.

The quote style.

The intense quote style.

THE SUBTLE REFERENCE STYLE.

THE INTENSE REFERENCE STYLE.

The book title style.

The list paragraph style.

The toc 1 style.

The toc 2 style.

The hyperlink style.

Appendix B. UTHealth Brand Standards

Logos

[Here is a link to the online logo library.](#)

Colors

[Here is a link the color standards page.](#)

Color Group	Name
Traditional	University Orange
Traditional	University Blue
Traditional	University Gray
Secondary	Gulf Blue
Secondary	Mustard Gold
Secondary	Dark Sage
Secondary	Dusty Lavender
Neutral	Light Sage
Neutral	Sand

Typefaces

[Here is a link to the typefaces standards page.](#)

We don't have easy access to many of these fonts. Here are some that we can use:

- Garamond
- Univers
- Times New Roman
- Avenir
- Helvetica
- Arial
- Calibri

Part II

GRA Tasks

IRB Tasks

IRB Documents










































































Local copies of all IRB documents for the DETECT project are located in our SharePoint Document Library at [Documents/General/IRB Private](#). You may need to read, copy, and/or update these files from time to time.

iRIS

Virtually all IRB documentation will be submitted and stored in [UTHealth's Integrated Research Information Software \(iRIS\)](#). If you have iRIS-related questions, you may contact the iRIS help line at [713-500-7960](tel:713-500-7960).

As part of your responsibilities as a GRA, you may or may not be asked to directly create or edit research protocols/documents in the iRIS system. After you sign in, you will land on the iRIS home screen. There can be multiple sections and studies listed on your iRIS home screen, but there should be a little pencil and paper icon on the left-hand side of every active study and study update that you have access to. Here is an example from Dr. Cannell's iRIS home screen with the pencil and paper icon highlighted in red.

17 result(s) found...

Click to open Study Worksheet	Study Name	Review Board	IRB Number	IRB Expiration	Study Title	Principal Investigator	Actions
	Emergency Counseling for Child Molesters	Committee for the Protection of Human Subjects	HSC-SPH-23-0105		Development of Clear Abuse Through Emergency Care Techniques - Protocol for Primary Care DETECT-APC Baseline APS	Cannell, Michael Stael, PhD, MPH	     
	Active - No Continuing Review	Committee for the Protection of Human Subjects	HSC-SPH-22-0733	06/30/2025	Detection of Elder Abuse Through Emergency Care Techniques for Research for Primary Care Clinicians	DET-APC Phase 1 - Cannell, Michael Stael, PhD, MPH	     
	Emergency	Committee for the Protection of Human Subjects	HSC-SPH-19-0139		DET-APC Phase 1 - Knowledge Assessment and Quality Improvement	Cannell, Michael Stael, PhD, MPH	     
	Active - No Continuing Review	Committee for the Protection of Human Subjects	HSC-SPH-18-0630	02/09/2025	Detection of Elder Abuse Through Emergency Care Techniques (DET-APC)	Cannell, Michael Stael, PhD, MPH	     
	Expired	Committee for the Protection of Human Subjects	HSC-SPH-21-0374	06/30/2023	Implementation and Evaluation of a Statewide Police Peer-Support Network at the University of Texas	Cannell, Michael Stael, PhD, MPH	     
	Active and with Reapproval	Committee for the Protection of Human Subjects	HSC-SPH-19-0632	12/31/2023	UTSP Peer Support	Cannell, Michael Stael, PhD, MPH	     
	Active	Committee for the Protection of Human Subjects	HSC-SPH-21-0272	11/30/2023	in Health to Increase Service Utilization in Adults Experiencing Perinatal Abuse	Perinatal Care Management - Cannell, Michael Stael, PhD, MPH	     
	Active - No Continuing Review	Committee for the Protection of Human Subjects	HSC-SPH-19-0461	04/30/2024	Management of Depression and Social Isolation in Older Adults with Memory Impairment with HHS	SA-Caregiver for Depression - Cannell, Michael Stael, PhD, MPH	     
	Active - No Continuing Review	Committee for the Protection of Human Subjects	HSC-SPH-19-0639	05/31/2024	A health and safety interview for adults aged 65 and older who recently used emergency medical services	Cannell, Michael Stael, PhD, MPH	     
	Active - No Continuing Review	Committee for the Protection of Human Subjects	HSC-SPH-19-0639	05/31/2024	DET-APC Phase 1 - Knowledge Assessment and Quality Improvement	Cannell, Michael Stael, PhD, MPH	     
	Active - No Continuing Review	Committee for the Protection of Human Subjects	HSC-SPH-20-0725	06/30/2024	DET-APC Phase 1 - Knowledge Assessment and Quality Improvement	Cannell, Michael Stael, PhD, MPH	     

IRB Change Requests and Amendments

One task that we commonly ask GRA's to assist us with is submitting change requests and amendments. Change requests and amendments are typically used to let the IRB know that we intend to change something about the way we are conducting our study. For example, if

we originally told the IRB that we would pay research participants \$20 and we later decide to pay them \$25, then we will need to submit that change to the IRB for approval.

To access a change request form, click on the pencil and paper icon for the study you are working on. The next screen should look something like this.

Protocol Items	
	Study Application
	Informed Consent
	Other Study Documents

Initial Review Submission Form	
	IRB - Change requests and Amendments
	IRB - Continuing Review Form
	IRB - Data Safety Monitoring Report (DSMR)
	IRB - IND Safety Report
	IRB - Internal Adverse Event
	IRB - Personnel Change Request Form
	IRB - Protocol Deviation
	IRB - Removal of Key Study Personnel Form
	IRB - Study Closure Report
	IRB - Study Miscellaneous Form
	IRB - Unanticipated Problem Report

One of the options will be **IRB - Change requests and Amendments** – click it. On the next page, click the **Add a New Form** button. Each protocol is somewhat unique, so it’s difficult to provide instructions on precisely what to type into the form. However, here are some notes that previous GRAs found useful about the DETECT project specifically. Please keep in mind that we can change any part of the form before our final submission.

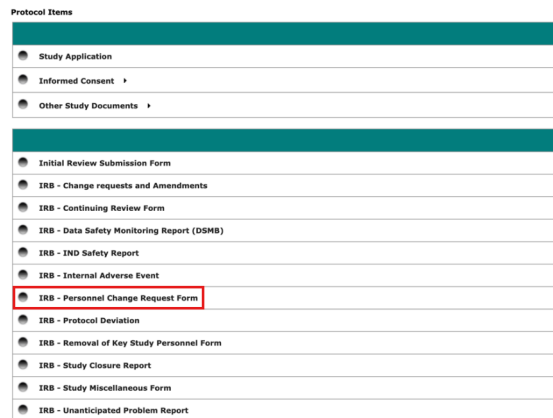
- Update number: If this is the first protocol update, then the update number will be 1. If it’s the second, then 2, and so on.
- Short description. Just do your best to write something succinct and descriptive.
- Does the revision to this protocol require a formal change to the title: Typically, not.
- Memorial Hermann location questions: Typically, the answer to all of these is “No.”
- Type of revisions: Select those that seem applicable.
- Revision description and rationale: What changes do we want to make and why do we want to make them?
- Who is initiating this change: Typically, Investigator-initiated.
- Risk increase: Typically, the revision will not increase risk to participants.
- Designated Department Approval/Head: Bijal Bala.
- Funding: DETECT is federally funded by the National Institutes of Health (NIH) and the National Institute on Aging (NIA).

It can often be helpful to start by looking at other change requests that have been submitted and approved. At the end of the day, we ask that you just fill out the form(s) to the best of your ability. When you are done, please email the PI and program manager to let them know. They will review all of the forms and likely make edits before the final submission.

IRB Personnel Change Requests

Another common task we ask GRAs to help with is adding or removing personnel from the study.

To access a personnel change request form, click on the pencil and paper icon for the study you are working on. The next screen should look something like this.



The screenshot shows a web interface titled "Protocol Items". It contains a list of items, each with a circular icon and a text label. The items are: Study Application, Informed Consent, Other Study Documents, Initial Review Submission Form, IRB - Change requests and Amendments, IRB - Continuing Review Form, IRB - Data Safety Monitoring Report (DSMB), IRB - IND Safety Report, IRB - Internal Adverse Event, IRB - Personnel Change Request Form (highlighted with a red box), IRB - Protocol Deviation, IRB - Removal of Key Study Personnel Form, IRB - Study Closure Report, IRB - Study Miscellaneous Form, and IRB - Unanticipated Problem Report.

One of the options will be **IRB - Personnel Change Request Form** – click it. On the next page, click the **Add a New Form** button. Here are some notes for filling out the form that previous GRAs found useful. Please keep in mind that we can change any part of the form before our final submission.

- Are you requesting a change to the principal investigator: No.
- Are you requesting the addition of a co-investigator: Maybe. Ask the PI or Program Manager if you are unsure.
- Are you requesting changes to key study personnel: This is the most common scenario.
- To add co-investigators or key personnel, we will need to have their CITI certificate and they will have to complete a COI. Both of these documents are explained above.
- Are you removing anyone from the study at this time: If we are, please just write their name and a brief explanation. For example, “Jane Smith. She graduated in May.”
- Typically, we prefer not to add and remove personnel on the same form. So, if we need to add Jon Smith to the protocol and we need to remove Jane Smith from the protocol, we should do so using two separate forms.
- Significant financial interest: This is almost always “No.” So, please select “No.” The PI will change it to “Yes” before the final submission if necessary.
- Additional information: You can leave this blank.
- Designated Department Approval/Head: Bijal Bala.
- Funding: DETECT is federally funded by the National Institutes of Health (NIH) and the National Institute on Aging (NIA).

At the end of the day, we ask that you just fill out the form(s) to the best of your ability. When you are done, please email the PI and program manager to let them know. They will review all of the forms and likely make edits before the final submission.

Data Management and Analysis Tasks

Most, if not all, of our research projects will involve the management and analysis of data. This section of the SOP includes information intended to make it easier for us to find and use the files we need to access, manage, and analyze project data. These files broadly include project data files, programming code files, and general project documentation files. All three will be covered below. However, there are a couple of general topics that we should familiarize ourselves with first. Namely, protective health information (PHI), reproducible research, and GitHub.

Protected Health Information

We should already be familiar with PHI from our [CITI training courses](#). However, this concept is so important that it bears repeating here. The UC Berkely Human Research Protection Program website summarizes PHI this way:

Protected Health Information (PHI) is any information in the medical record or designated record set that can be used to identify an individual and that was created, used, or disclosed in the course of providing a health care service such as diagnosis or treatment. HIPAA regulations allow researchers to access and use PHI when necessary to conduct research. However, HIPAA applies only to research that uses, creates, or discloses PHI that enters the medical record or is used for healthcare services, such as treatment, payment, or operations.

It further goes on to list 18 identifiers that can be used to identify the individual associated with the health records. They are:

1. Names;
2. All geographical subdivisions smaller than a State, including street address, city, county, precinct, zip code, and their equivalent geocodes, except for the initial three digits of a zip code, if according to the current publicly available data from the Bureau of the Census: (1) The geographic unit formed by combining all zip codes with the same three initial digits contains more than 20,000 people; and (2) The initial three digits of a zip code for all such geographic units containing 20,000 or fewer people is changed to 000.

3. All elements of dates (except year) for dates directly related to an individual, including birth date, admission date, discharge date, date of death; and all ages over 89 and all elements of dates (including year) indicative of such age, except that such ages and elements may be aggregated into a single category of age 90 or older;
4. Phone numbers;
5. Fax numbers;
6. Electronic mail addresses;
7. Social Security numbers (SSN);
8. Medical record numbers (MRN);
9. Health plan beneficiary numbers;
10. Account numbers;
11. Certificate/license numbers;
12. Vehicle identifiers and serial numbers, including license plate numbers;
13. Device identifiers and serial numbers;
14. Web Universal Resource Locators (URLs);
15. Internet Protocol (IP) address numbers;
16. Biometric identifiers, including finger and voice prints;
17. Full face photographic images and any comparable images; and
18. Any other unique identifying number, characteristic, or code (note this does not mean the unique code assigned by the investigator to code the data)

Reproducible Research

We will strive to make our projects conform to best practices for promoting reproducible research. Many resources that describe what reproducible research is, and why it's important, are available on the internet, and we encourage you to look at some of them. Briefly, here is an excerpt from [Wikipedia](#) that is good enough for our purposes:

“The term reproducible research refers to the idea that scientific results should be documented in such a way that their deduction is fully transparent. This requires a detailed description of the methods used to obtain the data and making the full data set and the code to calculate the results easily accessible. This is the essential part of open science.

To make any research project computationally reproducible, general practice involves all data and files being clearly separated, labelled, and documented. All operations should be fully documented and automated as much as practicable, avoiding manual intervention where feasible. The workflow should be designed as a sequence of smaller steps that are combined so that the intermediate outputs from one step directly feed as inputs into the next step. Version control should be used as it lets the history of the project be easily reviewed and allows for the documenting and tracking of changes in a transparent manner.”

Because the data we work with almost always includes (PHI), we will not typically be able to make our data freely available to the general public. However, we will do our best to conform to the remaining elements of reproducible research described above. Two of the most important tools we will use to make our research more reproducible are [git](#) and [GitHub](#).

GitHub

[GitHub](#) is a website specifically designed to facilitate collaboratively creating programming code. In many ways, GitHub is a cloud-based file storage service like Dropbox, Google Drive, or OneDrive. However, GitHub contains some additional features that make it an exceptional tool for collaborating our research projects. Some of these features include:

- Repositories
- Projects
- Discussions
- Wikis

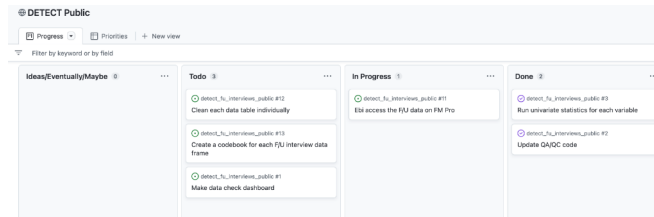
0.0.0.0.1 * GitHub Repositories

We will make extensive use of [GitHub](#) repositories. If you aren't already familiar with git and GitHub, please start by reading the relevant chapters in [R for Epidemiology](#). It's probably a good idea to read those chapters even if you are already familiar with git and GitHub because they describe how we will use our GitHub repositories. Here are some key summary points to keep in mind:

- Almost all programming code, code documentation, and even a large amount of general project documentation will flow through our projects' GitHub repositories.
- Almost all of our programming tasks will be tracked using GitHub projects.
- Please work with the project PI and/or program manager to make sure you are able to locate and access the GitHub repositories and GitHub project(s) associated with the research project(s) you are working on.

0.0.0.0.2 * GitHub Projects

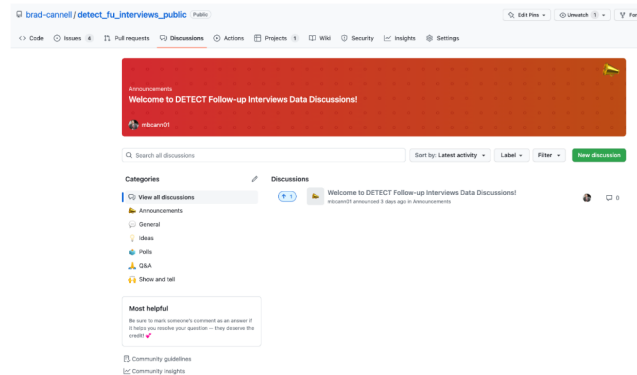
GitHub projects – we will sometimes refer to them as project boards – allow us to organize, track, and communicate about [GitHub issues](#). The name “issues” sort of implies a problem, but that isn't how we will use them. We will use issues as tasks that need to be completed, and we will use project boards to track those tasks. In fact, for the rest of this continuity guide, I will refer to them as tasks instead of issues.



The example above is from the DETECT project and we can see examples of tasks that are yet to be started (i.e., Ideas/Eventually/Maybe and Todo), in progress, and already completed (i.e., Done). We can click on the tasks to open a detailed view where we can add a description, we can add sub tasks, and we can leave messages for each other about the task.

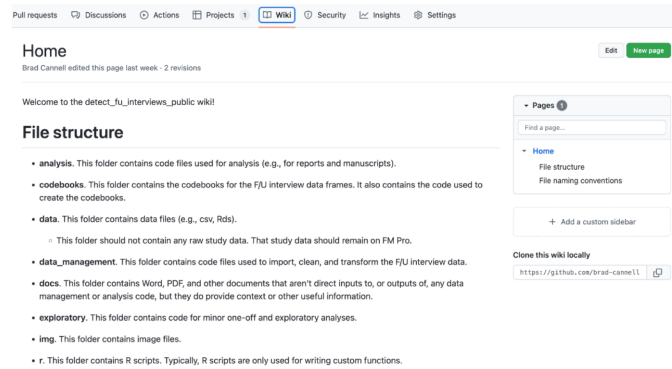
There are a couple of advantages to messaging each other about tasks in GitHub instead of sending emails. First, the messages stay with the tasks they are about. You don't have to go search for them later, and all the information/context you need is in one place. Second, when you graduate and leave UTHealth, your email account will be deleted. However, that doesn't necessarily mean you will want to quit working on a project. Fortunately, if we've been communicating in GitHub instead of through email, nothing important is lost.

0.0.0.0.3 * GitHub Discussions



As the name indicates, GitHub discussions are a place where we can have discussions about the project. For example, we can exchange ideas and we can ask each other general questions that don't necessarily pertain to a specific task.

0.0.0.0.4 * GitHub Wikis



We can generally think of GitHub wikis as an SOP for a specific repository. Wikis aren't about what to do – that's what tasks are for. Rather, wikis tend to contain information about how to navigate and use the repository and how to complete our tasks in a consistent way. For example, the screenshot above shows part of the [DETECT follow-up interview repository's wiki](#). In this particular screenshot, we can see instructions for navigating and using the repository's file/folder structure.

Communicating About Research Projects

Over time, I have come to really appreciate the number and quality of tools that GitHub provides for collaboratively coding and working on research projects. It truly is an amazing tool in my opinion. Having said that, I understand how it can be pretty intimidating – or at least overwhelming – at first. So, here is a little cheat sheet to get us started.

Tool	Task	Example
Email	General communication that isn't about a specific project.	"What time are we meeting next week?"
Microsoft Planner	Assign and communicate about tasks, but not tasks that are specific to a single project.	Task: Update CITI training.
GitHub wikis	General information about a specific project, but not communication about a specific task.	"Data was collected between [date 1] and [date 2]."
GitHub discussions	Communication about a specific project, but not a specific task.	"Did we measure depression in this study?"
GitHub issues (tasks)	Communication about specific tasks.	[Assuming there is a task that directs us to perform a linear regression] "We decided to use X, Y, and Z as predictors in the model, right?"

Here are some additional notes about communication to keep in mind:

- Please follow the guidance in R4Epi when you have a coding question.
- When creating commits, please follow the guidance in R4Epi.
 - Scroll down to the paragraph that begins with, "The first line is called the commit message."

Separation of Data and Code

By design, public GitHub repositories are not secure – they are publicly available. So, why do we use them? As discussed above, we use them because we want our research to be as transparent and reproducible as possible. Having said that, we have an obligation to balance reproducibility with the protection of the participants who make our research possible. Our best effort at striking that balance will be to upload almost everything except data to our GitHub repositories and make it publicly available. Other people will still be able to use the data, but they will need to follow certain procedures.

We believe the reproducible research approach is the most ethical and productive way to conduct research; however, it does create some additional complications for us. Namely, we

can't store the data in our code repository, but we still want to use relative file paths in our code (Please read [this chapter in R4Epi](#) if you aren't familiar with the difference between relative and absolute file paths). In other words, the file paths we use in the code should work on team member's computer without having to make any alterations to it. There are at least a couple of different ways we can accomplish this.

1. Create a **data** folder in the repository, but make sure to [gitignore](#) all of the data files in the data folder. a. We will then exchange data to add to the folder using a process that is separate from cloning the repository. For example, we will exchange the data using a shared OneDrive folder, you will copy the data from the shared OneDrive folder into the **data** folder in your local repository.
2. Another option that is sometimes available is to store the data in a remote database (e.g., FileMaker Pro). Then, we connect to the data using ODBC. a. When using this method, we have to make sure we don't add our database username and password to the code that will be publicly available on GitHub. Additionally, we need to make sure that the code we use to access the remote database is identical on every team member's computer. A credential storage application like [Keyring](#) helps us meet both of these needs.

What if we do accidentally upload data, PHI, or passwords? Don't freak out! Just let the PI know and we will fix it together.

Writing Programming Code

When writing programming code we generally will follow the guidance given in the [coding tools and best practices part of R4Epi](#). If you haven't already done so, please go ahead and read those chapters. In addition to reading about the coding style we will use, there are a couple of R packages that can help you style your code. They are [lintr](#) and [styler](#). We highly recommend that you give them a try.

Here are some additional guidelines that may not jump out at you.

- Use [Quarto](#) documents.
 - [Click here to read about Quarto files in R for Epidemiology](#).
 - We will almost always write our code in Quarto documents (as opposed to R scripts). However, we generally don't need to render them into HTML, Word, or PDF documents. The exception to this guideline includes:
 - * We will typically write functions in R scripts when they will be used in more than one Quarto file.
 - * We will write Shiny applications in R scripts.
 - * If the project we are working on is an R package (as opposed to a research project), all functions will be written in R scripts.
- Don't add dates or your name to code files. That is what versioning is for.

- **Do this:** 01_data_import.
- **Do not do this:** 01_data_import_mbc_edits, 01_data_import_v2, 01_data_import_2023_06_14.

Getting Started Task List

Now that you have some basic information about our team process for working on data management and analysis, here is a quick task list to get you started. It should be applicable for just about any project you are working on.

- Make sure you've read the most applicable sections of R4Epi. Please pay special attention to:
 - The [chapter on asking questions](#).
 - All of the [chapters in the coding tools and best practices part](#) of the book.
 - All of the [chapters on using git and GitHub](#).
- Make sure you have already been added to the IRB protocol.
- Download all necessary software to your computer (e.g., RStudio, git, FM pro driver, GitKraken).
- Create all necessary subscriptions (e.g., GitHub).
- Fork the GitHub repository to your GitHub account.
- Clone the GitHub repository to your computer.
- Create a test pull request.
- Make sure you can access any data needed for the project.
- Start looking through the tasks on the GitHub project board.