

Lab 0: Getting Started

FirstName LastName YourEmail@u.northwestern.edu

Due: TBD

1 Overview

This lab will help you set up R and RStudio and create a working directory for this class on your computer or the cloud. We will then connect your working directory to GitHub for version control.

2 Set up R and RStudio

We will use R and RStudio to work on lab assignments this quarter. If you have do not have them already, follow the instructions [here](#). If you have a Chromebook, check [here](#).

If your version of R is older than 4.4.x, then you should consider uninstalling it and starting with a fresh install. RStudio wil prompt you once a new version is available and give you instructions to update.

3 Prepare to create PDF files

For this course, you will submit assignments in `.pdf` format. This means we need to prepare R to render `.qmd` files into PDFs.

If you have previously succeeded at creating PDF files in your current version of R and RStudio, you may skip this step. If you have not done this before or have a fresh installation, then follow these steps:

1. Launch RStudio and open the `.qmd` version of this lab. The default name is `lab1.qmd`.

It is a good idea to not modify the naming convention for now.

2. To create PDF reports for assignments in this class, you need to have a version of \LaTeX installed. \LaTeX is a typesetting software, and we won't use it directly in this class, but R will call it under the hood to generate reports. We will use [TinyTex](#), an R-friendly \LaTeX distribution. Running the code below will install TinyTex. **You need to do this once after a fresh installation of R and RStudio.** You may skip this step in Posit Cloud

```
# Install the tinytex R package, only need to do this once
install.packages('tinytex')

# Use the package to install TinyTex within R
tinytex::install_tinytex()
```

3. At the top of the `.qmd` file, replace “FirstName LastName” with your preferred first and last name. For example, I would write “Gustavo Diaz” without the quotation marks
4. Replace “YourEmail@u.northwestern.edu” at the top with your NU email. For example, I would write “gustavo.diaz@northwestern.edu” (in my case, my email has a different extension)

4 Create a working directory

Now we will create a working directory that we will use to keep all of our the assignments for this course organized:

1. Make a folder for this class somewhere on your own computer or in Posit Cloud, preferably not on your desktop. For example, mine is `/Users/gustavodiaz/Projects/ps403` because this is local version of our course GitHub repository
2. In RStudio, go to **File > New Project** in the toolbar at the top. This should pop up a new window with options
3. Choose the *Existing Directory* option. Browse and select the folder that you just created. Then select *Create Project*. RStudio will reload and open the newly created project
4. Inside this new project folder, create a folder called `labs`
5. Save this `.qmd` file in the newly created `labs` folder
6. To confirm that it worked, run the following code to print your working directory, it will also show up in the PDF version:

```
getwd()
```

```
[1] "/Users/gustavodiaz/Projects/ps403/labs"
```

5 Set up GitHub

You now have a dedicated RStudio project for this course. This is enough to work on and submit assignments. However, we will also take the chance to establish good workflow habits early on by connecting to a GitHub repository.

The main advantage of GitHub is **version control**, which keeps track of every version of every file saved on your repository. For our purposes, this will be helpful when coding gets very messy and wish to revert to a previous working version of a file.

To use GitHub, you will need first need to create an account on <https://github.com>. I recommend you use your Northwestern email address so that you can benefit from the free [education benefits](#). You can, for example, get access to live AI coding assistance via Copilot.

Once your account is created, we need to associate it with your GitHub account. To do so, follow the instructions [here](#). Use the code chunk below to work on configuring GitHub for RStudio.

```
# Include your work here
```

6 Connect to a GitHub repository

After your account is created, we will connect our *local* RStudio project to a *remote* GitHub repository.

To do so, follow these steps:

6.1 Enable git in RStudio

1. With your project open in RStudio, navigate to **Tools > Project Options > Version Control**
2. Click the *SVN/Git* tab and select **git** as the version control system. It will ask you to restart RStudio to make the change
3. After RStudio reopens, confirm that there is a *Git* tab in the environment pane (upper right by default)

6.2 Create a GitHub repository

1. Go to your GitHub account and click the green button on the upper left-hand side to create a new repository. Choose whether to make it public or private (I recommend private if this is your first time)
2. **DO NOT** choose a template, nor add a README file, .gitignore, or license. You can include those later if you wish
3. Take note of the code chunk at the bottom after the message that goes “...or push an existing repository from the command line.” It looks something like this:

```
git remote add origin https://github.com/gustavo-diaz/test.git
git branch -M main
git push -u origin main
```

4. Back to RStudio, locate the *Terminal* tab. If you do not see it, navigate to **Tools > Terminal > New Terminal**
5. Type in or copy-paste the code in step 3 one line at a time and pass it to the terminal with the *Enter/return* key

6.3 Push files to GitHub

Your local project is now connected to the remote repository on GitHub! The final step is to *Push* your files into the repository:

1. Under the *Git* tab in RStudio, select the files you want to synchronize. I recommend right-clicking in the *.Rproj file and select *Ignore*, then select your *.qmd and .gitignore and press the *Commit* button

2. Type in a commit message, for example “first commit”, press *Commit* and close the new window

You are all set! All you need to remember now is to *commit* and *push* changes to GitHub often. If you work across multiple devices, you should also remember to *pull* at the beginning of every session.

We can talk more about using GitHub and version control for reproducible research during our course meetings.

7 Create a PDF report and submit

To make a PDF report, press the **Render** button at the top of the text editor in RStudio. This may take a while the first time you do it, and you may be prompted to install additional \LaTeX dependencies. When done, the PDF version will pop-up in a new window and the PDF file will be saved in the same folder as the `.qmd` version.

Once you are done, you can find a link to upload your lab on the [Canvas](#). This is how you will submit all the assignments for this course. Remember to also *commit* and *push* to GitHub!