

Workshop Technical Guide

Part 1: Managing Github Repositories

1. Getting started:

- Login to your account both on GitHub desktop and on the web browser.
- Go to the [HEGSRR](#) website and see if you could find the three repositories named as: [RPr-Chakraborty-2021](#), [UCGIS-Workshop](#), and [HEGSRR-Template](#)

2. Copy the three repositories to your personal GitHub account:

- Go to the [HEGSRR-Template](#) and click `Use this template`.
- You may change the `Repository name`, with the prefix of RPr for reproduction analysis and RE for replication analysis.
- Add a description to the repository if you like and set the repository to public.
- Go to the [RPr-Chakraborty-2021](#) repository and find the `Fork` option on the top right corner. You may leave the repository name as it is.
- Go to the [UCGIS-Workshop](#) and 'fork' this repository as well.

3. Clone the three repositories to your local computer:

- On your GitHub desktop, go to `File -> Clone Repository`, where you will see all of the repositories you have access to.
- Choose the [RPr-Chakraborty-2021](#) repository and choose the file path on your local computer where you want to save it to and hit `Clone`.
- Repeat the same process for the other two repositories you just forked.

4. Manipulating files on GitHub repositories:

- Go to the file directory on your local computer where you saved the repositories and see if you can find three folders, one for each cloned repository.
- On your GitHub desktop app, switch the `Current Repository` to [HEGSRR-Template](#) and click `Open in Atom`.
- Open the file named `readme.md`, make some changes to the file, and save it.
- Go to `Packages -> Markdown Preview -> Toggle Preview OR Toggle Github Style` to preview the file. Note that you might need to click `Toggle Break on Single New Line` before generating the previews. On Windows use the keyboard shortcut `ctrl + shift + m`
- Go back to GitHub desktop. A list of changed files and preview of the `diff` or differences should appear. At the bottom-left, provide a summary for the changes you have made and click `Commit to main` and then `Push` the repository.
- Go back to your own GitHub account page on the web browser and see if you can find the most recent updates you just committed.
- On the web browser, find the same `readme.md` file and click on the little pencil icon to the right.
- Make any changes you like and commit the changes directly on the web browser.
- Go back to GitHub desktop, and click on `Fetch origin` and then `Pull`.

5. More on file manipulation:

- Open the [HEGSRR-Template](#) repository in Atom and go to `data/raw/public`.
- Right click on the public folder -> `New File`.
- Write a line in the file and save it.
- Repeat the same process, but create the file in `data/raw/private` folder.
- Commit and push the changes.
- Go to the Github page, where you should be able to see the file you created only in the `public` folder but not the `private` folder.

- Go back to the Github desktop, click on `History` tab right next to `Changes`.
- Right click on the last change -> `Revert Changes in Commit` and `push` the changes.
- Check the GitHub page and your local computer to confirm that the last change has been reverted.

Part 2: Understanding the Workflow of Research Papers

1. Identify important information:

- Using Chakraborty's paper as an example, we have highlighted the *data sources* in blue
- *Methodology and data processing* steps in pink
- *Intermediary* (where we could check and compare results) as well as any **final results** in orange.

2. Generate a workflow diagram:

- Research for data sources as well as their metadata.
- Use different shapes and/or colors to differentiate between components, such as data sources and intermediary layers. Ovals for data layers and rectangles for processes are consistent with most graphic models in GIS.
- Use arrows and labels to indicate step progression based on the methodology section of the paper.
- Alternatively, you may cut figures out from the paper, write the steps on note cards, and arrange them accordingly.

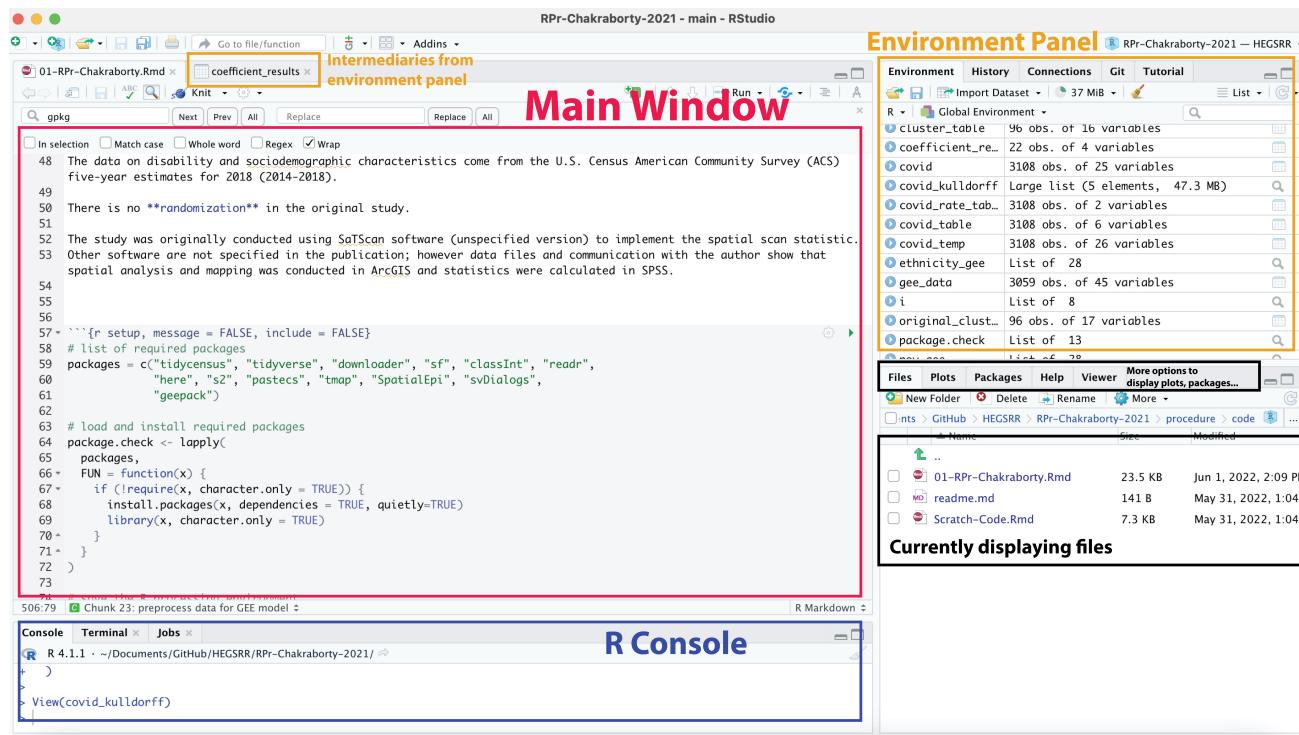
Part 3: Running the R Markdown

1. Opening the R Markdown:

- Go to the directory where you saved your GitHub repositories. Find the `RPr-Chakraborty-2021` repository and open the `R-project`.

2. RStudio orientation:

- The main window displays the R Markdown file
- The console allows you to write code directly in it and run.
- The environment panel displays all intermediaries created, which you could click on any of them to view.
- The lower right panel has a miscellaneous function of displaying files, plots, R packages installed, and allows you to search for any particular R function.



- In the `Files` panel, go to `Files` -> `RPr-Chakraborty-2021` -> `procedure` -> `code` -> `01-RPr-Chakraborty.Rmd`.

3. Basic elements of R Markdown:

The screenshot shows the RStudio interface with the following annotations:

- Text:** A blue box highlights the section starting with `## Query American Community Survey Data`.
- Code block:** A red box highlights the code block starting with ````{r Load ACS Data, message = FALSE, eval = FALSE}```.
- Comments:** An orange box highlights the comment `# Query disability demographic data with geographic boundaries`.
- Add a code block here:** A tooltip near the Run button says "Add a code block here".
- Alternatively, select the lines and run the code here:** A tooltip near the Run button says "Alternatively, select the lines and run the code here".
- Run the code here:** A tooltip near the Run button says "Run the code here".

The R Markdown code includes:

```

506:79 | 01-RPr-Chakraborty.Rmd | coefficient_results | R Markdown
  1 01-RPr-Chakraborty.Rmd x coefficient_results x
  2
  3 In selection   Match case   Whole word   Regex   Wrap
  48 The data on disability and sociodemographic characteristics come from the U.S. Census American Community Survey (ACS)
  49 five-year estimates for 2018 (2014–2018).
  50 There is no **randomization** in the original study.
  51
  52 The study was originally conducted using SaTScan software (unspecified version) to implement the spatial scan statistic.
  53 Other software are not specified in the publication; however data files and communication with the author show that
  54 spatial analysis and mapping was conducted in ArcGIS and statistics were calculated in SPSS.
  55
  56
  57 ````{r setup, message = FALSE, include = FALSE}
  58 # list of required packages
  59 packages = c("tidycensus", "tidyverse", "downloader", "sf", "classInt", "readr",
  60             "here", "s2", "pastecs", "tmap", "SpatialEpi", "svDialogs",
  61             "geopack")
  62
  63 # load and install required packages
  64 package.check <- lapply(
  65   packages,
  66   FUN = function(x) {
  67     if (!require(x, character.only = TRUE)) {
  68       install.packages(x, dependencies = TRUE, quietly=TRUE)
  69       library(x, character.only = TRUE)
  70     }
  71   }
  72 )
  73
  74 # save the R processing environment
  75 writeLines(capture.output(sessionInfo()),
  76             here("procedure", "environment", "r_environment.txt"))
  77 ...
  78
  79 ## Query American Community Survey Data
  80
  81 This will require an API key for the census, which can be acquired easily here:
  82 [api.census.gov/data/key_signup.html](https://api.census.gov/data/key_signup.html)
  83 This query can take some time to run...
  84
  85 ````{r Load ACS Data, message = FALSE, eval = FALSE}
  86 # get API Key
  87 # we could store this in the raw/private or scratch folder and load if the
  88 # researcher has already entered it once
  89 census_api_key(dlgInput("Enter a Census API Key",
  90   Sys.getenv("CENSUS_API_KEY"))$res,
  91   overwrite = TRUE)
  92
  93 # Query disability demographic data with geographic boundaries
  94 acs <- get_acs(geography = "county",
  95   table = "S1810",
  96   year = 2018,
  97   output = "wide",
  98   cache_table = TRUE,
  99   geometry = TRUE,
  100  keep_geo_vars = TRUE)
  
```

- Comments begin with a hashtag `#`
- If a code block produces outputs, they appear below the code block
- Data frames and objects appear in the `Environment` panel
- Some researchers prefer to connect to GitHub directly from RStudio.