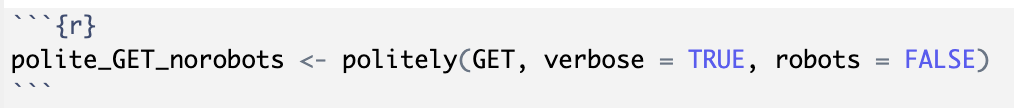
**API User Guide**

This document will teach you how to successfully create a rectangular data table through an API for Refuge Restrooms, a project designed to inform the public on safe restrooms for transgender, intersex, and non-binary people throughout the U.S. and other areas of the world. If you have any questions, feel free to reach out to me at [mmburton5@gmail.com](mailto:mmburton5@gmail.com).

1. Under “[Transportation Public APIs](https://github.com/public-apis/public-apis?tab=readme-ov-file#transportation)”, there should be a link titled “[REFUGE Restrooms](https://www.refugerestrooms.org/api/docs/#!/restrooms)”. You can access the link provided here but check that Auth is “No” and “HTTPS” is “Yes” on the GitHub site first.
2. A computer code with text

   Description automatically generated with medium confidenceA screenshot of a computer program

   Description automatically generatedNext, open up a new RMarkdown file in RStudio. Make sure the following code chunk is in the file:
3. You’ll want to load in the R packages needed to properly execute this assignment. Here are the ones I used:
4. Next, using the polite package, you’ll create a function called polite\_GET\_norobots. This function will help you get specifically HTTP data from websites while respecting the website’s API rules (why we use politely).
5. A close-up of a web page

   Description automatically generatedOnce you’ve set up this function, we can use it to properly get website data from the public API. For Refuge Restrooms, their base URL is [www.refugerestrooms.org/api](http://www.refugerestrooms.org/api), but since we want to specifically get the 100 most recent datapoints for safe restrooms, you can manually add on “/v1/restrooms?per\_page=100” (?per\_page=100 is a query to specify our parameters). We’ll save this as a list called “response”.
6. A close-up of text

   Description automatically generatedBefore you can get this as a dataset, you need to specify the list “response” as text and make sure it’s in JSON format. Then, you can use the function “fromJSON” to transform our data into a dataset called “refuge\_restrooms”.
7. Almost done! Next, you’ll need to do a bit of data cleaning. Because these data points are user reported, some people choose to abbreviate a state, while others write out the full name. To keep things consistent, you’ll need to manually change the abbreviations to full names using the “recode” function under “mutate”. Because the dataset changes with A close-up of a computer screen

   Description automatically generatedevery new user submission, the abbreviations I changed might not be the same as yours. Here’s an example:
8. A white background with green text

   Description automatically generatedFinally, you’ll want to save your dataset as a .csv file so the data points won’t change. You’ll use the function “str” to string the data, create an object for your name, and use “write\_csv” as shown here:
9. That should be all you need to start analyzing your data and creating visualizations! Again, feel free to contact me if you have any further concerns, or anything else you think I should’ve included in this user guide (email listed above).