NATIONAL ENDOWMENT FOR THE HUMANITIES

Draper, Jessica; Christoffersen, Katherine; Bessett, Ryan; & Carvalho, Ana. 2021.
Bilingual Voices in the U.S./Mexico Borderlands. National Endowment of the Humanities Award.
University of Texas Rio Grande Valley and University of Arizona.

Revising .txt Transcripts with R

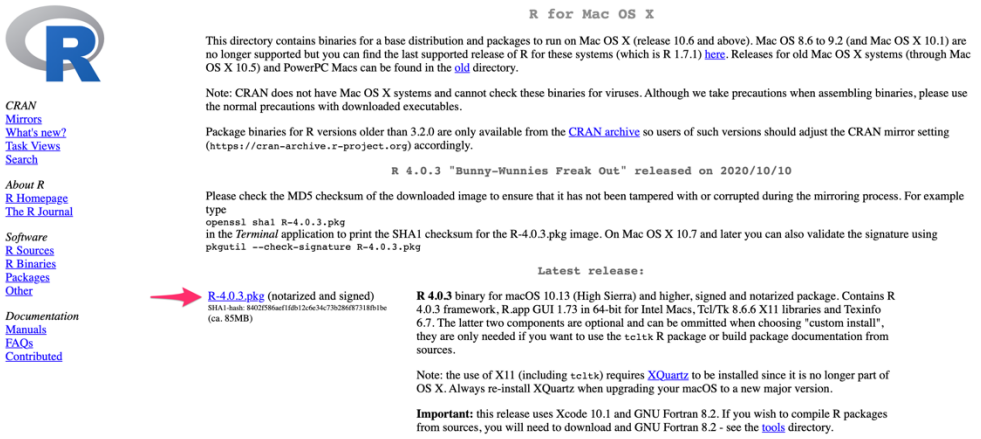
Step 1 & 2 (March 2021)

R is a free software environment and programming language for statistical computing and data analysis. It is also great for data manipulation, including regular expressions for identifying, extracting, and replacing patterns in texts – a convenient tool for revising transcripts!

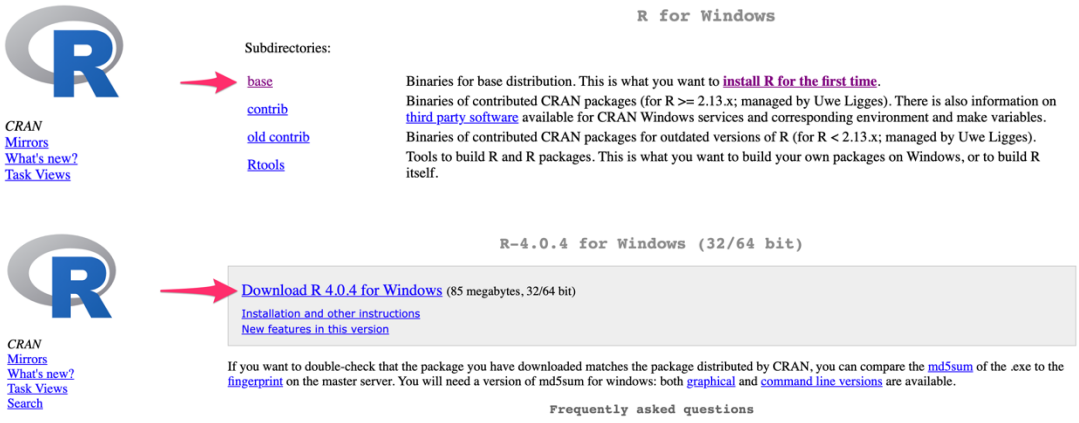
Setup

To begin, you will need to download two things: **R** and **RStudio**.

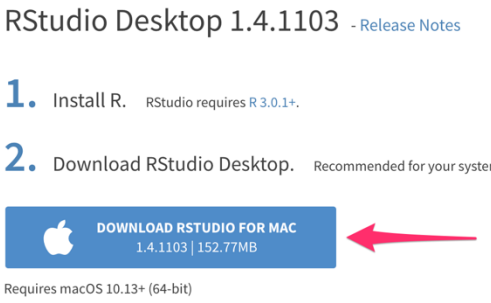
1. **Download R** for your system; open the file and install
    - **Mac:** click the .pkg file link under "Latest release"



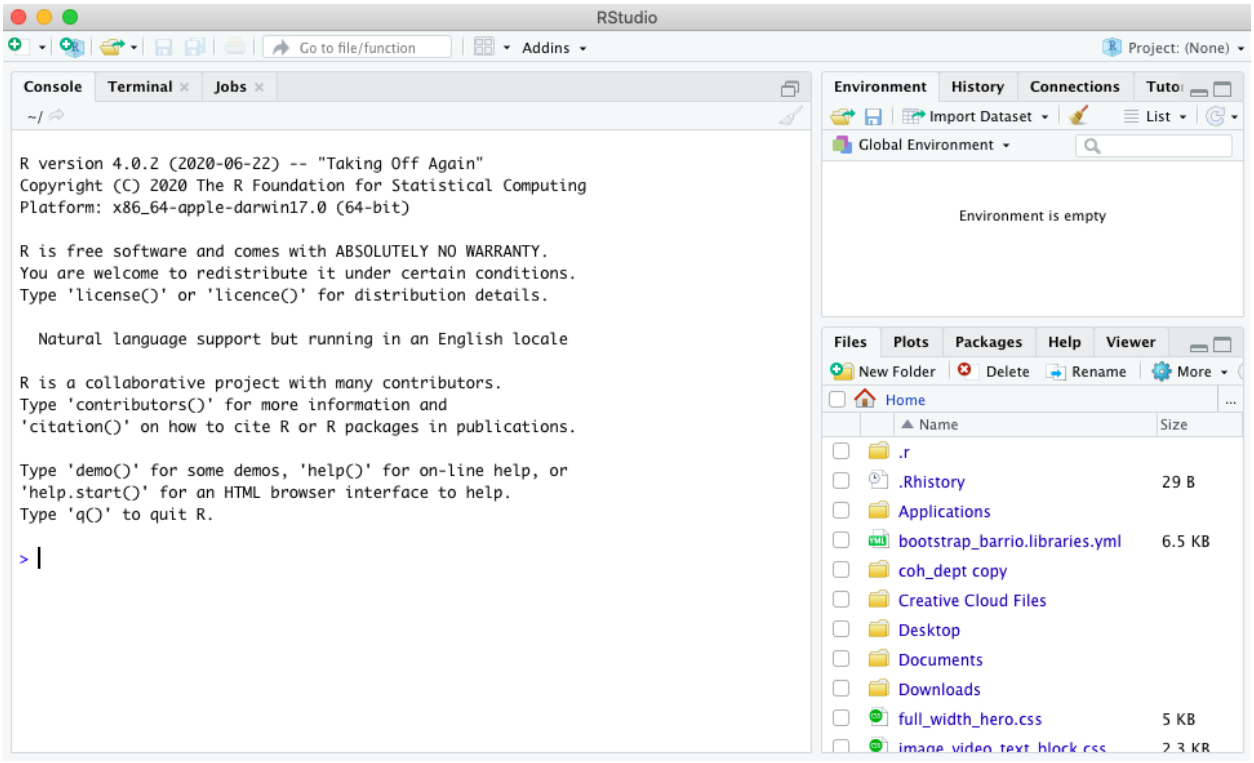
    - **Windows:** click the "base" link then click the download link on the subsequent page



2. Once R is installed, **download RStudio** (Mac or Windows); open the file and install



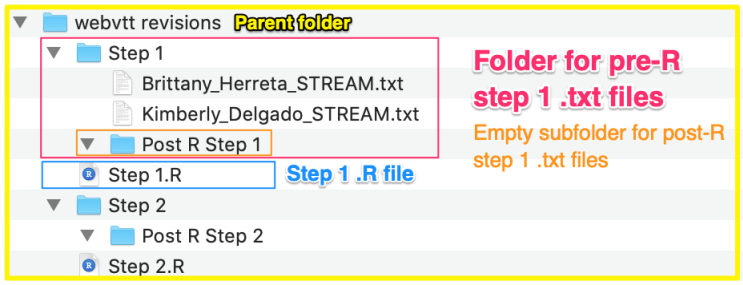
3. Once **RStudio** is installed, try to open the application. It should look something like this:



4. Before running any scripts, organize your transcript and .R files in your file directory in the following way:

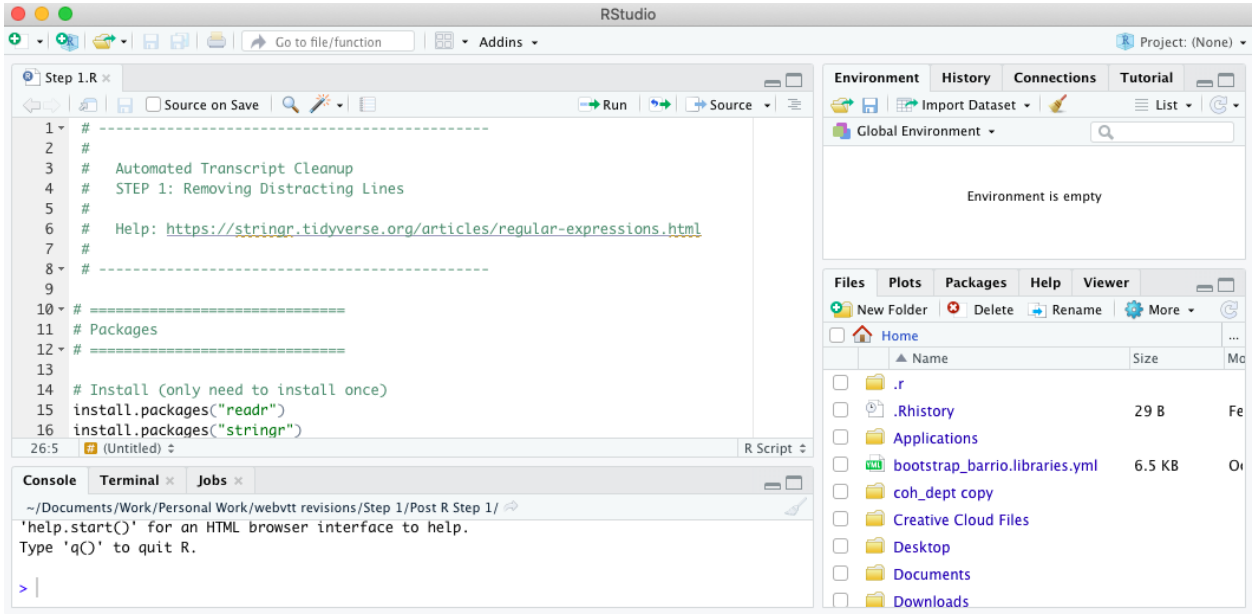
   - .R and .txt files live in the same parent folder (e.g. “Documents/webvtt revisions” or along these lines)
   - The .txt files live in their own “Step 1” folder
   - The “Step 1” folder has an empty subfolder (e.g. “Post R Step 1” or “After Step 1”) for saving post-R revised files
   - This structure is used for Step 2 as well

In the end, the folder structure should look something like:



Running Step 1.R

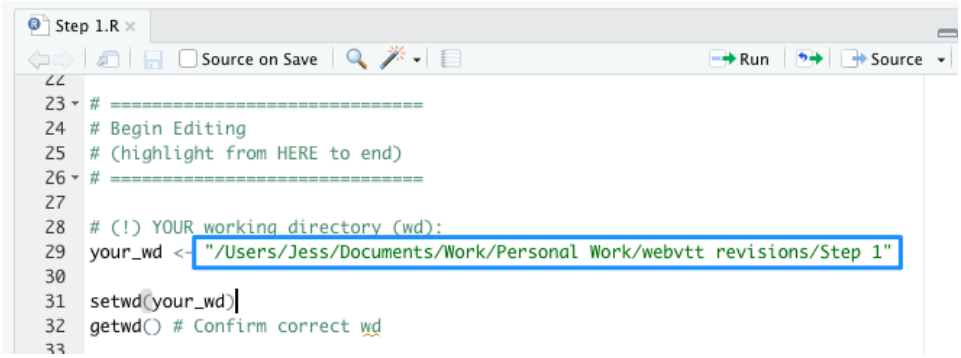
1. Once the .txt files have been added to the Step 1 folder, open **Step 1.R** in RStudio. It should look something like:



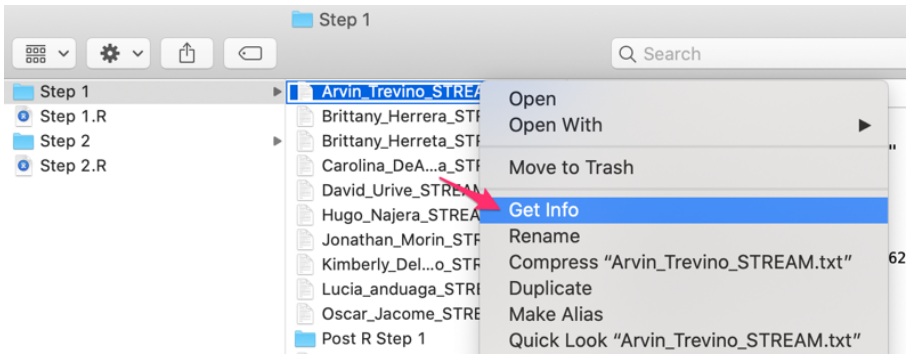
2. What you see in the Step 1.R file is the R programming language. Green text indicates comments, not programming code. Comments are skipped over when the code is run. They are created using the preceding "#"

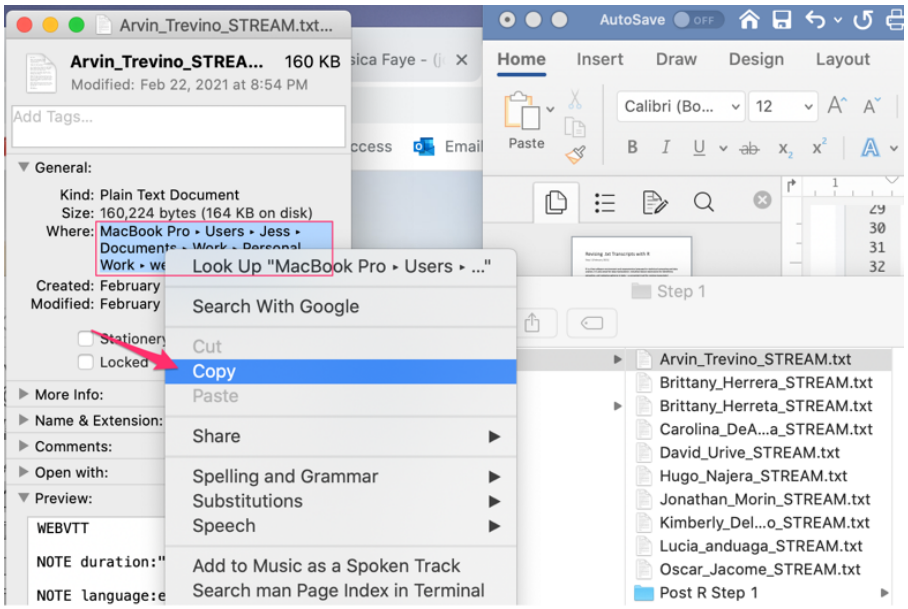
3. Before running this code:

a. Update the "working directory" path on line 29 to reflect your local file location where your Step 1 .txt files are stored *(path should end with "Step 1")*

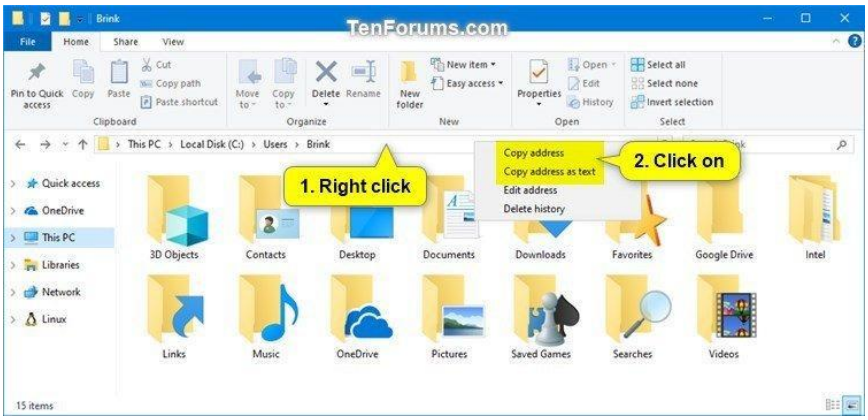


i. To find this location on a **Mac**, right-click on a .txt file within the Step 1 folder and select "Get Info." Then highlight the "Where" text and click copy. Then, paste between the quotation marks in the R script on line 29.





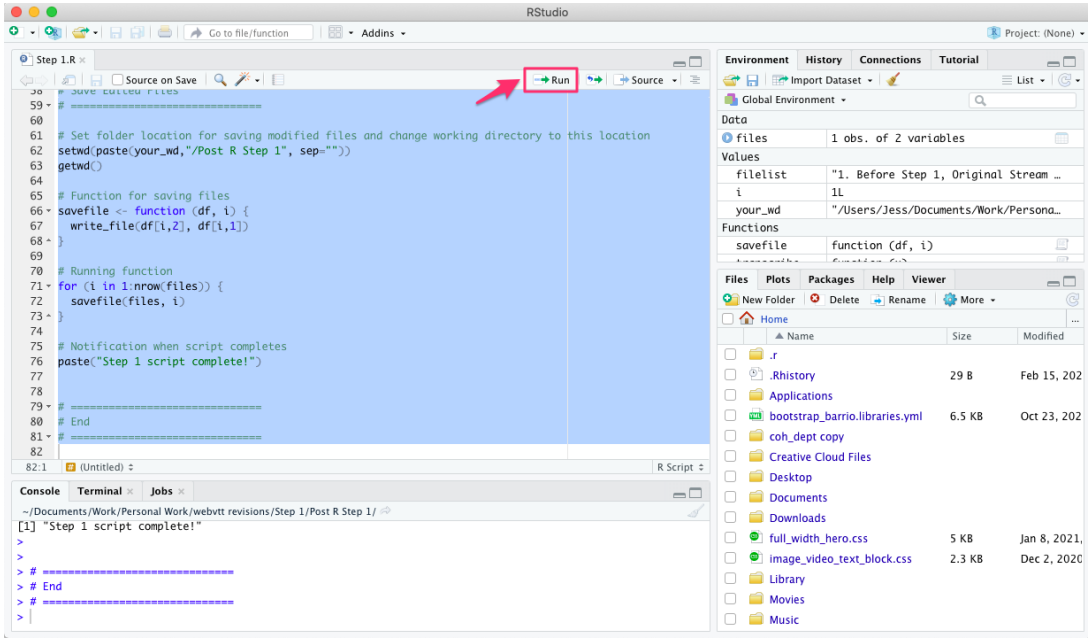
ii. To find this location in **Windows**, open up the Step 1 folder in File Explorer. In the address bar above, right click and select "Copy address" OR press shift and then right click on a .txt file and select "Copy as path." Paste between the quotation marks in the R script on line 29. Then, **change all backslashes (\) to forward slashes (/).**



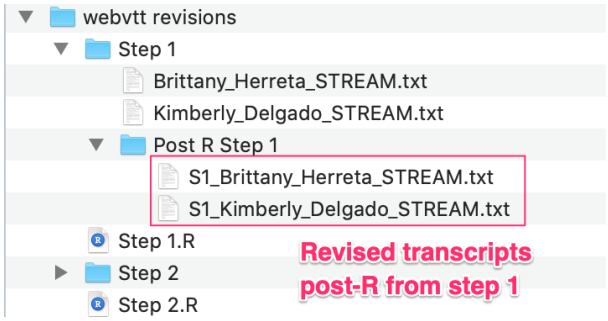
b. On line 62, the code uses your working directory location and adds a subfolder to indicate where the new revised transcripts should be saved. Make sure this subfolder name matches the one you created earlier (e.g. "Post R Step 1")

```
56
57 # ============================
58 # Save Edited Files
59 # ============================
60
61 # Set folder location for saving modified files and change working directory to this lo
62 setwd(paste(your_wd,"/Post R Step 1", sep=""))
63 getwd()
64
```

4. Now you are ready to run the R script! Highlight the entire code and click "Run" in the top right. You'll know it's finished when you see "End" in the bottom console area.



5. Now open the "Post R Step 1" folder from your file directory. It should now include all .txt files from its parent folder but renamed with the prefix "S1_". *When you open these files, the "NOTE Confidence" lines and random letters/numbers should be removed.*



6. Save the .R file with your updated working directory information (File > Save). You won't have to update the file directories again for this step.

    a. Optionally, before saving, you can comment out lines 15-16 by adding a "#" before the line, making it green. This means the code will skip over the install commands next time the code is run, shortening the time it takes to run the

script. Once these packages have been installed the first time, you typically don't need to install them again (unless you get an error telling you otherwise). Going forward, you only need to highlight from line 23 down before running the code.

```
10 # ============================
11 # Packages
12 # ============================
13
14 # Install (only need to install once)
15 install.packages("readr")
16 install.packages("stringr")
17
```

```
10 # ============================
11 # Packages
12 # ============================
13
14 # Install (only need to install once)
15 #install.packages("readr")
16 #install.packages("stringr")
17
```

Commented out, no longer part of code

7. Repeat this process as necessary!

Running Step 2.R

1. Once the speaker tags have been added to the .txt files, add them to the Step 2 folder.
    a. Be sure to review these text files to make sure there are no obvious errors or issues; the code should catch most formatting issues, but there may be some it could miss if it's not setup in this general format:

```
00:00:10.050 --> 00:00:10.920
<v INV> Example text of a single line

00:00:12.980 --> 00:00:17.930
<v PAR> Example text of a line
with a break. This is okay, the code
will fix this to remove line breaks.

00:00:17.930 --> 00:00:22.880
continuing example text from the line above; the code
will collapse timestamps within a speaker's dialogue
```

2. Open **Step 1.R** in RStudio
3. **Before running this code:**
    a. Just like in Step 1, update the "working directory" path on line 31 to reflect your local file location where your Step 2 .txt files are stored *(path should end with "Step 2").* See Step 1 again for more detail on how to find the path.

b. On line 162, the code uses your working directory location and adds a subfolder to indicate where the new revised **Step 2** transcripts should be saved. Make sure this subfolder name matches the one you created earlier (e.g. "Post R **Step 2**")

4. Now you are ready to run the R script! Highlight the entire code and click "Run" in the top right. You'll know it's finished when you see "End" in the bottom console area.

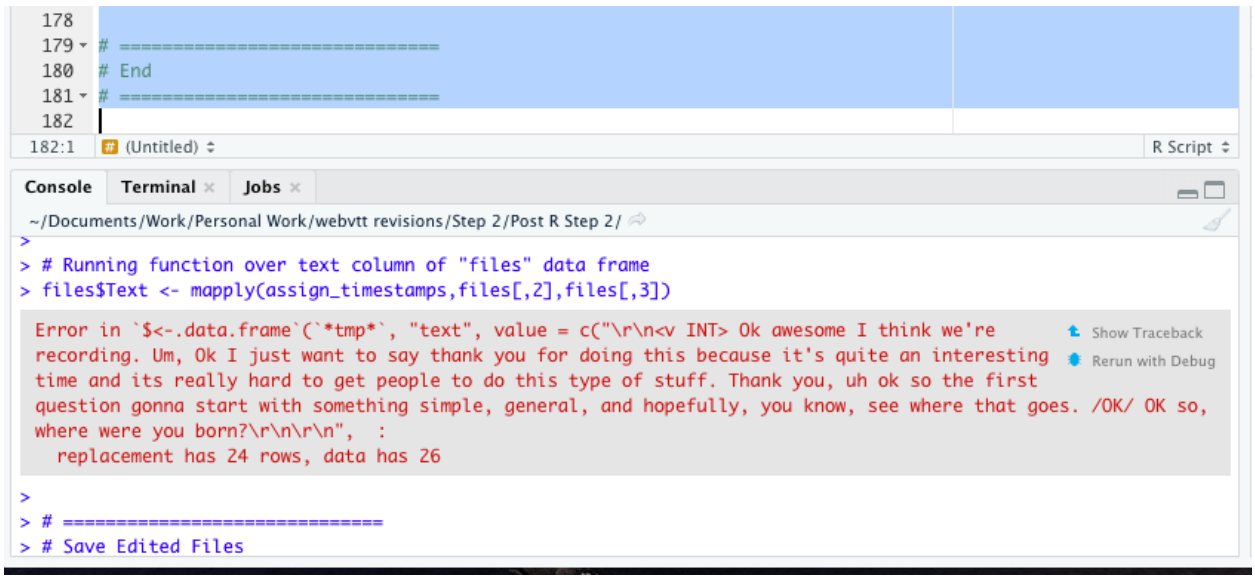
5. Now open the "Post R Step 2" folder from your file directory. It should now include all .txt files from its parent folder but renamed with the prefix "S2_".

6. **When you open these S2_.txt files, speaker dialogues should be collapsed within a single timestamp, alternating between speakers and the text should wrap; no "00:00:000" timestamps should exist.**

   If you see a problem, review the original .txt files to make sure there is nothing in there that might be breaking the code from running smoothly.

   For example, if you do see "00:00:000" timestamps throughout the text in the resulting S2_ .txt files, this means there is a formatting issue in at least one of the original .txt files that the code ran through in the Step 2 folder. If even just one file has a formatting issue, it will break the timestamp function code for all files in the Step 2 folder.

   To find the file with the formatting issue, after running the code in R, scroll up in the bottom console area until you see `red error text`. It will not tell you specifically which file the error is in, but it should include a partial text sample that you can use as a clue to find the .txt file that is causing trouble.



Once you've identified the correct .txt file, review the entire file for any formatting issues. It will likely be related to how the timestamps are formatted (e.g. an extra "-->" or odd spacing or characters that deviate from the desired pattern). When found, fix the issue(s) manually, save the file, and run the R script again. Check the console for any further errors and, if present, repeat this process again.

7. When finished running the code, save the .R file with your updated working directory information (File > Save). You won't have to update the file directories again for this step.

    a. Optionally, just like in Step 1, before saving, you can comment out lines 15-17 by adding a "#" before the line, making it green. This means the code will skip over the install commands next time the code is run, shortening the time it takes to run the script.

8. Repeat this process as necessary!

NATIONAL ENDOWMENT FOR THE HUMANITIES

This project has been made possible in part due to funding from the National Endowment of the Humanities. Exploring the Human Endeavor.
http://www.neh.gov