STAT495 - Homework

Shukry Zablah and Emily Ye

2019-10-23

The Dickinson Search Engine

Summary

Our app includes the original function of picking a random poem and expands it by including a reactive search bar that allows for users to search for a word or phrase and view the lines of Emily Dickinson's poems matching the search term(s) and their respective titles.

For example, if the user searches for the word "love", our app will be able to read in, match, extract, and display all the lines from every poem that has the word in it! This way users can search for phrases from poems they like but are not getting results in a normal search engine.

Learning Experiences

For Emily, the following functions were new:

- data table outputs in Shiny (Using DT in Shiny, 2019)
- `stringr` package and its functions (`str_detect`, `regex`)

For Shukry, the following functions things were new:

- `readLines`: We spend so much time in Shiny that I had to look this up!
- `fs::dir_ls`: The `fs` package aims to make managing directory data a tidy process. This function is similar to the base one but this interfaces nicely with other functions in the package that show more information from the files if needed.

Aspirations

Among the things that we wanted to do but we didn't have time for there are two that were closest to becoming reality:

1) Clicking the table row and having access to the file that corresponded to the row clicked.
2) Search by other attributes (e.g. mood) to get results. This keeps in line with the search engine theme.

References

"Using DT in Shiny." 2019. Retrieved from https://rstudio.github.io/DT/shiny.html.