teea_session_6

N.Paterno

8/5/2021

For this session we went over paths, directories, projects, Git and GitHub. Instead our typical recap, I'll walk through the definition of each piece and try to use analogies to make it more relatable. As we discussed via text, we're going to redo this lesson next week (on 8/13) at no cost.

Paths, Directories and Projects

A *path* is the digital address for a file. To relate it to transportation, we'll think of this as a bus stop.

A *directory* is a fancy name for a folder. The *working directory* is the folder that RStudio is currently reading files from. You can determine your current directory with

```
getwd()
```

If a bus stop is the *path*, then a *directory* is analagous a bus and a *working directory* is the bus we are riding. If we tell RStudio to look for a file (usually to import data) that is not in the current working directory, it will throw an error that says the file doesn't exist. For simplicity let's say we have two bus routes: red line and blue line. If we are on the blue bus, we can't get off at a red bus stop.

The best way to make sure that we're in the right directory (on the right bus) is to use a project. RStudio uses projects to explicitly label the working directory.

Without a project, its like the busses are not labeled. We would get on a bus, assume its the one we want (the red line) but when we ask the driver for our stop, he tells us we're on the wrong bus. Using a project is analagous to painting the red line bus red and the blue line bus blue. So we'll know before we even board what stops we can get off at. This means the *project* is like a bus route number.

The benefit to this is that as long as we open the project, RStudio will be looking in the right place for our files.

Git & GitHub

Git and GitHub are used for backup and colaboration.

Git is version control software. It exists separate from R and RStudio. We'll actually spend some time on Friday working with Git outside of RStudio. I think getting a grasp on that outside will make it easier to integrate into your workflow in RStudio. To use it on a specific folder, we initialize a Git repository for the folder. The main commands for Git are *commit*, *push* and *pull*.

The *commit* command is for when we want to save a snapshot of our folder. This creates a full copy of our folder and compresses it (to save storage space on your computer). Once we've completed a task, we can *commit* the file changes before moving on to the next task. That way if we break our code, we can revert back to the previous commit. Even if we save a new file, we can revert back to the old commit.

The *push* and *pull* commands are for working with GitHub. GitHub creates an online (cloud) back up of your local repository. *Push* will move commits from your computer to the cloud. *Pull* will move changes made online down from the cloud to your computer.

Using *GitHub* also makes it possible to share the same project (the same bus) with colaborators (other passengers). A colaborator can make a *pull request* letting the repository owner know they'd like to add a file; this would be like a passenger requesting a new bus stop. As long as it doesn't create any conflicts, the owner can *merge* the request (the county/city can add a new bus stop).