**Overall Theme: What is GitHub?**

*What does GitHub do generally?*

*What industries use GitHub the most?*

*Can GitHub be used with other major tools (e.g. Python)?*

*Why exactly is GitHub commonly used in the business world?*

Simply put, GitHub is a place to keep versions of documents. What makes it different from Dropbox, Box, GDrive, and other file storage providers is that it is specifically designed to keep versions of code. The code could be written in any programming language (R, Python, Java, C#, C++, JavaScript, and more). Git is the protocol for keeping track of the versions (e.g., actions to push new changes, compare changes, replay other collaborators’ changes on top of your work). Think of this as a more sophisticated and specialized "Track Changes" tool that Microsoft Word has. After tracking changes, you can push the documents up to GitHub so that many people can have access. So instead of emailing code back and forth, like a Word document, people just push it to one place: GitHub. Because GitHub is a site to keep versions of code, it's naturally adopted by industries that do software engineering. Facebook, Google, and Amazon leap to mind, but there are industrial companies, retailers, and many more companies that create and maintain software for their business, which requires software engineers to keep code in GitHub.

**Overall Theme: What are the basics I need to know?**

*What are some useful commands to know in Git Bash?*

*How do you know when to work in the shell and when to work in RStudio?*

*How can I remove a commit on GitHub?*

*If a commit has been deleted, how do you recover it?*

*How do you merge branches?*

*What are the most common tasks we will use GitHub for in our jobs?*

RStudio has an interface (the "Git" pane) for performing actions without having to write any code (e.g., checkout repository, stage commits, commit, push, switch branches, look at file diffs). These are the most common and most important actions that you need to know how to perform. Each time you click a button in the interface, understand that there is a command in git that is actually responsible for performing that action for you. RStudio is just giving you a pretty button to click. For this reason, you never really need to use the shell; however, it is really good practice to perform the operations from the shell so that you understand the general order of operations and can later do more complicated things like fetch changes, fix merge conflicts, merge branches, rebase against another repository, etc. All of the exciting and more complicated things are not available in the RStudio interface. For this reason, it is great to start in RStudio to learn, but explore from the shell to continue learning if you really intend to collaborate effectively. Interestingly, RStudio doesn't contain a button for the very simple task of deleting a commit. GitHub was designed to keep track of versions of code and files, so it goes against the purpose of the entire system to remove history because keeping the history is what is important. You cannot recover code from a commit after it has been deleted, so be careful when doing this.