Task of week3

There is not a lot to do here other than what is asked to follow as below:

# Git Repository

### **Start using Github to manage course materials**

# Reading

* Chapters [4-5 in R4DS](http://r4ds.had.co.nz/)
* Chapters [4-8 in Happy Git and Github for the useR - Installation](http://happygitwithr.com/installation-pain.html)
* Chapters [13-15 in Happy Git and Github for the useR - Installation](http://happygitwithr.com/)

# Tasks

* RStudio and [the Git GUI](https://www.youtube.com/watch?v=E2d91v1Twcc)
* Install git on your computer
* Configure Git
* Make sure git works in R-Studio
* Optionally sign up for the [GitHub Education pack](https://education.github.com/pack)
* Click on [this link to access the course repository](https://classroom.github.com/a/aARrUulU)
* Create a new project in Rstudio and connect it to the new repository in GitHub. Helpful instructions are [here](http://happygitwithr.com/rstudio-git-github.html#clone-the-new-github-repository-to-your-computer-via-rstudio)
* Edit the README.md file in your repository to include a brief description of the repository (e.g. “Coursework for Spatial Data Science”).
* Stage and Commit your changes to Git (using the git tab in the upper right of RStudio)
* Introduce yourself to git following [these instructions](https://happygitwithr.com/hello-git.html)
* Push the repository up to GitHub
* Confirm that the changes are visible on your github page

# Background

Version control systems (VCS) allow developers to maintain a record of how their code has changed over time. When used properly, a VCS can help a developer track down the exact point in time when a bug was introduced or fixed, easily undo changes, and collaborate with other developers.

There are many types of version control systems. Some of the more popular ones include CVS, subversion, mercurial, and git. In recent years, git has quickly become the most popular of the group.

## **Git**

Git stores files in a type of database called a repository or repo. Most data science teams that work with git keep a central repository on a server somewhere that everyone on the team can access. This repository stores the files and the history of every change made to each file, including who made the changes and when those changes were made.

Git works with groups of changes called commits. A single commit might have many changes associated with it. Those changes might include updates to, existing files, the addition of new files, or the removal of files.

## **Example**