Project Management With RStudio

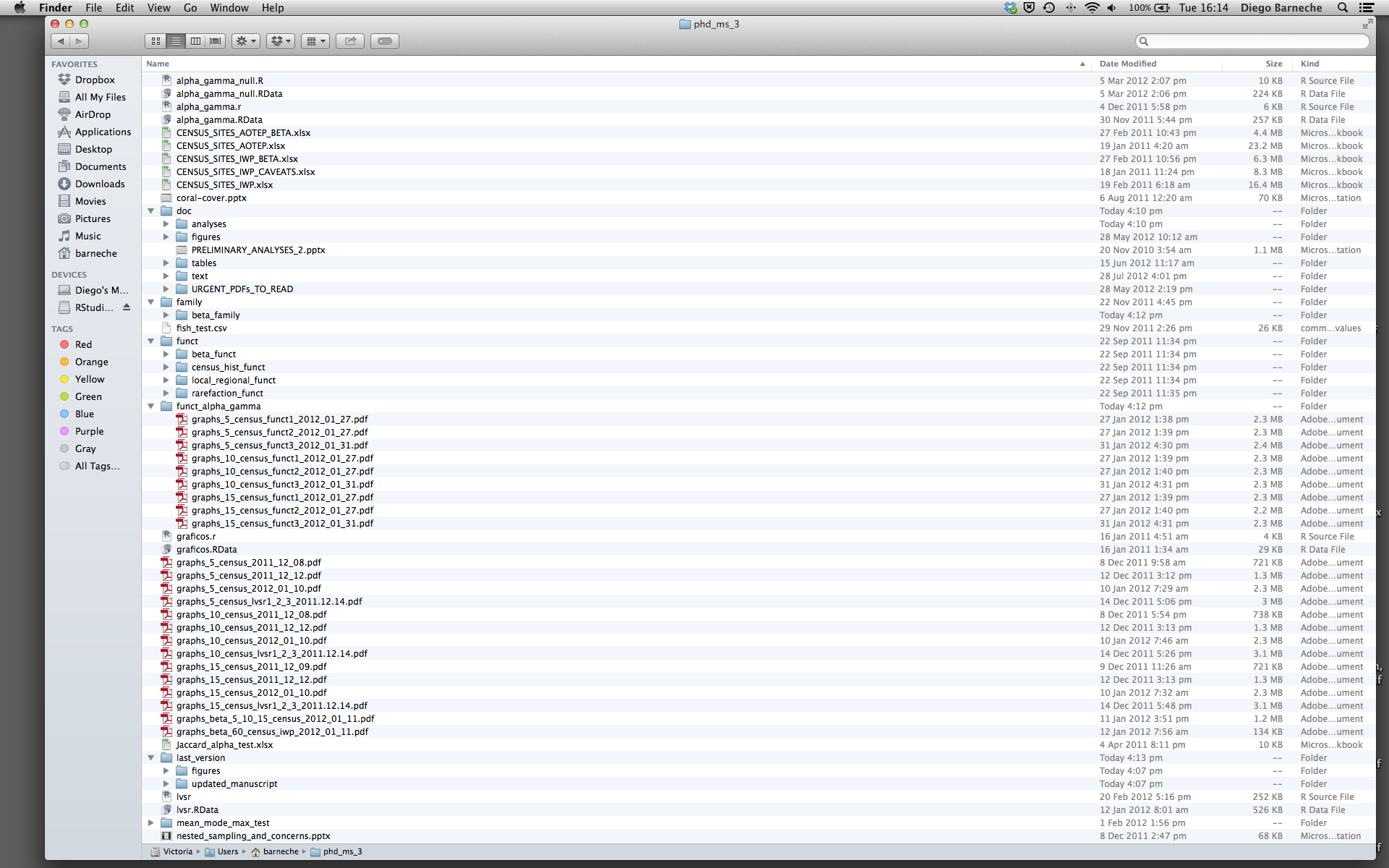
## Introduction

The scientific process is naturally incremental, and many projects start life as random notes, some code, then a manuscript, and eventually everything is a bit mixed together.

Managing your projects in a reproducible fashion doesn’t just make your science reproducible, it makes your life easier.

— Vince Buffalo (@vsbuffalo) April 15, 2013

Most people tend to organize their projects like this:



There are many reasons why we should *ALWAYS* avoid this:

1. It is really hard to tell which version of your data is the original and which is the modified;
2. It gets really messy because it mixes files with various extensions together;
3. It probably takes you a lot of time to actually find things, and relate the correct figures to the exact code that has been used to generate it;

A good project layout will ultimately make your life easier:

* It will help ensure the integrity of your data;
* It makes it simpler to share your code with someone else (a lab-mate, collaborator, or supervisor);
* It allows you to easily upload your code with your manuscript submission;
* It makes it easier to pick the project back up after a break.

## A possible solution

Fortunately, there are tools and packages which can help you manage your work effectively.

One of the most powerful and useful aspects of RStudio is its project management functionality. We’ll be using this today to create a self-contained, reproducible project.

## Challenge: Creating a self-contained project

We’re going to create a new project in RStudio:

First, we are going to download the course content. Download the zipfile from: <https://github.com/bcgov/intro-to-r.git> and save it somewhere where you can find it (e.g., your desktop), then unzip it. You should have a folder called intro-to-r with a README.md file and a data folder.

1. Click the “File” menu button, then “New Project”.
2. Click “Existing Directory”.
3. Browse to the folder we just extracted.
4. Click the “Create Project” button. Now when we open this project with RStudio, all of our work on this project will be entirely self-contained in this directory.

## Best practices for project organization

Although there is no “best” way to lay out a project, there are some general principles to adhere to that will make project management easier:

### Treat data as read only

This is probably the most important goal of setting up a project. Data is typically time consuming and/or expensive to collect. Working with them interactively (e.g., in Excel) where they can be modified means you are never sure of where the data came from, or how it has been modified since collection. It is therefore a good idea to treat your data as “read-only”.

### Data Cleaning

In many cases your data will be “dirty”: it will need significant pre-processing to get into a format R (or any other programming language) will find useful. This task is sometimes called “data munging”. Storing these scripts in a separate folder, and creating a second “read-only” data folder to hold the “cleaned” data sets can prevent confusion between the two sets.

### Treat generated output as disposable

Anything generated by your scripts should be treated as disposable: it should all be able to be regenerated from your scripts.

There are lots of different ways to manage this output. Having an output folder with different sub-directories for each separate analysis makes it easier later. Since many analyses are exploratory and don’t end up being used in the final project, and some of the analyses get shared between projects.

## Tip: Good Enough Practices for Scientific Computing

[Good Enough Practices for Scientific Computing](https://github.com/swcarpentry/good-enough-practices-in-scientific-computing/blob/gh-pages/good-enough-practices-for-scientific-computing.pdf) gives the following recommendations for project organization:

1. Put each project in its own directory, which is named after the project.
2. Put text documents associated with the project in the doc directory.
3. Put raw data and metadata in the data directory, and files generated during cleanup and analysis in a results directory.
4. Put source for the project’s scripts and programs in the src directory, and programs brought in from elsewhere or compiled locally in the bin directory.
5. Name all files to reflect their content or function.

### Separate function definition and application

One of the more effective ways to work with R is to start by writing the code you want to run directly in an .R script, and then running the selected lines (either using the keyboard shortcuts in RStudio or clicking the “Run” button) in the interactive R console.

When your project is in its early stages, the initial .R script file usually contains many lines of directly executed code. As it matures, reusable chunks get pulled into their own functions. It’s a good idea to separate these functions into two separate folders; one to store useful functions that you’ll reuse across analyses and projects, and one to store the analysis scripts.

### Version Control

It is important to use version control with projects. Go [here for a good lesson which describes using Git with RStudio](https://swcarpentry.github.io/git-novice/14-supplemental-rstudio/).