



Marine Ecological Modelling Global Climate Change

Dissemination of results under the Open Science framework

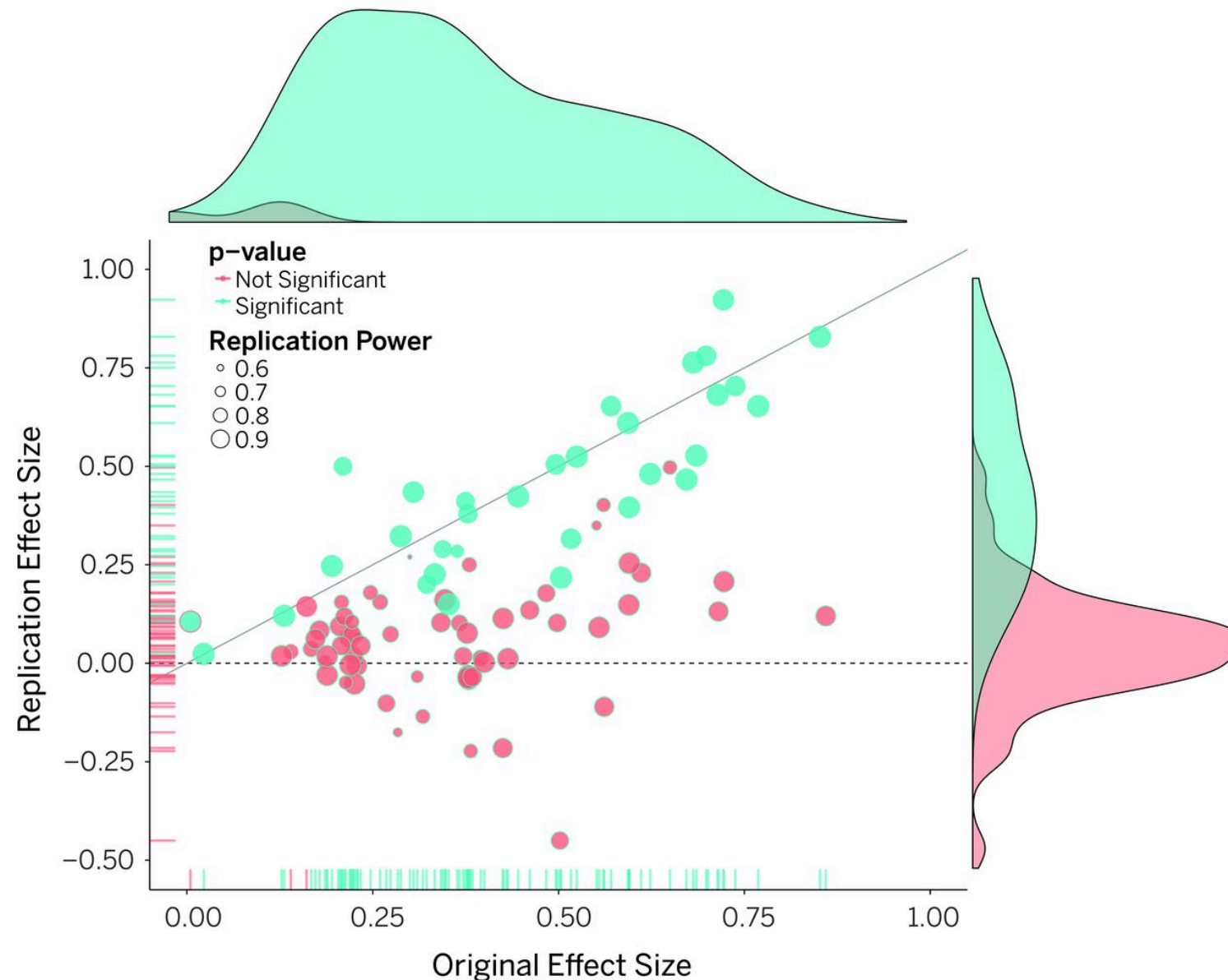
Jorge Assis, PhD // jmassis@ualg.pt // jorgemfa.medium.com
2020, Centre of Marine Sciences, University of Algarve



Open Science

Most scientists spend much time thinking about the types of data they need for their studies. Relatively **little effort is spent** considering **how to store, analyze** and **share their data**.

It is increasingly important to store and document scientific data in ways that facilitate: (i) Open Science; and (ii) their effective retrieval and interpretation in the future.



A replication of 100 studies show “97% of original studies had significant results ... 36% of replications had significant results”. (abstract). If no bias is assumed in the original results, there is little potential for replication (lack of open science).

** Diagonal line represents replication effect size equal to original effect size.



Open Science

Sharing information is ideal in science, but the reality looks like this:

A scientist **collects data** and **stores it on a machine**.

He writes or modifies a few small programs (which also reside on his machine) to analyze that data.

With results, he writes and submits a paper. He **might include data** (a growing number of journals require it) **but not the code**.

The paper is published and might include a link to a copy of data, but the **paper** itself will be **behind a paywall**.



"FINAL".doc



FINAL.doc!



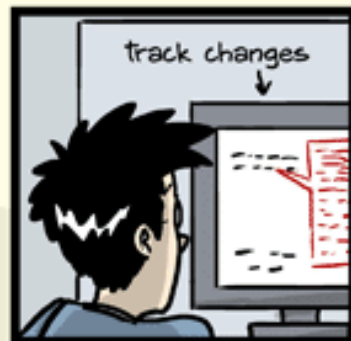
FINAL_rev.2.doc



FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5.
CORRECTIONS.doc



FINAL_rev.18.comments7.
corrections9.MORE.30.doc



FINAL_rev.22.comments49.
corrections.10. #@\$%WHYDID
ICOMETOGRADSCHOOL?????.doc





Open Science

For a growing number of scientists the process looks like this:

The data is stored in an open access repository like **figshare** or **Zenodo** and given its own **Digital Object Identifier (DOI)**.

The scientist creates a new **repository on GitHub** to hold the work.

As he does the analyses, changes are pushed to that repository.

When he is happy with the state of the paper, he posts a version to **preprint server** to invite feedback from peers.

The **published paper includes links to the preprint, to the code and data repositories**, which makes it easier for other scientists to replicate or use the work as starting point for additional research.



Open Science

Associating open data can increase the citation rate of scientific papers by as much as **69%** regardless the journal impact factor, date of publication, and the author's country of origin. But more than 80% of data never makes it to a repository.

Research articles that have been made Open Access are up to 600% more cited than those that have not.



Open Science

Simple Rules for Reproducible Computational Research:

For every result, keep track of how it was produced;

Avoid manual data manipulation steps (unique script does it all);

Archive the exact versions of all external programs used;

Version control all scripts (Git);

Record intermediate results, when possible in standardized formats;

For analyses that include randomness, note underlying random seeds.

e.g., in R use `set.seed(42)`;

Provide public access to scripts (Git) and data (Open repository).



Open access repositories

Online open access repositories allow researchers to **preserve and share their research outputs, including figures, datasets, images, and videos.**

E.g., Figshare and Zenodo are free to upload content and free to access, in adherence to the principle of open data.



Version control systems

Keep track of changes, creating different versions of our files -allows us to decide which changes will be made to the next version (**changes are called commits**).

Complete history of commits make up a repository, which can be kept in sync across different computers, facilitating collaboration.

Version control systems **start with a base version** of the document and then **record changes** you make each step of the way. You can rewind to any state of the document.

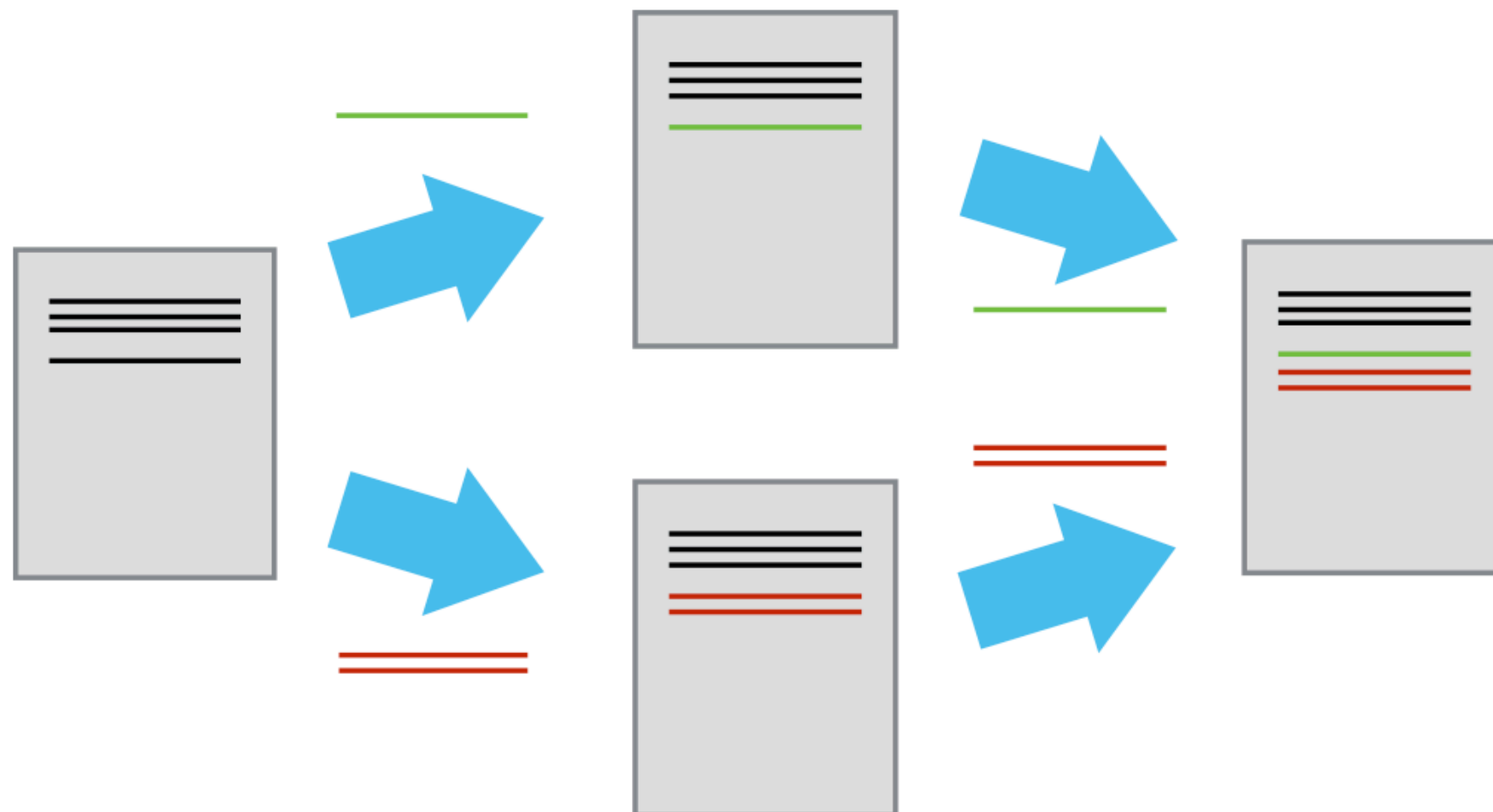




Version control systems

Users can make **independent changes** on the same document.

Unless multiple users make changes to the same section of the document - a conflict - you can **incorporate multiple sets of changes** into the same base document.










Using Git from RStudio

Version control are useful when developing data analysis scripts. RStudio has built-in integration with Git with a nice interface for many common operations.

RStudio allows to create a **project associated with a Git repository** to track the development and changes of the project over time, revert to previous versions and collaborate with others.




 Search or jump to...  [Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)   

Create a new repository


A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner

Repository name *

 jorgeassis ▾


 /

modelSeagrass 


Great repository names are short and memorable. Need inspiration? How about [supreme-invention?](#)

Description (optional)

A repository of R scripts to model the distribution of a seagrass species

☒  **Public**

Anyone can see this repository. You choose who can commit.


☐  **Private**

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☒ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.


Add .gitignore: **None** ▾

Add a license: **None** ▾ 




Create repository

Start by creating a new repository (e.g., GitHub).





[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)


  

[jorgeassis](#) / [modelSeagrass](#)


[Unwatch](#) 1 [Star](#) 0 [Fork](#) 0

[Code](#) [Issues](#) 0 [Pull requests](#) 0 [Actions](#) [Projects](#) 0 [Wiki](#) [Security](#) [Insights](#) [Settings](#)

[modelSeagrass](#) /



Drag files here to add them to your repository
Or [choose your files](#)



Commit changes

Add files via upload

Add an optional extended description...

☒ Commit directly to the `master` branch.
☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

[Commit changes](#) [Cancel](#)

The simplest way of sharing code is by uploading files to the repository.



GitHub interface showing the repository **jorgeassis / modelSeagrass**.

Repository description: A repository of R scripts to model the distribution of a seagrass species.

Repository statistics: 3 commits, 1 branch, 0 packages, 0 releases, 1 contributor.

Branch: master. Actions: New pull request, Create new file, Upload files, Find file, Clone or download.

Commit history:

- jorgeassis Rename myscript.R to myscript.R (Latest commit 690f2a0 now)
- README.md Initial commit (2 minutes ago)
- myscript.R Rename myscript.R to myscript.R (now)

File view: README.md

README content:

modelSeagrass

A repository of R scripts to model the distribution of a seagrass species

Permanently available at <https://github.com/jorgeassis/modelSeagrass>



README.md

Level of information to include in the readme file:

Repository name;

Project summary;

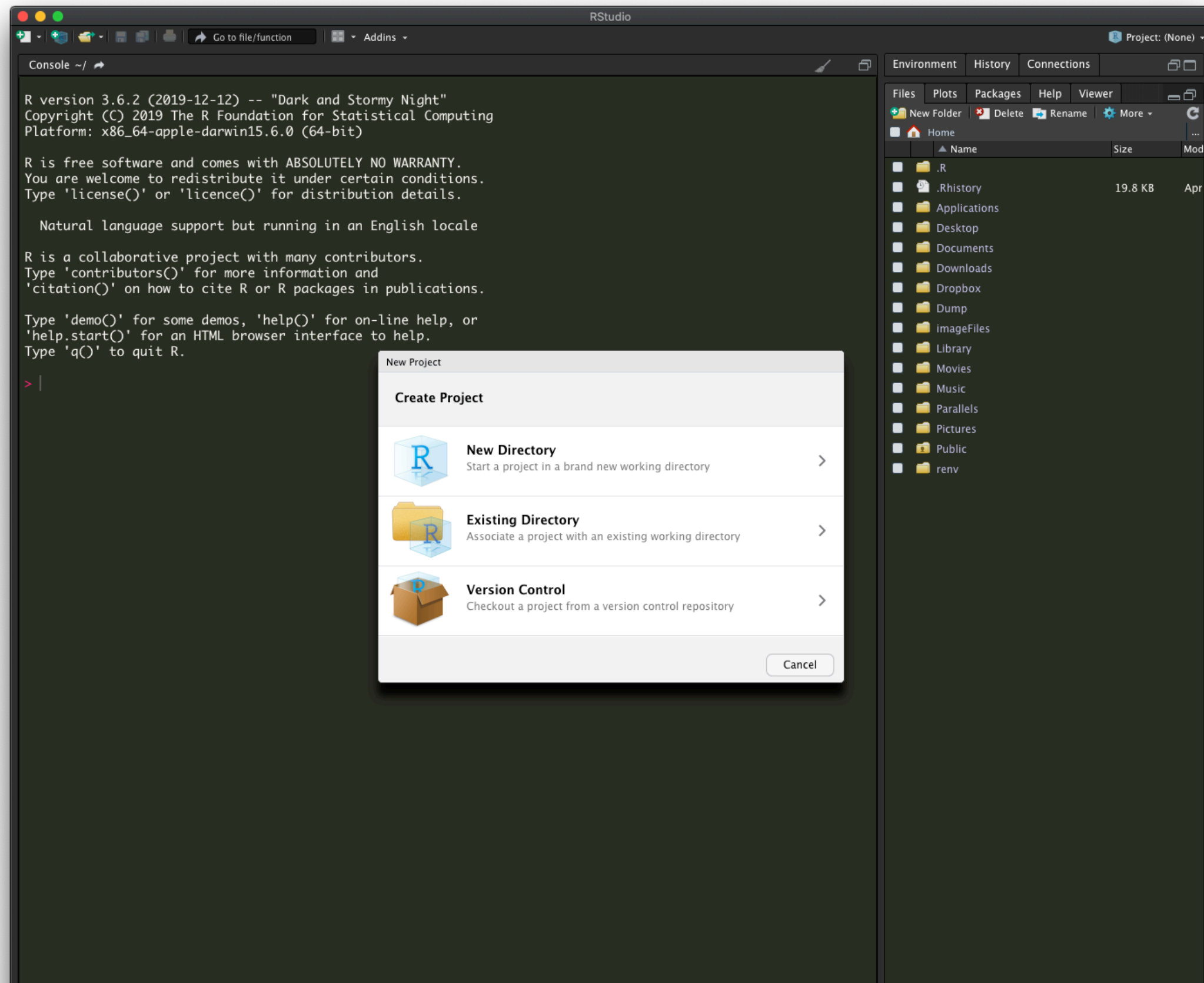
Funding information;

Primary contact(s) information;

Location of data and supporting information.

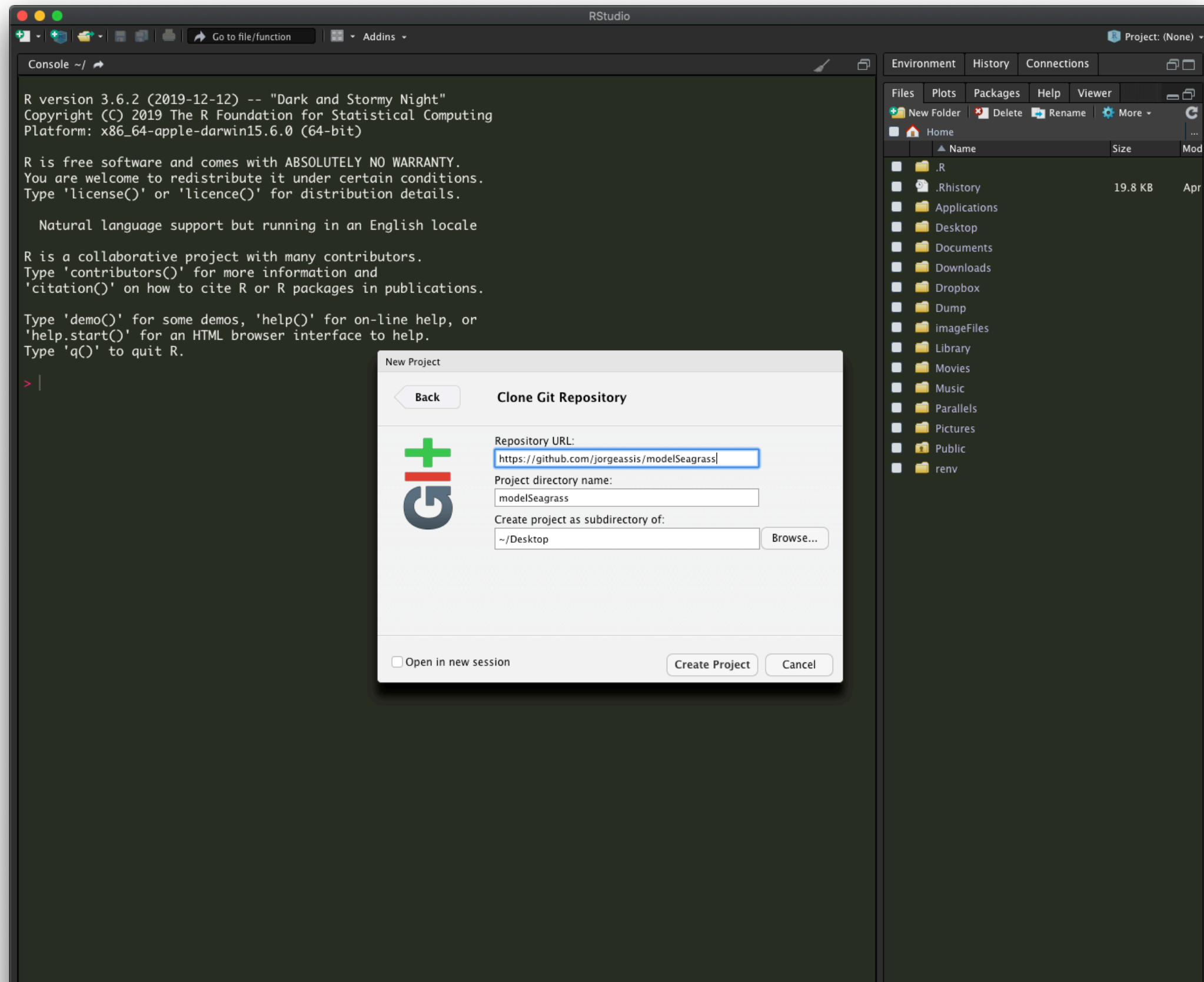


Proper integration of RStudio with a repository (e.g., GitHub).





Proper integration of RStudio with a repository (e.g., GitHub).





Saved edited files can be submitted by clicking on “Commit...”

The screenshot displays the RStudio IDE interface for a project named "modelSeagrass". The main editor window shows a script file named "myscript.R" with the following R code:

```
1 rm(list=ls()[ls()!="v"])]
2 gc(reset=TRUE)
3
4 source("Dependencies/mainFunctions.R")
5 source("0. config.R")
6
7 # -----
8 #
9
10 # I can edit the data in R and submit to github
```

The right-hand pane is divided into two sections. The top section, titled "Environment", shows the "Diff" and "Commit" buttons, and a table of staged files:

Staged	Status	Path
<input type="checkbox"/>		.gitignore
<input type="checkbox"/>		modelSeagrass.Rproj
<input type="checkbox"/>		myscript.R

The bottom section of the right-hand pane is the "Files" pane, showing the project structure:

Name	Size	Mod
..		
.gitignore	40 B	Apr
modelSeagrass.Rproj	205 B	Apr
myscript.R	175 B	Apr
README.md	90 B	Apr

The bottom pane is the "Console", showing the R version and environment information:

```
R version 3.6.2 (2019-12-12) -- "Dark and Stormy Night"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

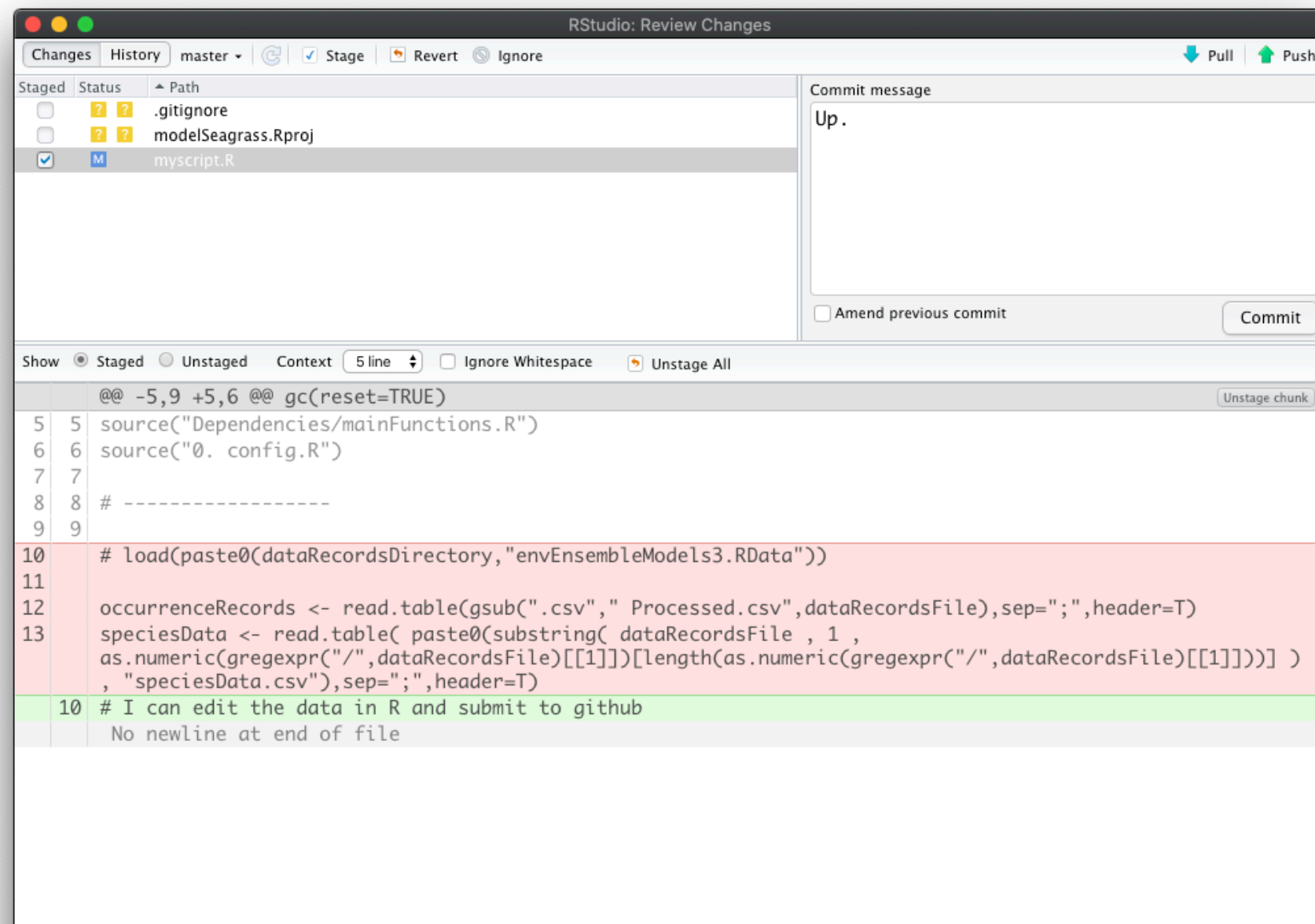
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```



A dialogue opens where we can select which files to commit, and enter a commit message. The icons in the “Status” column indicate the current status of each file (e.g., M for modified). Clicking on a file shows information about changes in the lower panel. Once everything is the way we want it, we click “Commit”. Changes can be pushed by selecting “Push Branch” from the Git menu.





The image shows a GitHub repository page for 'jorgeassis / modelSeagrass'. The repository description is 'A repository of R scripts to model the distribution of a seagrass species'. The file 'myscript.R' is selected, showing its content. The file has 10 lines (6 sloc) and 175 Bytes. The code is as follows:

```
1
2 rm(list=(ls()[ls()!="v"]))
3 gc(reset=TRUE)
4
5 source("Dependencies/mainFunctions.R")
6 source("0. config.R")
7
8 # -----
9
10 # I can edit the data in R and submit to github
```

The commit history shows a commit by 'Jorge Assis Up.' with hash '85d1c81' made 34 seconds ago. The file is currently on the 'master' branch.

Allows working in RStudio with a version control and backup system.