Tutorials for the Ocean Health Index

# Software for OHI

The Ocean Health Index works primarily with freely available, open-source software. To work the the Toolbox, there are several software programs you will need to download.

# Required Software

### R

Download [**R from cran.r-project.org**](http://cran.r-project.org/) and follow the install instructions for your computer.

If you already have R installed, check [**cran.r-project.org**](http://cran.r-project.org/) for updates. There are frequent updates available, and the current version is identified on the website. Compare what is available with what you already have on your computer by typing sessionInfo() into your R console (see also: [learning R](http://ohi-science.org/pages/learning_r.html)).

# Optional Software

### RStudio

We highly recommend working with [**RStudio**](http://www.rstudio.com/products/RStudio/), which is an interface that makes working with R much easier. RStudio also has great compatibility with **GitHub**, which we recommend for conducting regional assessments so you can collaborate with your team easily. RStudio does not get updated as often as R does, but check for updates regularly.

## Rmarkdown

[Rmarkdown](http://rmarkdown.rstudio.com/) is a powerful way to track progress as you use the Toolbox to reflect the characteristics and priorities in your study region. Here are some [Rmarkdown tips](https://github.com/OHI-Science/ohimanual/blob/master/tutorials/Rmarkdown_tips/Rmarkdown_tips.md#rmarkdown-tips).

# Accessing GitHub repositories

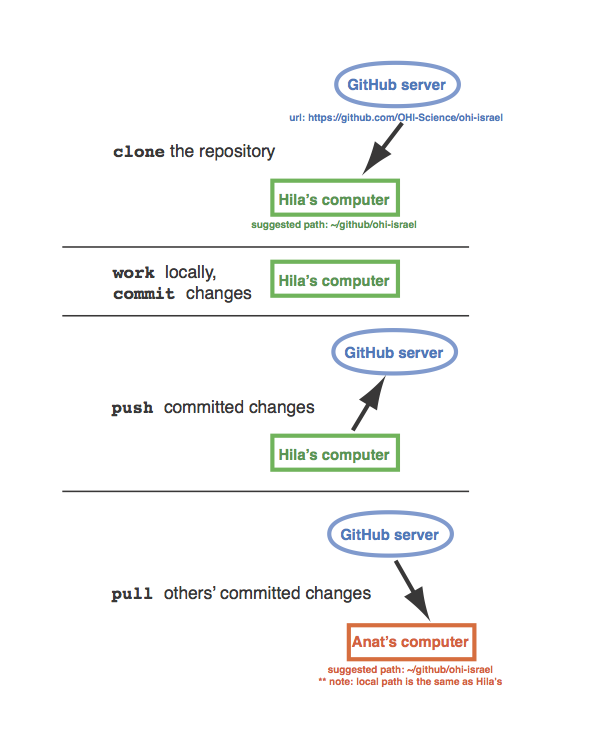
## OHI and GitHub

The Ocean Health Index (OHI) Toolbox is written in [R](http://cran.r-project.org/) and uses [GitHub](http://github.com), an [open-source development platform](http://en.wikipedia.org/wiki/GitHub) to develop and share software and data. GitHub has powerful versioning capabilities, which allow changes to be archived and tracked by each user. This is incredibly useful to not only document what work has been done, but how it differs from work done in the past, and who is responsible for the changes. GitHub enables easy collaboration with others, including the OHI team.

OHI has several repositories ('repos') where data and code are stored. From the [GitHub glossary](https://help.github.com/articles/github-glossary#repository):

A repository is the most basic element of GitHub. They're easiest to imagine as a project's folder. A repository contains all of the project files (including documentation), and stores each file's revision history. Repositories can have multiple collaborators and can be either public or private.

[OHI-Science](https://github.com/OHI-Science) is the 'owner' of OHI repositories, and individual users contribute to these repositories when they have permission. This simple example illustrates GitHub's collaborative workflow with the ohi-israel repo owned by OHI-Science:



### OHI regional assessments and GitHub

We recommend that groups interested in conducting OHI assessments do so through GitHub. This will enable collaboration and transparency, and will provide access to the latest developments in the Toolbox software. To get started, follow the steps below. The OHI team will create a repository for your regional assessment.

It is also possible to conduct a regional assessment without GitHub. We do not recommend this because it is more difficult for you to track progress and decisions, and for us to help or advise you. However, if this is preferable, see the instructions [here](https://github.com/OHI-Science/ohimanual/blob/master/tutorials/accessing_a_repo_without_GitHub/accessing_a_repo_without_GitHub.md). We can always get your filesystem onto GitHub later.

## Getting started

### What is needed?

![alt text(./fig/overview\_requirements.png)

### Install R and RStudio

Make sure you have the most current version of [R and RStudio](https://github.com/OHI-Science/ohimanual/blob/master/tutorials/software_for_OHI/software_for_OHI.md).

### Create a GitHub account

To get started, [signup](http://github.com) for a GitHub account, and provide your username to [bbest@nceas.ucsb.edu](mailto:bbest@nceas.ucsb.edu) or [lowndes@nceas.ucsb.edu](mailto:lowndes@nceas.ucsb.edu) so you can access your ohi-[assessment] repository.

### Install *git* Application

*git* is required to work behind the scenes on your computer. [Download](http://git-scm.com/downloads) and install *git*. (Here are a few [tips](https://github.com/OHI-Science/ohiprep/wiki/Setup#git)).

You will then need to set up your Git Identity, which identifies you with any changes made. You will use the command line:

* **Mac**: launch the Terminal application (Applications > Utilities > Terminal)
* **Windows**: go to command line in Windows (Start > Run > cmd)

Substitute your GitHub user information with the user John Doe:

git config --global user.name jdoe

and then:

git config --global user.email johndoe@example.com

You can check settings with the following:

git config --list

Quit the Terminal after typing:

exit

### Create your ohi-[assessment] repo

Contact the OHI team ([bbest@nceas.ucsb.edu](mailto:bbest@nceas.ucsb.edu) or [lowndes@nceas.ucsb.edu](mailto:lowndes@nceas.ucsb.edu)) to create a repository for your group. The repository will be stored on github.com and called OHI-Science/ohi-[assessment]: for example, github.com/OHI-Science/ohi-israel.

### Clone your repo to your computer

Once there is a repository for your OHI regional assessment called OHI-Science/ohi-[assessment], you can decide whether you will clone and work directly from that repository or from a forked repository. There are benefits to both approaches:

* Working directly from the repository is simplest: you can make changes on your local computer and push them directly to the online repository, as in the figure above. You do not need to send pull requests; simply clone from OHI-Science/ohi-[assessment] (see [Cloning options](https://github.com/OHI-Science/ohimanual/blob/master/tutorials/accessing_a_repo/accessing_a_repo.md#cloning-options)) and push/pull from there.
* [Forking](https://help.github.com/articles/fork-a-repo) a repository is best when you have multiple collaborators working on the same repository. This would allow you to commit changes to your local version, push commits up to github for offsite archiving, and eventually make a pull request to have those changes merged back to ohi-science/ohi-[assessment] while your collaborators do the same. This is a good way for someone who is not in the core team of the assessment to contribute comments.

\*\* With either approach, we recommend creating a folder called github in your root directory so that you can clone the repository to this file path: ~/github/ohi-[assessment]. This will make collaborating much simpler since everyone will work with the same file path.

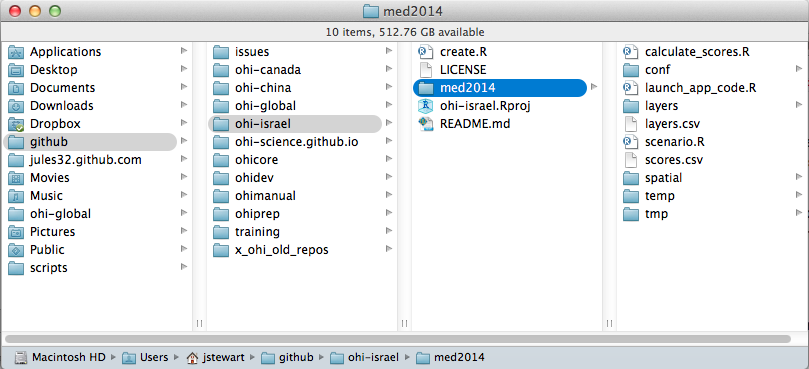
#### Cloning options

There are several options to clone the ohi-[assessment] repository to your local machine:

1. The GitHub App for [Mac](https://mac.github.com/) or [Windows](https://windows.github.com/). After installing the GitHub App, click Clone in Desktop button from the online repository (see the *Getting started guide* on a [Mac](https://mac.github.com/help.html) and [Windows](https://windows.github.com/help.html))
2. [The command line](https://help.github.com/articles/fork-a-repo#keep-your-fork-synced).
3. [RStudio](https://github.com/OHI-Science/ohiprep/wiki/Setup#rstudio). This is best after the initial clone, since RStudio occassionally has trouble with setting the username / password.

## Working locally

All of the files available from the online server repository https://github.com/OHI-Science/ohi-[assessment] are now cloned on your local computer, in the location you specified (we recommend cloning the repository to this file path: ~/github/ohi-[assessment]).



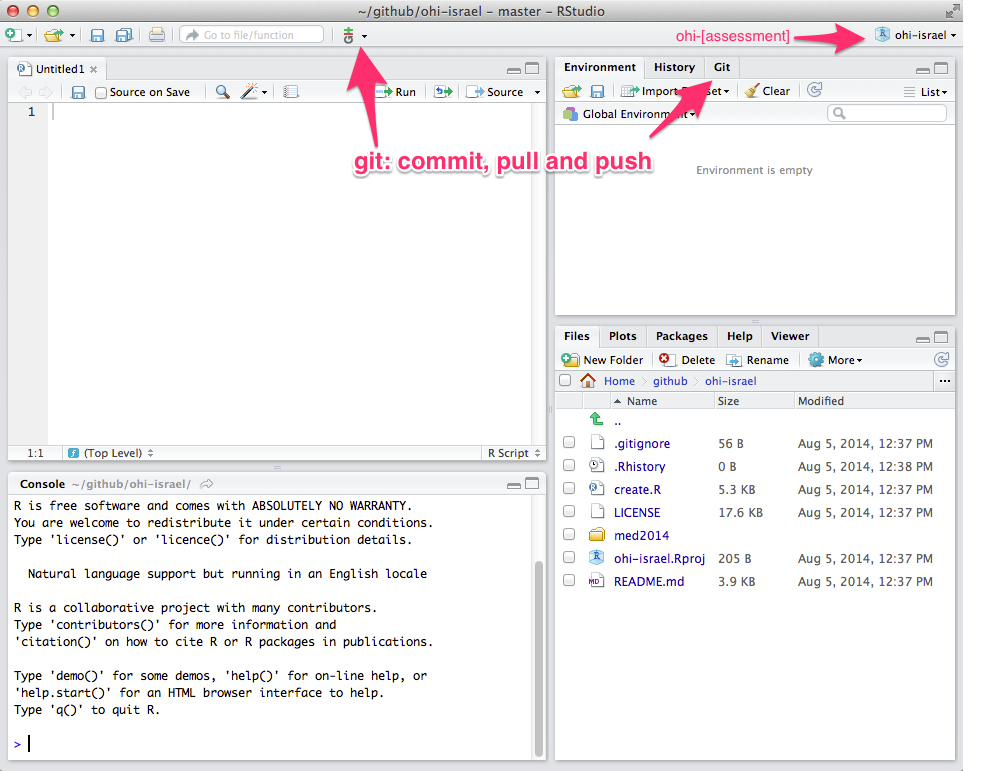
You will work on your computer to change the files in ~/github/ohi-[assessment] to reflect the desired modifications your team has identified for your regional assessment. All changes within this folder will be tracked by GitHub when you [commit](https://github.com/OHI-Science/ohimanual/blob/master/tutorials/accessing_a_repo/accessing_a_repo.md#committing-locally) these changes, *even if you make these changes outside of R or RStudio*. This means that you can delete or paste files in the Mac Finder or Windows Explorer, edit layers.csv and goals.csv in Excel or a text editor, and modify functions.r in RStudio, and commit all these changes to GitHub to track the changes you make.

## Committing and pushing changes locally

Committing and pushing changes can be done using [RStudio](https://github.com/OHI-Science/ohimanual/blob/master/tutorials/accessing_a_repo/accessing_a_repo.md#using-rstudio-to-commit-changes) or the [Github App](https://github.com/OHI-Science/ohimanual/blob/master/tutorials/accessing_a_repo/accessing_a_repo.md#using-github-app-to-push-commits-to-github-ohi-assessment-repository)

### Using RStudio to commit and push changes to GitHub ohi-[assessment] repository

Launch your ohi-[assessment] project in RStudio by double-clicking the ohi-[assessment].Rproj file. From RStudio, Commit locally, associating a message with each set of changes.

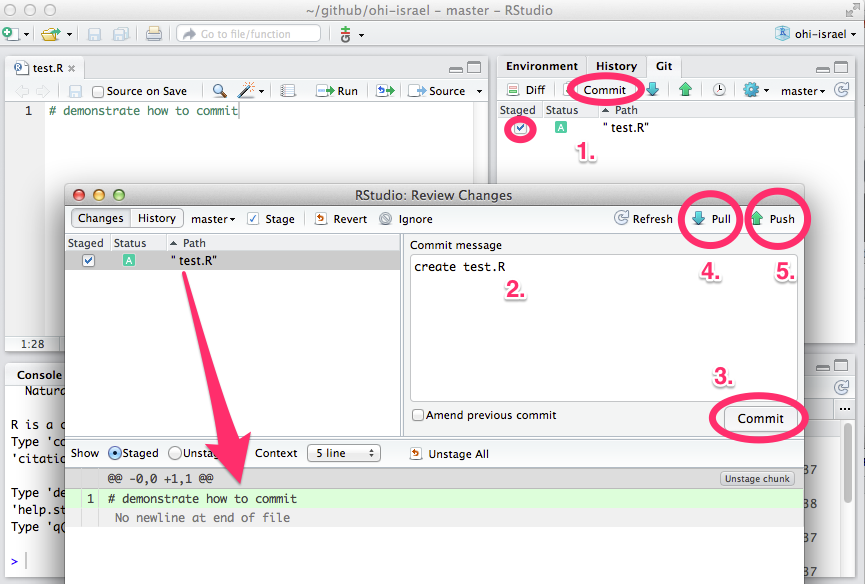


When you modify or add a file, the file will appear in the 'Git' window. In the example below, the file test.R was created.

1. Clicking the 'Staged' box and the 'Commit' button opens a new window where you can review changes.
2. Type a commit message that is informative to the changes you've made.

* Note 1: there will often be multiple files 'staged' at the same time, and so the same commit message will be associated with all of the updated files. It is best to commit changes often with informative commit messages.
* Note 2: clicking on a staged file will identify additions and deletions within that file for your review

1. Click 'Commit' to commit the changes and the commit message
2. Pull any changes that have been made to the online repo. This is important to ensure there are no conflicts with updating the online repo. See [Keeping synchronized](https://github.com/OHI-Science/ohimanual/blob/master/tutorials/accessing_a_repo/accessing_a_repo.md#keeping-syncronized).
3. Push your committed changes to the online server at github.com/OHI-Science/ohi-[assessment]. Your changes are now visable online.



### Using Github App to push commits to GitHub ohi-[assessment] repository

The Github App can also be used to commit and sync any changes made locally in the github folder. To do so, write a summary and description in the respective message bars in the Github App window, then click on commit and sync.

![alt text(./fig/github\_app.png)

## Keeping syncronized

Always pull before pushing commits.

### Pull commits to get changes from anyone else.

#### More Information

* [presentation: Reproducible science with the Ocean Health Index](http://bbest.github.io/talks/2014-06_OHI-repro-sci/#1)
* [wiki: Using GitHub](https://github.com/OHI-Science/ohiprep/wiki/Using-GitHub)
* see www.oceanhealthindex.org and ohi-science.org for more information and resources.

# Accessing repositories without GitHub

### OHI regional assessments without GitHub

It is possible to conduct a regional assessment without having a GitHub account. You can do a one-time download from the GitHub repository and then work locally on your computer without syncing to the GitHub server. We do not recommend this because it is more difficult for you to track progress and decisions, and for us to help or advise you. We recommend [accessing a repository with GitHub](https://github.com/OHI-Science/ohimanual/blob/master/tutorials/accessing_a_repo/accessing_a_repo.md) but provide instructions here of how to access a repository without Github. We can always get your filesystem onto GitHub later.

## Getting started

### Install the latest version of R

Download [**R**](http://cran.r-project.org/) and install this on your computer. If you already have R installed, check the website for updates. There are frequent updates to the R software, and the current version is identified on the websited. Compare what is available from their wesite with what you already have on your computer by typing sessionInfo() into your R console. (This will also identify packages you have installed).

### Install the latest version of RStudio

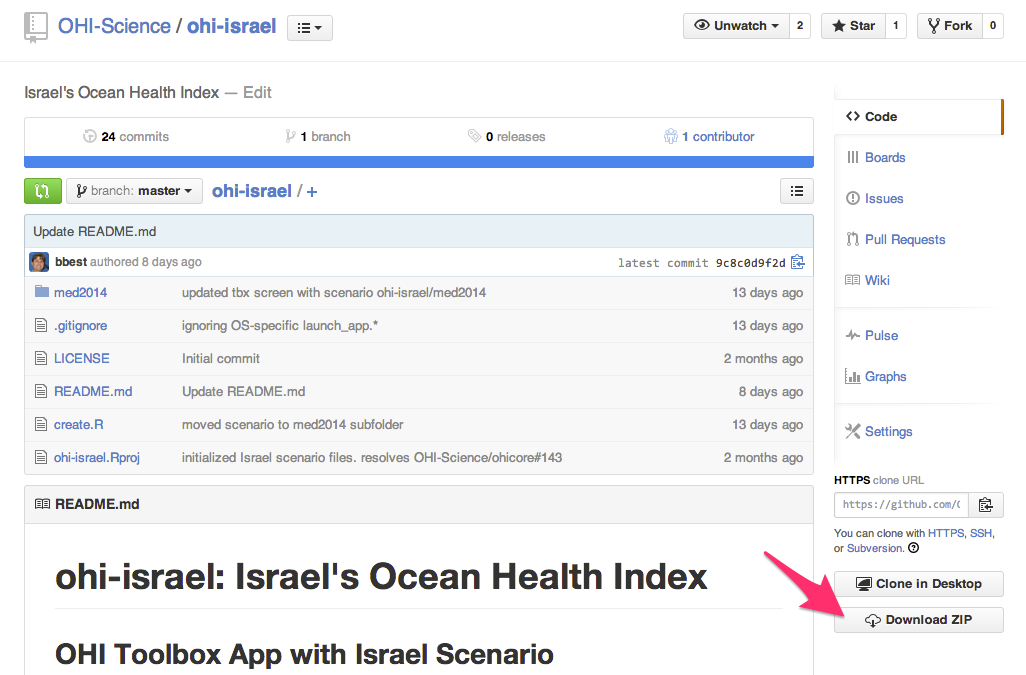
While not required, we highly recommend working with [**RStudio**](http://www.rstudio.com/products/RStudio/), which is an interface that makes working with R much easier. RStudio also has great compatibility with GitHub so you can collaborate with your team directly from your R code. RStudio does not get updated as often as R does, but it is good to check for updates regularly.

### Create your ohi-[assessment] repo

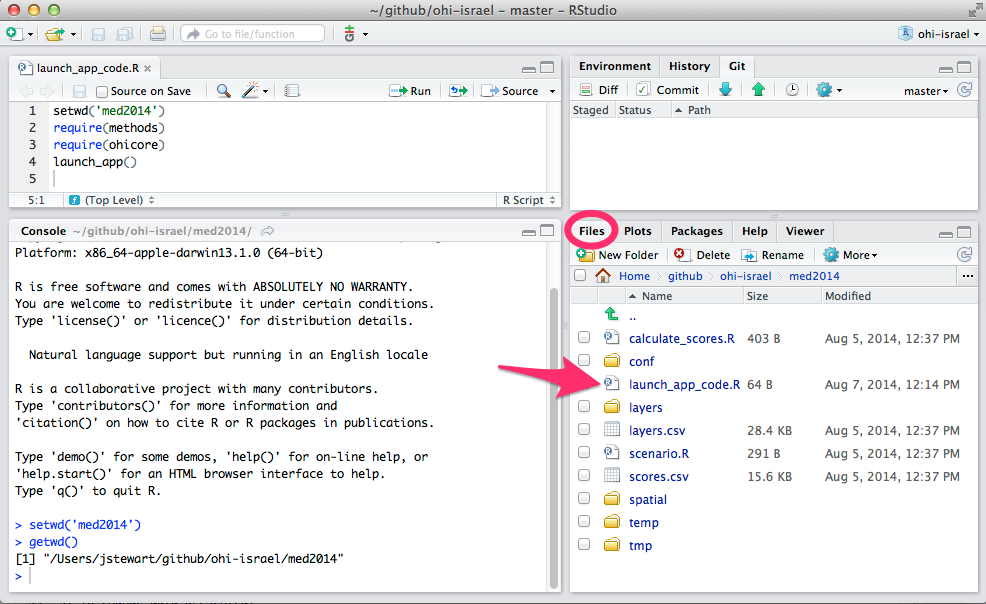
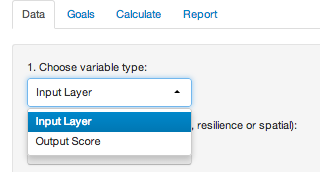
Contact the OHI team ([bbest@nceas.ucsb.edu](mailto:bbest@nceas.ucsb.edu) or [lowndes@nceas.ucsb.edu](mailto:lowndes@nceas.ucsb.edu)) to create a repository for your group. The repository will be stored on github.com and called OHI-Science/ohi-[assessment]: for example, github.com/OHI-Science/ohi-israel.

### Download the repo to your computer

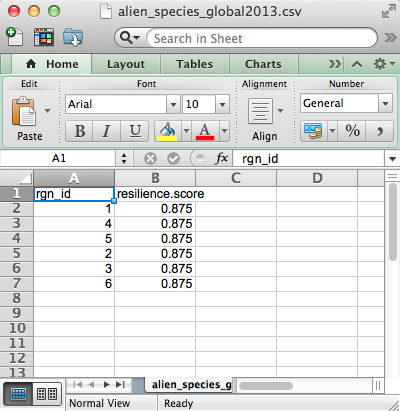
When your repository has been created, enter the url in to an internet browser (for example, github.com/OHI-Science/ohi-israel). Click Download ZIP.



### Working locally

1. Unzip the downloaded folder ohi-[assessment]-master and save the folder on your computer. We recommend creating a folder called github in your root directory and saving your repository to this file path: ~/github/ohi-[assessment]. This will make collaborating much simpler since everyone will work with the same file path.
2. Double-click ohi-[assessment]/ohi-[assessment].Rproj: this will launch RStudio. Be sure to have the [latest version of R and RStudio](https://github.com/OHI-Science/ohimanual/blob/master/tutorials/software_for_OHI/software_for_OHI.md).
3. Open ohi-[assessment]/[scenario]/launch\_app\_code.R (for example, ohi-israel/med2014/launch\_app\_code.R). You can open this from within RStudio: > 
4. Launch the Toolbox user interface Run launch\_app\_code.R to launch the Toolbox user interface. Notice that when the user interface opens, it may not be able to display *Output Scores*. This is because they have not been calculated yet: select *Input Layer* instead to see the map: > 
5. Navigate through the Toolbox file system Become familiar with the Toolbox [file system](https://github.com/OHI-Science/ohimanual/blob/master/tutorials/toolbox_manual/3_File_system.md#the-ocean-health-index-toolbox-manual). \*\*Note: this page is under active development\*\*
6. To change data presented: Within the folder ohi-[assessment]/[scenario]layers, open a .csv file and see that there are fewer regions represented than in the [global **layers** folder](https://github.com/OHI-Science/ohimanual/blob/master/tutorials/toolbox_manual/3_File_system.md#layers-folders).

This is because we have modified all the files to coincide with the six regions in the ohi-israel assessment. All these files have the suffix \_global2013.csv because they are just copies of the global assessment, with the same value copied into each regions. For example: *alien\_species\_global2013.csv* has six region identifiers (rgn\_ids, consistent with ohi-israel's six regions.

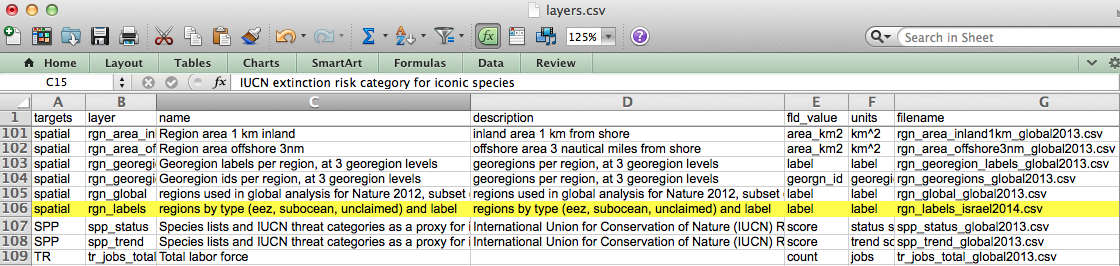


At this point, only one file is specific to the assessment: \*\*ohi-[assessment]/[scenario]/layers/rgn\_labels\_[scenario].csv\*\*, which is the file that describes all the regions (in this example, in Hebrew):



For the data layers that you will update, we suggest you replace the *\_global2013.csv* file, with a file that has a suffix identifying your scenario. So *alien\_species\_global2013.csv* would become *alien\_species\_med2014.csv* or *alien\_species\_israel2014.csv*. You can keep a copy of the old \*\_global2013.csv\* file in a separate foler or delete it (there is a record of it on the GitHub website ([example](https://github.com/OHI-Science/ohi-israel/blob/master/med2014/layers/alien_species_global2013.csv)).

1. Register the updated layers ohi-[assessment]/[scenario]/layers.csv is a registry (directory) of where the Toolbox looks for the data it uses. When you change a filename to \*\_china2014.csv\*, you will need to update the registry so the 'filename' column so the Toolbox know where to look.



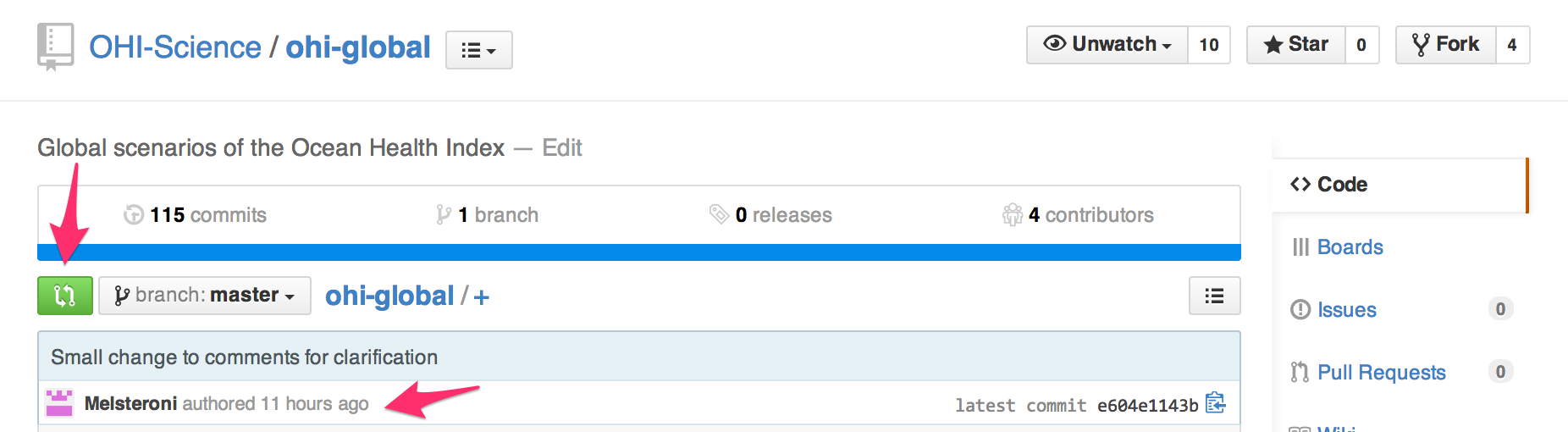
While you've been working on your forked repository locally on your computer (ex: remi-daigle/ohi-global), there may be committed updates in the owner's repository that is 'upstream' (ex: OHI-Science/ohi-global). To ensure that you are working with the most updated repository, you need to update your fork by creating a **pull request**. See some tips on process [here](https://github.com/OHI-Science/ohiprep/wiki/Using-GitHub#process).

## Updating your fork

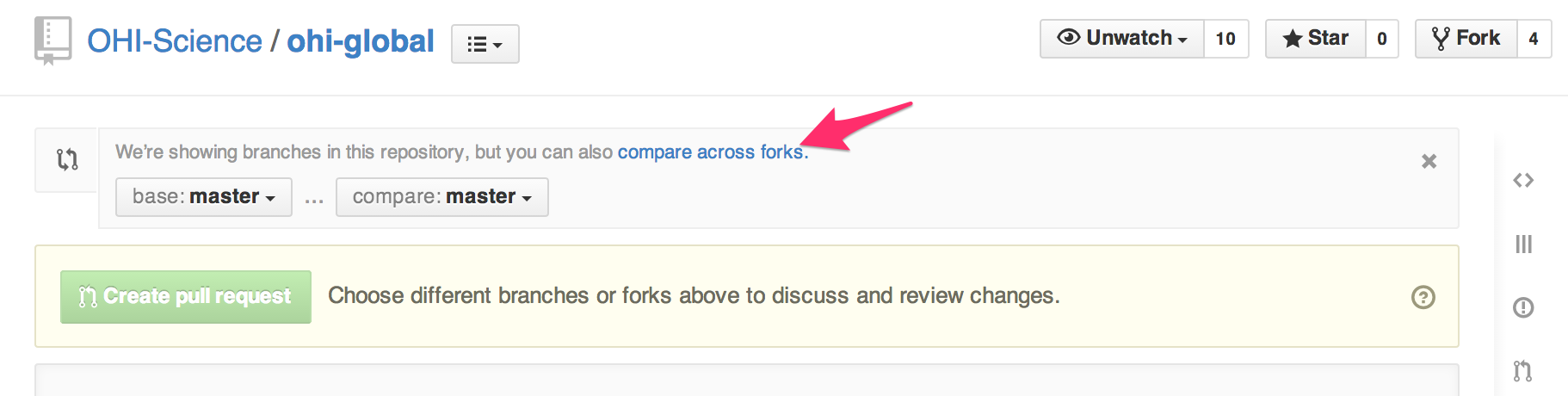
To do this on the command line, see [help.github.com](https://help.github.com/articles/syncing-a-fork).

To do this through Github:

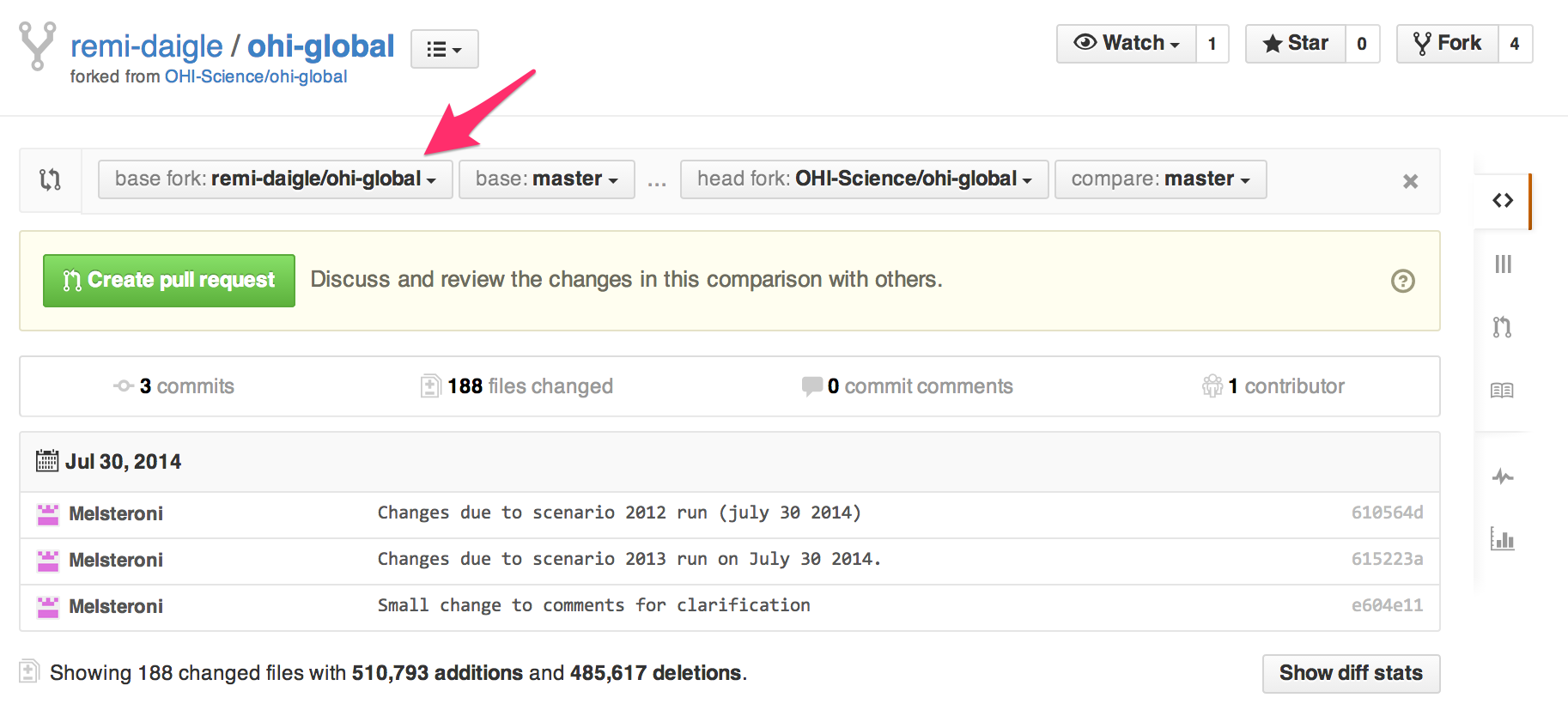
1. Go to the repository ('repo') belonging to the project owner 'upstream'. In this example, the owner is called OHI-Science and the repo is ohi-global. Note that there are recent updates by user Melsteroni. To pull the most recent version of ohi-global from OHI-Science, click on the green button.



1. You will need to sync forks (the default is to sync branches).

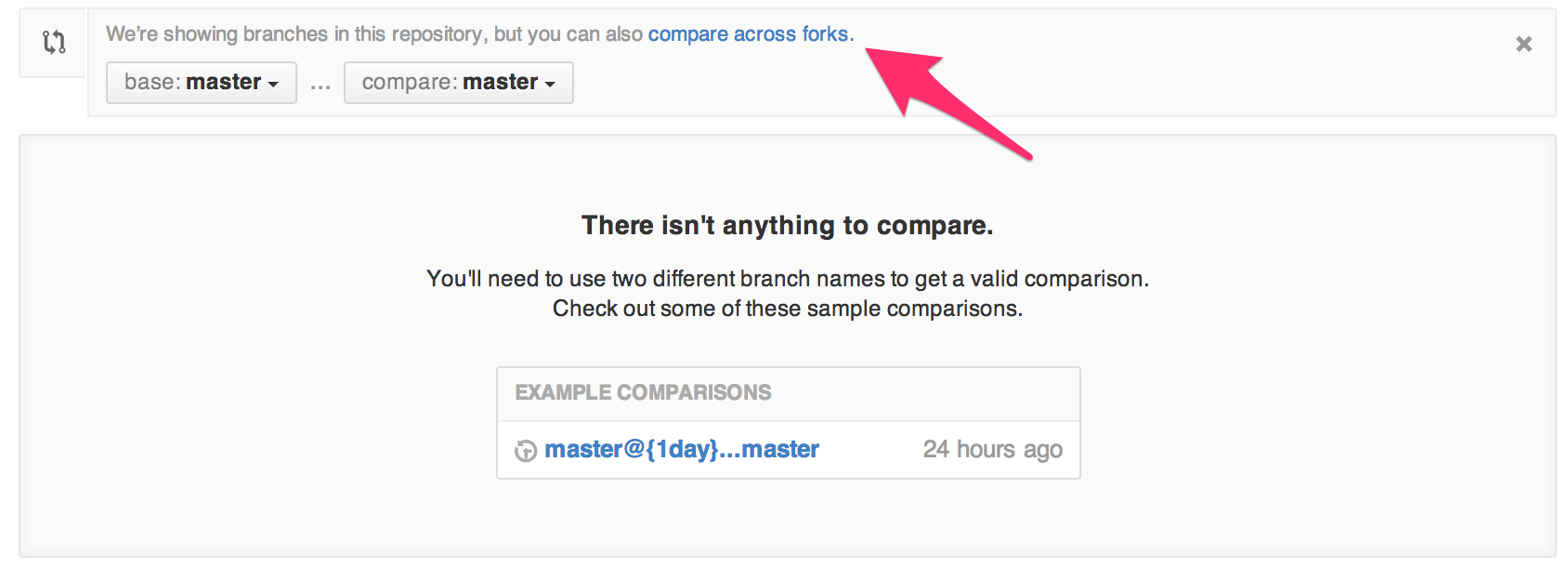


1. Change the base fork to your repo: in this example it is remi-daigle\ohi-global. The head fork is the owner's repo: OHI-Science\ohi-global. You can see those most recent commits by Melsteroni. Now you can click 'Create pull request'.

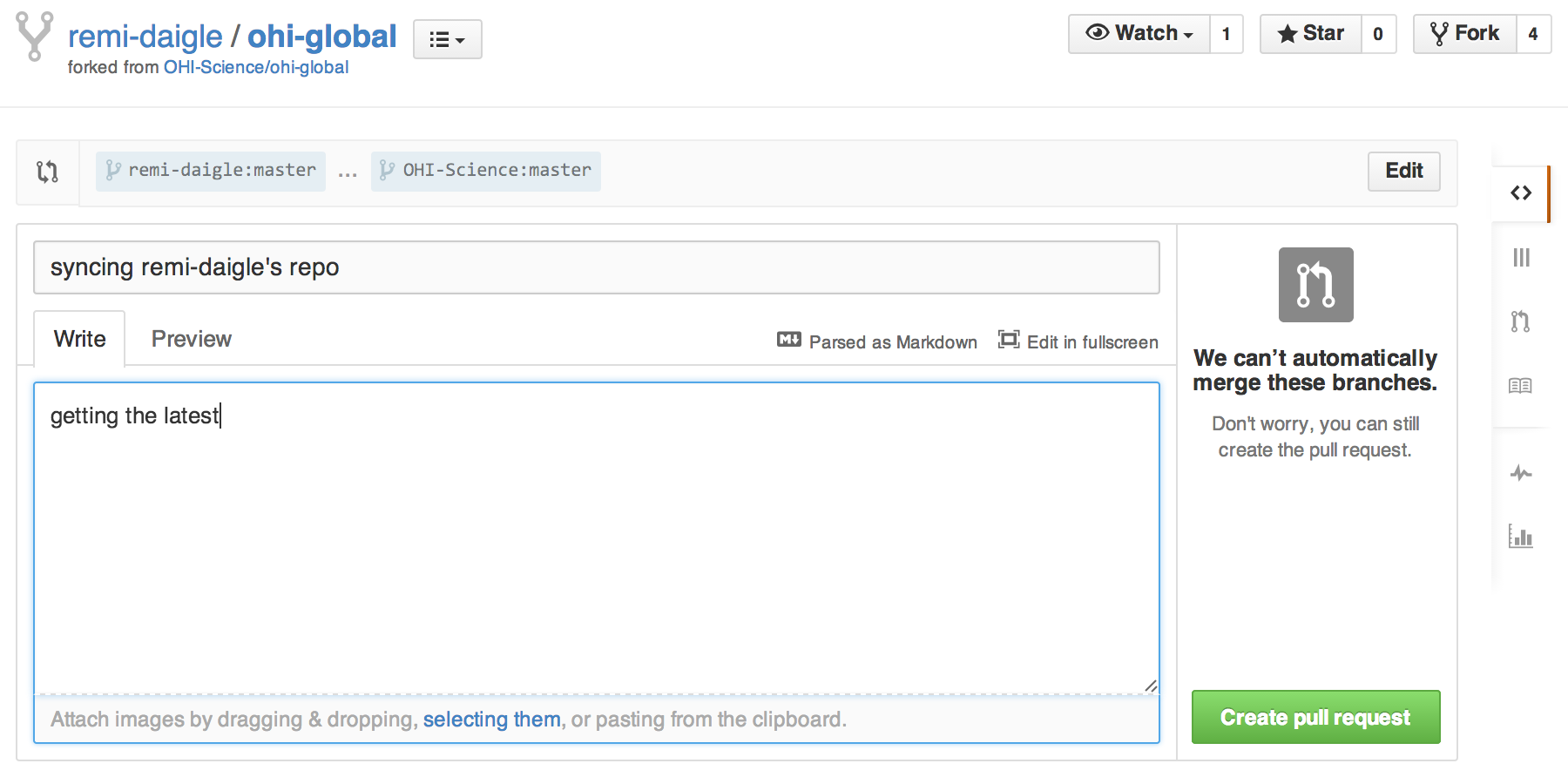


**Note:**

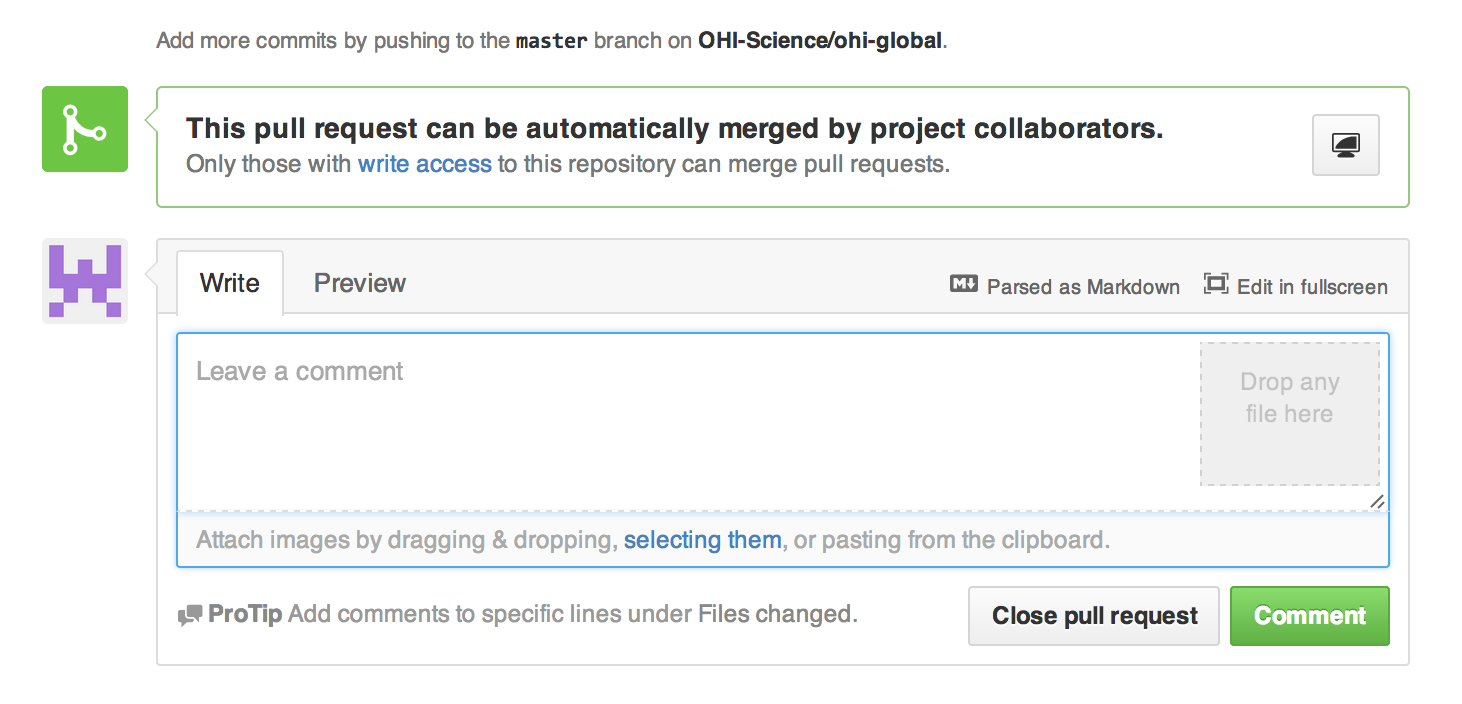
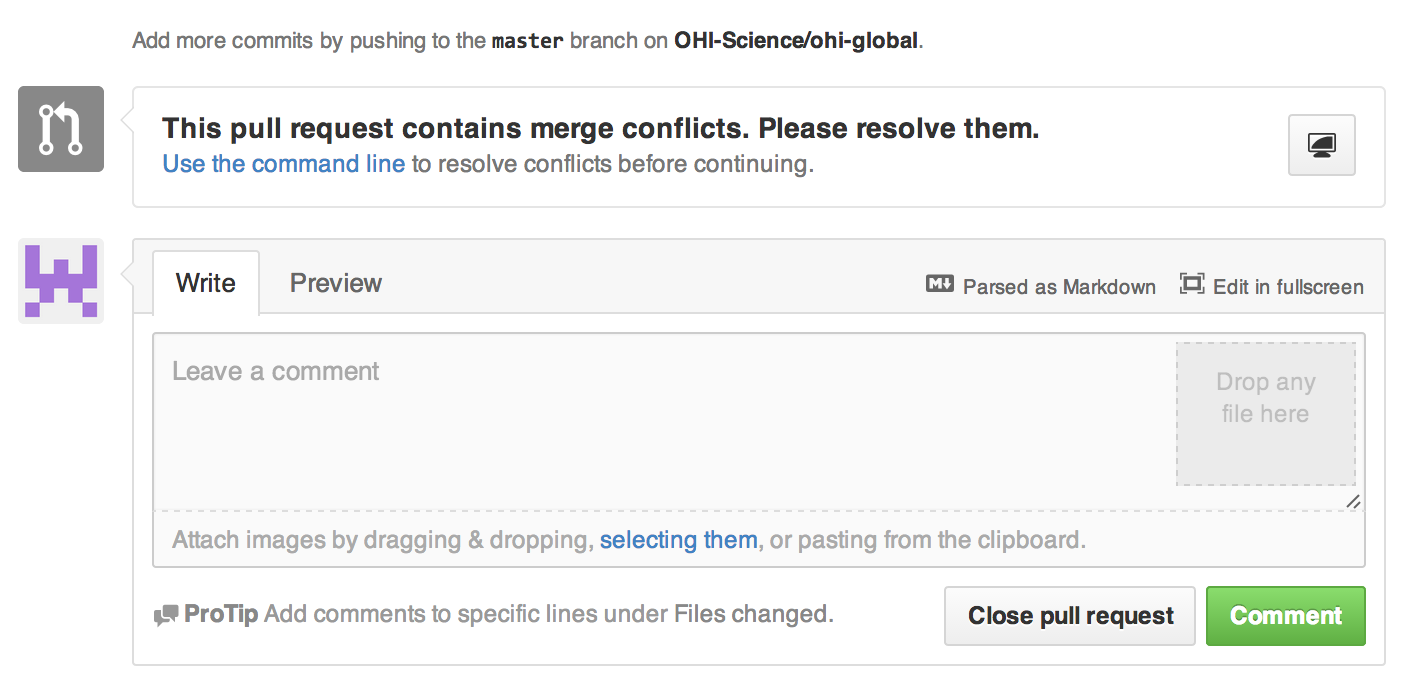
* when the base fork and head fork are identified, we are in the forked repo `remi-daigle/ohi-global'. It is also possible to create a pull request the the head fork from this location.
* if that if you get an error *There isn't anything to compare*, make sure you are still comparing forks, not branches.

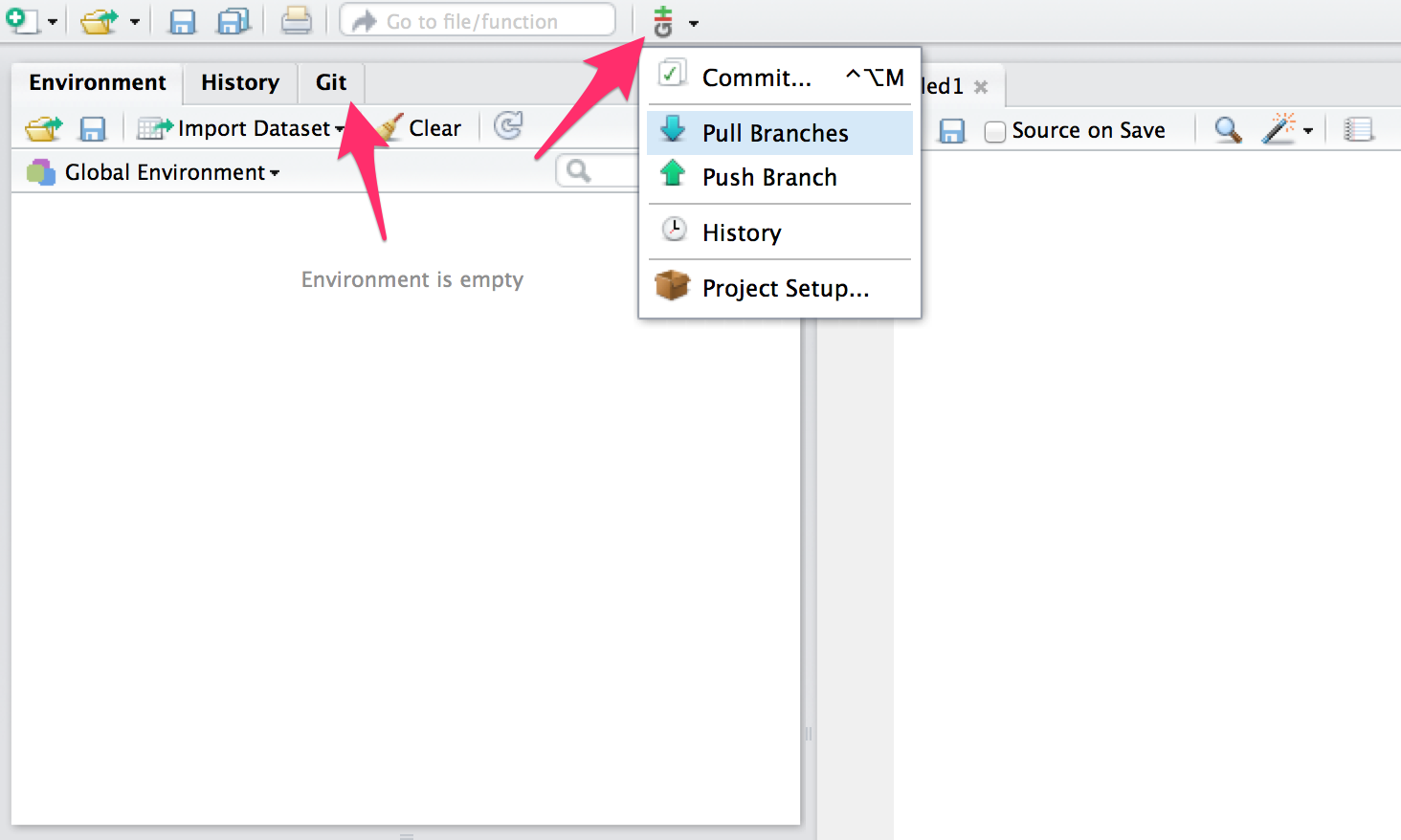


1. Enter a Title and a comment, when then allows you to click *Create pull request*



1. Scolling down on the next screen, you should either see:

* that you have been granted access 
* that you must request permission before merging
* you will be notified once you have been granted permission; then continue to step 6 below
* that you must resolve conflicts before merging 
* to resolve these conflicts, follow instructions from [OHI-Science](https://github.com/OHI-Science/ohiprep/wiki/Using-GitHub#resolving-conflicts).

1. Once your repo on on github.com is has been updated, you can pull your updated fork to your workspace in RStudio either from the Git pulldown menu or from the Git tab. 

# Rmarkdown Tips

# testing

## Rmarkdown

[Rmarkdown](http://rmarkdown.rstudio.com/) is a simple formatting tool for creating HTML, PDF, presentations, and Microsoft Word documents, and it can render R code within those formats. See this [great post](https://github.com/bbest/rmarkdown-example#writing-with-rmarkdown) as an introduction to its capabilities.

Here are a few examples of documents written in Rmarkdown, rendered in [.html](file:///Library/Frameworks/R.framework/Versions/3.1/Resources/library/dplyr/doc/introduction.html) and [.pdf](http://cran.r-project.org/web/packages/dplyr/dplyr.pdf) formats. Also, to see how Rmarkdown files can render code, look at the comparison of three OHI scenario goals.csv files in its [raw .rmd](https://github.com/OHI-Science/ohi-global/blob/master/table_goal_inputs.rmd) format and [rendered .md](https://github.com/OHI-Science/ohi-global/blob/master/table_goal_inputs.md) format.

## Getting started

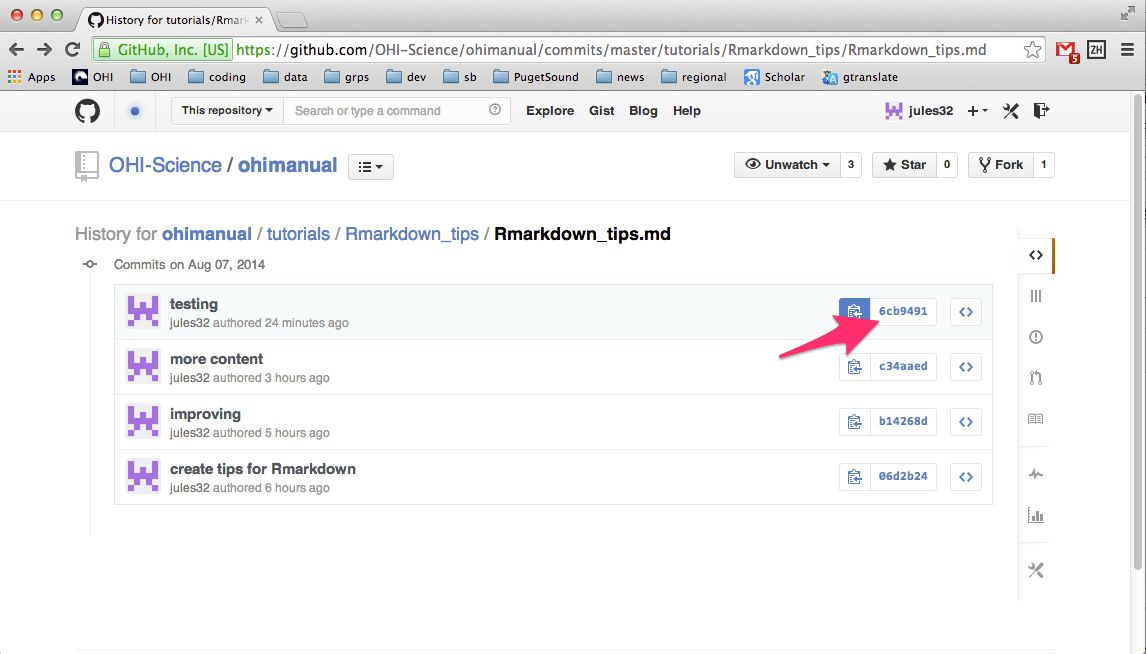
With the Ocean Health Index, we use Rmarkdown documents together with GitHub. To create or edit Rmarkdown content, you can do so with R and RStudio, or directly online on the GitHub interface.

### RMarkdown with GitHub

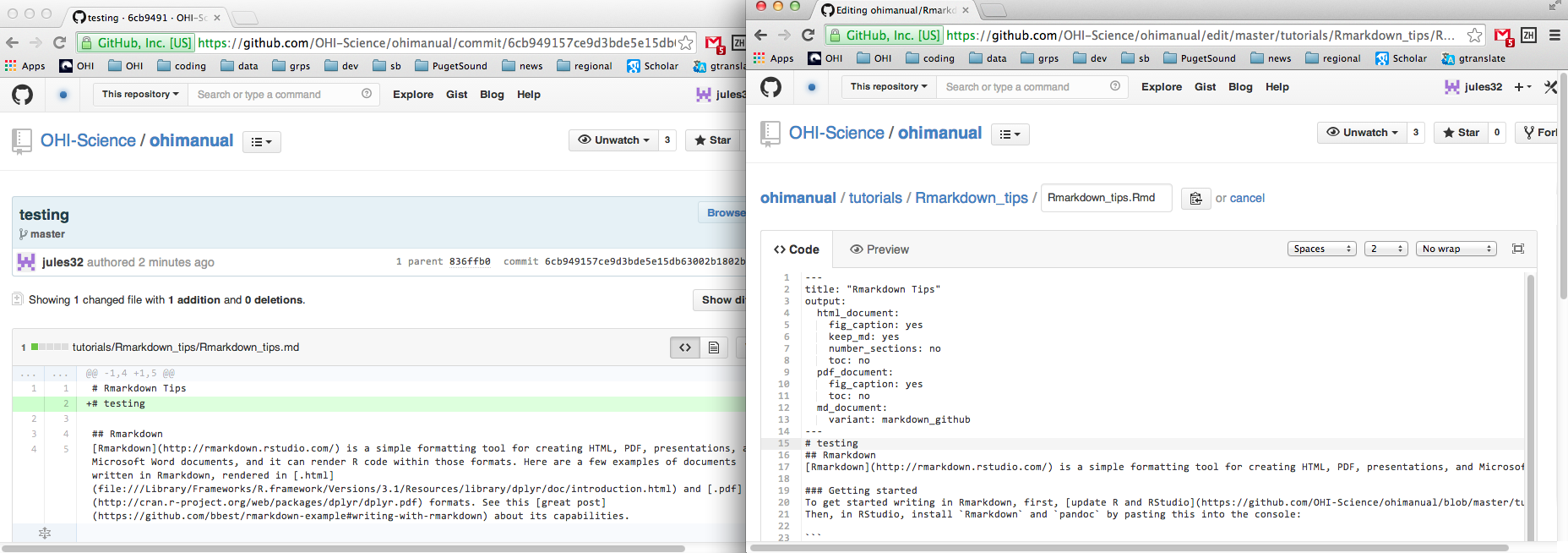
With an GitHub account and the correct permissions, you can [provide edits](https://help.github.com/articles/editing-files-in-another-user-s-repository) to files within github.com. This will actually create a copy of the file (that GitHub calls a 'fork') that you can edit and then suggest changes to the owner of the file.

Because GitHub cannot render .Rmd files, editing those files directly is difficult because you cannot immediately see your suggested edits. You are able to see rendered edits you make to .md files, but this will cause a problem because the next time the .Rmd file is rendered, it will overwrite the edits to .md. The temporary solution is to make edits to the .md file, so you can see the rendered changes, propose the file change, and then look at the changes side-by-side while you make those same changes to the .Rmd file.

Select the most recent change:



Make edits side-by-side:



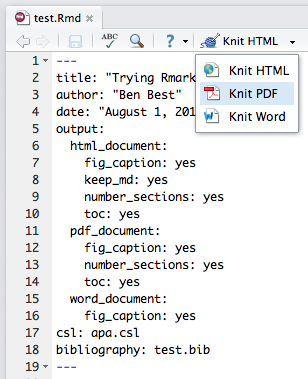
Your edits will be reviewed and merged into the documents by the file owner.

### RMarkdown with RStudio

To get started writing in Rmarkdown, first, [update R and RStudio](https://github.com/OHI-Science/ohimanual/blob/master/tutorials/accessing_a_repo/accessing_a_repo.md#getting-started).  
Then, in RStudio, install Rmarkdown (which also installs the required pandoc) with the pull-down menu or by pasting this into the console:

install.packages('rmarkdown')

Then, open a new [.Rmd document](https://github.com/bbest/rmarkdown-example#process). Instead of a blank document, it will give a template that you can **Knit** to see how it renders. Try it.



#### Editing

To add your own content, follow instructions about [syntax](http://rmarkdown.rstudio.com/authoring_basics.html). When you click the **Knit** button, a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. Rendering the .Rmd as both .html and .md documents is a good way to go. The .html file will show the rendered content, and is a good way to test that your syntax is correct. The .md file needs something something more to render its format--something that GitHub can provide. So when the md file is on GitHub, it will be very easy to read. If you're reading this tutorial on GitHub, you're reading the .md file.

Assign the output in the following way (spacing matters). Note that there are additional options you can specify (see the *Formats* tab of the [Rmarkdown page](http://rmarkdown.rstudio.com/) for more details.

---  
title: "Rmarkdown output formats"  
output:   
 html\_document:  
 fig\_caption: yes  
 keep\_md: yes  
 number\_sections: no  
 toc: yes  
 md\_document:  
 variant: markdown\_github  
---

### Asking for help

Rmarkdown is a powerful tool that can create publication-worthy documents, and has a lot of built-in options. However, like any new language or software, it takes a bit of time to learn how to use it. If there is something you are trying to do, Google your problem because it is likely that someone else has had the same question, and that someone else has had the answer. For example, here are a few solutions to problems you may have:

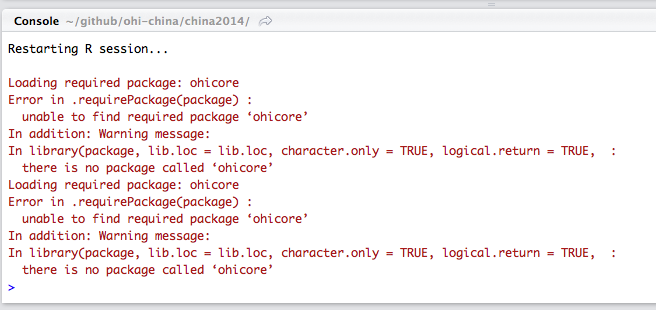
1. [continuing numbered lists](http://stackoverflow.com/questions/18088955/markdown-continue-numbered-list)
2. [saving a .md file with a different name and location than the .Rmd file](http://stackoverflow.com/questions/19989325/knit-rmd-file-to-md-and-save-the-md-file-one-level-up-with-a-different-name)

# Toolbox Troubleshooting

The Toolbox prints messages during its processing to help guide error checking and debugging. Here are a few troubleshooting tips.  
(\*\* note: this page is under development).

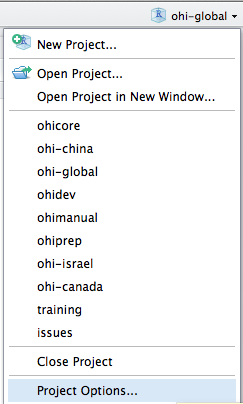
## Loading RWorkspace on Restart

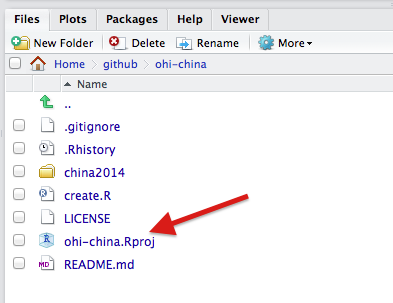
When you restart your R Session (Session > Restart R on a Mac), if you see that it is trying to load ohicore, it may give you an error:



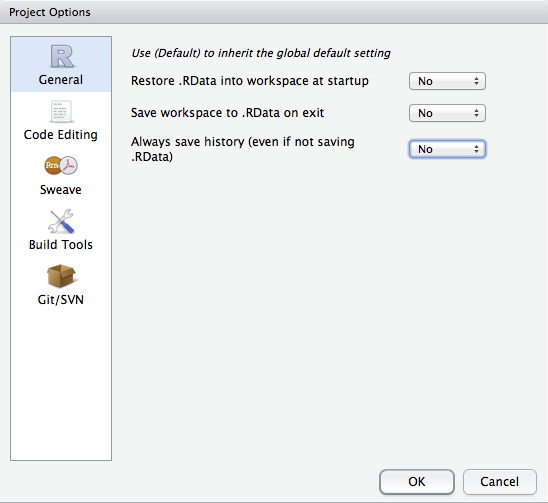
You do not want it to load ohicore or to save anything in your workspace. You will need to change the default setting from you **.Rproj** file. Steps to do this:

1. Go to Project Options, either in the pull-down menu or by double-clicking the .Rproj file:





1. Change all options to **No:**

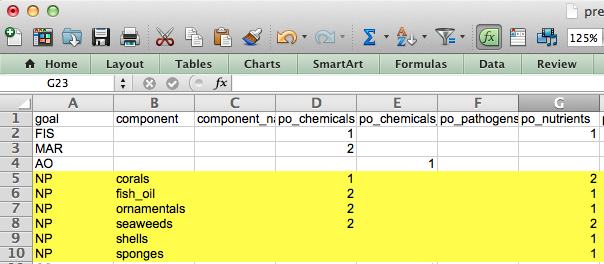


## Calculating Pressures...

### 'The following components for [goal] are not in the aggregation layer [layer]...'

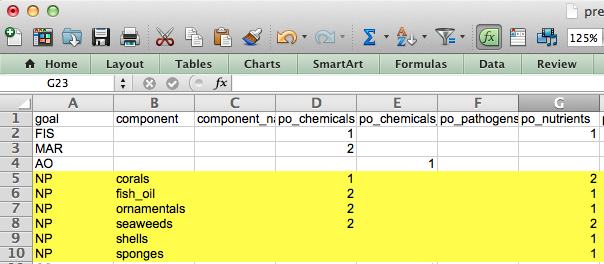
Example:

Running Setup()...  
Calculating Pressures...  
The following components for NP are not in the aggregation layer np\_harvest\_product\_weight categories (fish\_oil, ornamentals, seaweeds): corals, shells, sponges  
Error in data.frame(names(P), P) :   
 arguments imply differing number of rows: 0, 1



This error means you should update your pressures matrix because it expects there to be components that your region does not have.

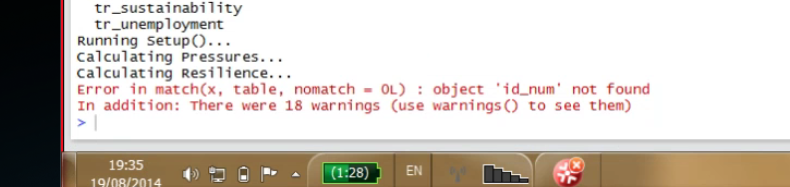
### 'Error in matrix...'

Example: > 

This error means there is an empty column in pressures\_matrix.csv, and the Toolbox cannot handle empty columns.

## Calculating Resilience ...

### 'Error in match(x, table, nomatch = OL) : object id\_num not found'



This error means you should check that there is at least one entry for each goal (for each row) in resilience\_matrix.csv.