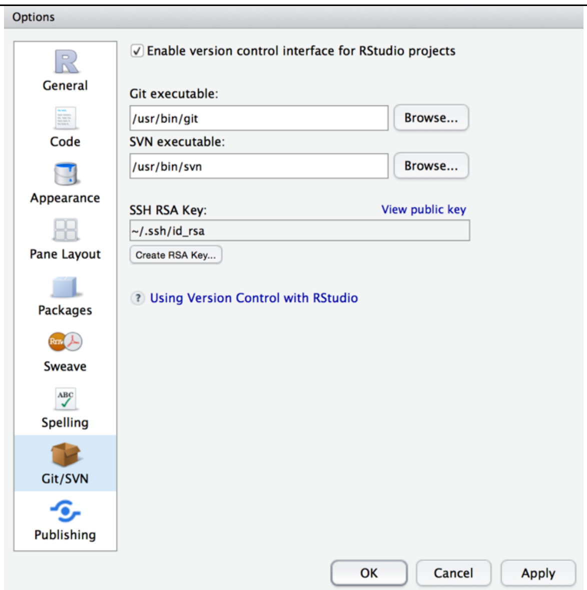
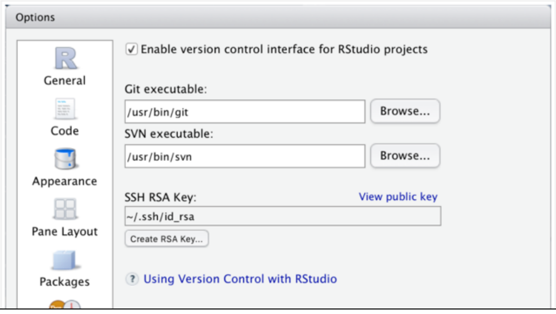
Github – R studio Cheat Sheet

Why use version control?

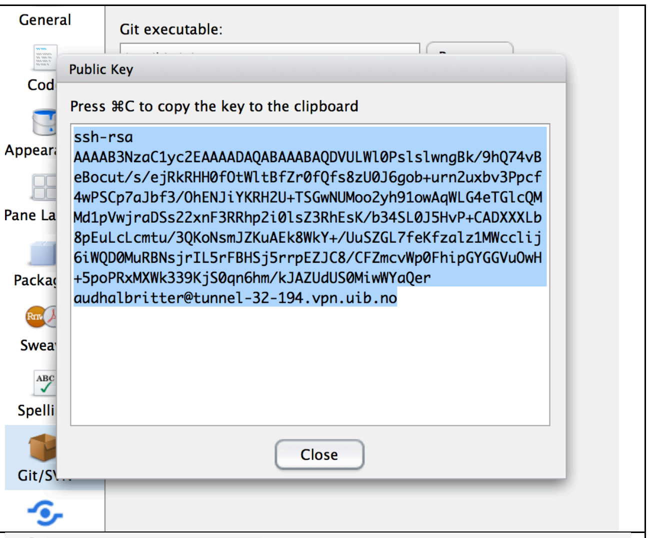
- Easy to share code
- Work on the same code at the same time
- Keeping track of changes in the code

Preparation:

1. Download Git or similar: https://git-scm.com/
   For UiB users git is on the Software Centre.
2. Get a github account: https://github.com/
3. Connect RStudio with Github

CONNECT RSTUDIO WITH GITHUB	
In RStudio click *Tools > Version Control,* select Git	
In RStudio, *Tools > Global Options*, select Git/SVN tab. Ensure the path to the *Git executable* is correct. This is particularly important in Windows where it may not default correctly (e.g. C:/Program Files (x86)/Git/bin/git.exe).	 ☑ Enable version control interface for RStudio projects Git executable: /usr/bin/git SVN executable: /usr/bin/svn SSH RSA Key: ~/.ssh/id_rsa Options General Code Appearance Pane Layout Packages Sweave Spelling Git/SVN Publishing Browse... Browse... View public key Create RSA Key... Using Version Control with RStudio OK Cancel Apply
Click *Create RSA Key* > Click *Create* And close this window.	 ☑ Enable version control interface for RStudio projects Git executable: /usr/bin/git SVN executable: /usr/bin/svn SSH RSA Key: ~/.ssh/id_rsa Options General Code Appearance Pane Layout Packages Browse... Browse... View public key Create RSA Key... Using Version Control with RStudio

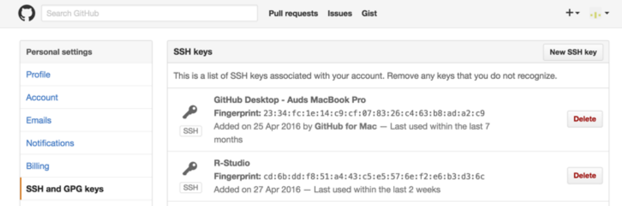
Click, *View public key,* and copy the displayed public key.



Press ⌘C to copy the key to the clipboard

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDVULWl0PslslwngBk/9hQ74vB
eBocut/s/ejRkRHH0fOtWltBfZr0fQfs8zU0J6gob+urn2uxbv3Ppcf
4wPSCp7aJbf3/OhENJiYKRH2U+TSGwNUMoo2yh91owAqWLG4eTGlcQM
Md1pVwjraDSs22xnF3RRhp2i0lsZ3RhEsK/b34SL0J5HvP+CADXXXLb
8pEuLcLcmtu/3QKoNsmJZKuAEk8WkY+/UuSZGL7feKfzalz1MWcclij
6iWQD0MuRBNsjrIL5rFBHSj5rrpEZJC8/CFZmcvWp0FhipGYGGVuOwH
+5poPRxMXWk339KjS0qn6hm/kJAZUdUS0MiwWYaQer
audhalbritter@tunnel-32-194.vpn.uib.no
```

Go to Git, click in the upper right corner and open your account *Settings*. Click the *SSH keys* tab. Click *New SSH key*. Paste in the public key you have copied from RStudio.
Title: RStudio



This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

GitHub Desktop - Auds MacBook Pro
Fingerprint: 23:34:fc:1e:14:c9:cf:07:83:26:c4:63:b8:ad:a2:c9
Added on 25 Apr 2016 by GitHub for Mac — Last used within the last 7 months

R-Studio
Fingerprint: cd:6b:dd:f8:51:a4:43:c5:e5:57:6e:f2:e6:b3:d3:6c
Added on 27 Apr 2016 — Last used within the last 2 weeks

Now connect RStudio and git. In RStudio, click *Tools > Shell*.
Type each line separately and click *Return:*

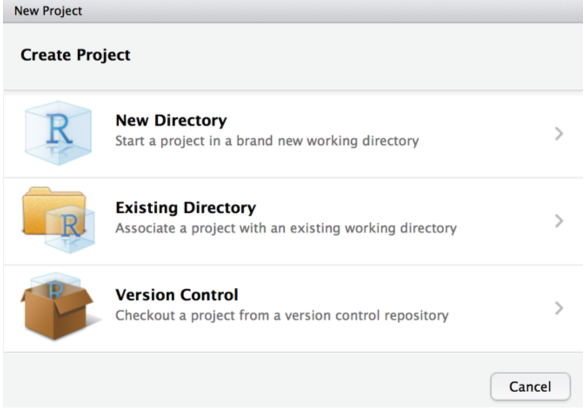
```
git config --global user.email "EnterYourEmailAddress"
git config --global user.name "EnterYourUsername"
```

OPTIONS:

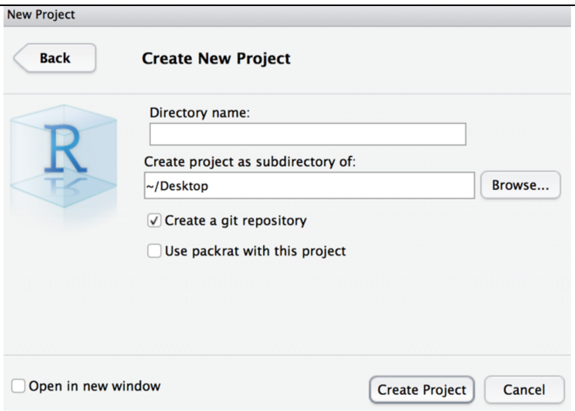
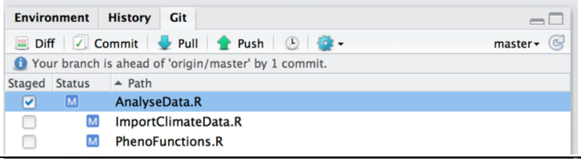
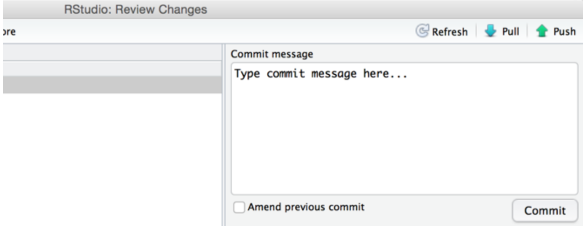
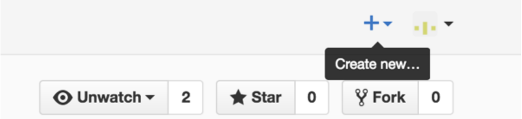
1. **Create a new RStudio Project and push it to Github**: You are starting a new RStudio project from scratch and want to connect it to Github.
2. **Clone an existing Github Repository to an RStudio Project**: You already have a Github repository and want to create an RStudio project or you want to collaborating with somebody's Github repository.
3. **Push an existing RStudio Project to a new Github Repository**: You have an existing RStudio project, which you have worked on for a while and now you want to create an Github repository to keep track on the changes you make.

1. CREATE A NEW R POJECT AND PUSH IT TO GITHUB

In RStudio click *File > New Project > New Directory*



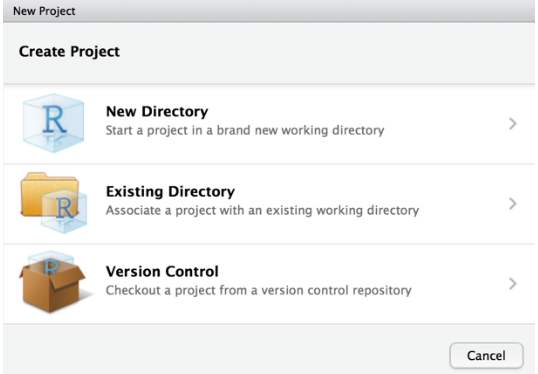
New Project
Create Project
New Directory
Start a project in a brand new working directory
Existing Directory
Associate a project with an existing working directory
Version Control
Checkout a project from a version control repository
Cancel

Name the project. Click *Create a git rpository* Create new script which you will add to your repository.	 New Project Back · Create New Project Directory name: Create project as subdirectory of: ~/Desktop · Browse... ☑ Create a git repository ☐ Use packrat with this project ☐ Open in new window · Create Project · Cancel
After saving your new script, it should appear in the *Git* tab on the *Environment / History panel*. Click the file you wish to add.	 Environment · History · Git Diff · Commit · Pull · Push · master Your branch is ahead of 'origin/master' by 1 commit. Staged · Status · Path ☑ M AnalyseData.R ☐ M ImportClimateData.R ☐ M PhenoFunctions.R
Now click *Commit* and enter an identifying message in *Commit message.* You have now committed the current version of this file to your repository on your computer/server. Now you want to *push* the contents of this commit to GitHub, so it is also backed-up off site and available to collaborators.	 RStudio: Review Changes Refresh · Pull · Push Commit message Type commit message here... ☐ Amend previous commit · Commit
Go to Github. Create a git *New repository*, and name it.	 Create new... Unwatch 2 · Star 0 · Fork 0

In RStudio, again click *Tools > Shell.*
Type each line separately and click *Return:*

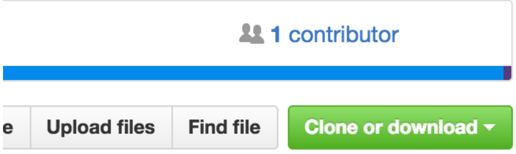
```
git remote add origin https://github.com/ YourGitUserName/YourGitProject.git
git config remote.origin.url git@github.com: YourGitUserName/YourGitProject.git
git pull -u origin master
git push -u origin master
```

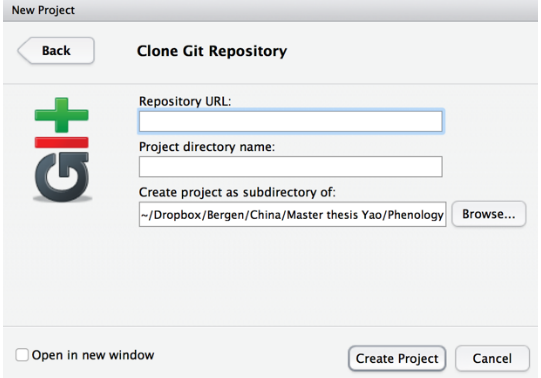
You have now pushed your commit to GitHub, and should be able to see your files in your GitHub account. The *Pull Push* buttons in RStudio will now also work. Remember, after each *Commit*, you have to *Push* to GitHub, this doesn't happen automatically

<table>
<tr><th colspan="2">2. CLONE EXISTING GITHUB REPOSITORY TO NEW RSTUDIO PROJCET</th></tr>
<tr><td>In RStudio click <em>File &gt; New Project > Version Control &gt; Git</em></td><td>


Start a project in a brand new working directory

Associate a project with an existing working directory

Checkout a project from a version control repository
</td></tr>
<tr><td>Go to the Git Resporitory and click on <em>Clone Git Respository</em> and entre it in <em>Respository URL</em>.</td><td>


</td></tr>
<tr><td>Entre <em>Project directory name</em>.
Click <em>Create Project</em></td><td>


Repository URL:

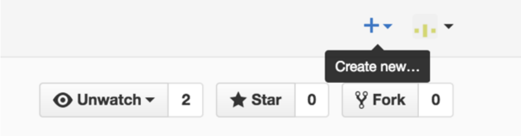
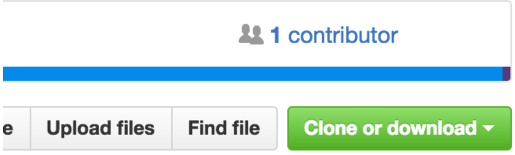
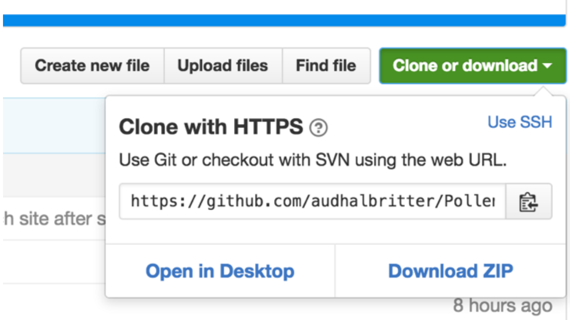
Project directory name:

Create project as subdirectory of: ~/Dropbox/Bergen/China/Master thesis Yao/Phenology

□ Open in new window
</td></tr>
<tr><td colspan="2">In RStudio, again click <em>Tools -&gt; Shell</em>
Enter and click <em>Return</em>

git config remote.origin.url git@github.com:EnterYourGitUserName/EnterYourGitProject.git</td></tr>
</table>

3. PUSH EXISTING RSTUDIO PROJECT TO GIT

Go to Github. Create a git *New repository*, and name it.	
Click on *Clone Git Respository*	 1 contributor e Upload files Find file Clone or download
Click the copy button	 Create new file Upload files Find file Clone or download h site after s Clone with HTTPS Use SSH Use Git or checkout with SVN using the web URL. https://github.com/audhalbritter/Pollei Open in Desktop Download ZIP 8 hours ago
Open your R project	

In RStudio, again click *Tools -> Shell*

Type each line separately and click *Return:*

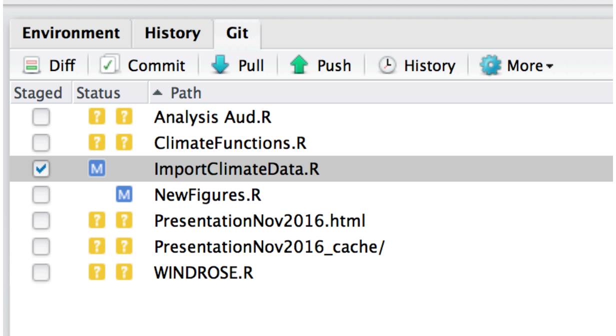
Copy the git url in line 5

```
echo "# test" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/EnterUserName/EnterGitProject.git
git push -u origin master
```

UserName and GitProject: if it is your repository enter your name, otherwise enter the username of the project owner.

GITHUB WORKFLOW

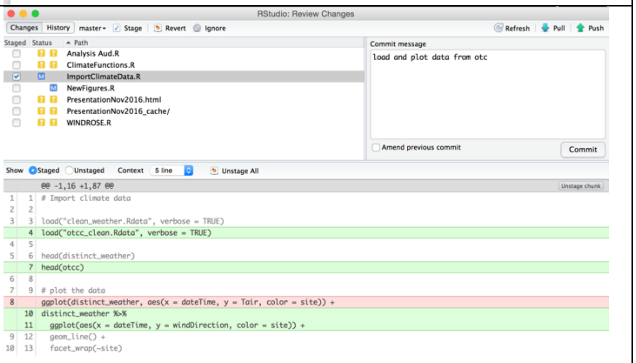
Create/edit/modify a file inside your repository. The modified file will turn up in the git panel, labeled with a M. Tick the box.



Staged	Status	Path
☐	? ?	Analysis Aud.R
☐	? ?	ClimateFunctions.R
☑	M	ImportClimateData.R
☐	M	NewFigures.R
☐	? ?	PresentationNov2016.html
☐	? ?	PresentationNov2016_cache/
☐	? ?	WINDROSE.R

Clickthe commit button and a new window will open. Stage the changes you made to the commit message (top right).
Click **commit** to commit these changes which creates a permanent snapshot of the file in the Git directory along with a message that indicates what you did to the file.

All the changes in the file will be shown in green and red color.
Green: code you have added
Red: code you have deleted



Staged	Status	Path
☐	? ?	Analysis Aud.R
☐	? ?	ClimateFunctions.R
☑	M	ImportClimateData.R
☐	M	NewFigures.R
☐	? ?	PresentationNov2016.html
☐	? ?	PresentationNov2016_cache/
☐	? ?	WINDROSE.R

Commit message: load and plot data from otc

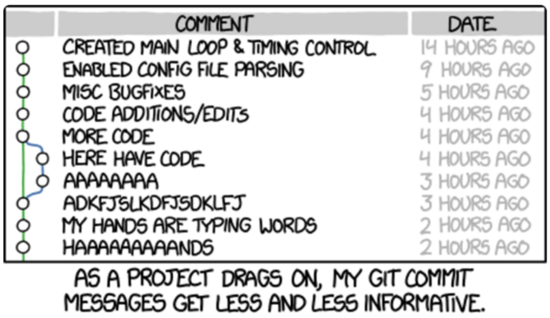
☐ Amend previous commit

Show ● Staged ○ Unstaged Context 5 line

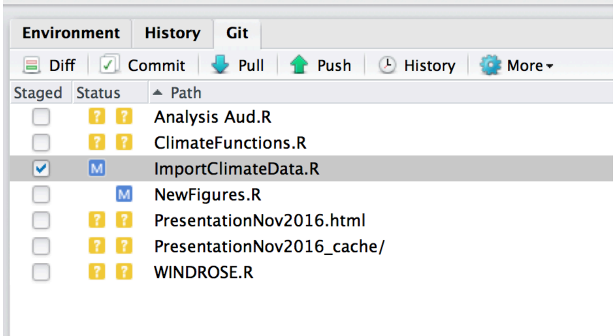
```
@@ -1,16 +1,87 @@
# Import climate data

load("clean_weather.Rdata", verbose = TRUE)
load("otcc_clean.Rdata", verbose = TRUE)

head(distinct_weather)
head(otcc)

# plot the data
ggplot(distinct_weather, aes(x = dateTime, y = Tair, color = site)) +
distinct_weather %>%
  ggplot(aes(x = dateTime, y = windDirection, color = site)) +
  geom_line() +
  facet_wrap(~site)
```

Commits are cheap. Commit often and provide useful messages so you can keep track of what you are doing. Don't do this:

(From xkcd)

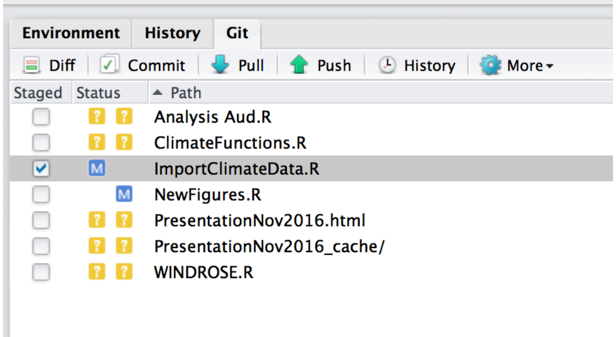


At this stage, everything is still on your hard drive. To upload your modifications (i.e., your commits) to GitHub you need to **push** (green arrow) to it.



Staged	Status	Path
☐	? ?	Analysis Aud.R
☐	? ?	ClimateFunctions.R
☑	M	ImportClimateData.R
☐	M	NewFigures.R
☐	? ?	PresentationNov2016.html
☐	? ?	PresentationNov2016_cache/
☐	? ?	WINDROSE.R

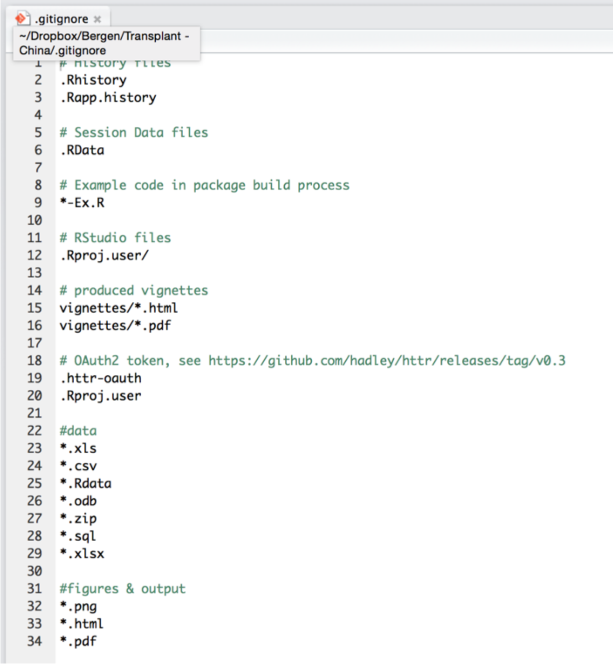
If you are working with other people that are also committing your shared repository on GitHub, you will need to **pull** (blue arrow) to bring their modifications into your local copy of the repository.



Staged	Status	Path
☐	? ?	Analysis Aud.R
☐	? ?	ClimateFunctions.R
☑	M	ImportClimateData.R
☐	M	NewFigures.R
☐	? ?	PresentationNov2016.html
☐	? ?	PresentationNov2016_cache/
☐	? ?	WINDROSE.R

Some rules:

- Commit code and plain text
- Don’t upload large files (e.g. data files)
- Don’t upload output files (figures)
- You can create a **gitignore** file where you can define, which rules can be uploaded to git.



```
# History files
.Rhistory
.Rapp.history

# Session Data files
.RData

# Example code in package build process
*-Ex.R

# RStudio files
.Rproj.user/

# produced vignettes
vignettes/*.html
vignettes/*.pdf

# OAuth2 token, see https://github.com/hadley/httr/releases/tag/v0.3
.httr-oauth
.Rproj.user

#data
*.xls
*.csv
*.Rdata
*.odb
*.zip
*.sql
*.xlsx

#figures & output
*.png
*.html
*.pdf
```

Useful commands in shell
List email and user name: git config --global --l
Change email and user name: git config --global user.email "EnterYourGitEmail.com" git config --global user.name "EnterYourGitUsername"
Information on repository: local git config --local --l
Make RStudio remember the username and password: git config remote.origin.url git@github.com:EnterYourGitUsername/EnterYourGitProject.git Example: git config remote.origin.url git@github.com:your audhalbritter/Phenology.git

References:
https://www.r-bloggers.com/rstudio-and-github/
http://r-bio.github.io/intro-git-rstudio/