





BarcelonaR - Workshop – 2nd September 2020



Workshop Setup:

Wi-Fi

Network Name: N/A

Password: N/A

Resources
 R (version 3.6.3)
 RStudio (version 1.3.1073)
 RStudio
 Git (version 2.28.0)





What is GitHub?

Git 🚯 is an open source version control system and is free to use!

GitHub is a code hosting platform that offers a user-friendly

web-based interface to use Git.

You can work on public or private **repositories** which means that you

can collaborate on projects with others from anywhere in the world!

Repository: A storage space where your project files and code live





Workshop aim:

Learn how to version control an R project using GitHub within Rstudio and the basics in using Git.

- Topics:
 - Install Git •
 - Setup a GitHub account •
 - y **GitHub** Create a GitHub repository •
 - Connect repository with R project •
 - Basic version control processes •



Why version control?



Because I want to...

- have a backup of the code
- see what has changed in the code over time
- be able to roll back my code to a previous version
- collaborate effectively with helpful documentation
- avoid multiple copies or versions



Workshop tasks



Introduction to GitHub with R



For this workshop we will take the step by step approach to exercises. I will cover one step and demonstrate it on my machine and then allow you a few minutes to repeat the step.

This will ensure that everyone is on the same page and troubleshoot early than when it is too late!





In order to start using the Git version control system we need to install it on our machine.

To download Git you need to go to: https://git-scm.com/downloads

and select your platform.

We then follow the next steps to install Git ...





Install Git



Introduction to GitHub with R

Downloading Git

Your download is starting...

You are downloading the latest (2.28.0) 64-bit version of Git for Windows. This is the most recent maintained build. It was released about 1 month ago, on 2020-07-28.

Click here to download manually, if your download hasn't started.

Other Git for Windows downloads

Git for Windows Setup 32-bit Git for Windows Setup.

64-bit Git for Windows Setup.

Git for Windows Portable ("thumbdrive edition") 32-bit Git for Windows Portable.

64-bit Git for Windows Portable.

The current source code release is version **2.28.0**. If you want the newer version, you can build it from **the source code**.

Open the executable file that you just downloaded and follow the

instructions (by clicking next) to complete the installation.





Install Git

🧄 Git 2.28.0	0 Setup			_		\times	
	n g the default edit n editor would you lik					>	
Use	Notepad++ as Git's	default editor			~		
	(NEW!) Notepad++	is a popular GUI (editor that can b	e used by Git.			
I	This editor is popula However, when con plugins disabled (to	figured via this op	otion, Git will call	Notepad++ with			
https://gitforv	windows.org/ ——						
			< Back	Next >	Can	cel	

There are several editors to choose from. Notepad++ is a simple editor that can be useful in general to have installed on your machine. However please note that when coding in R we use RStudio to edit our scripts.

• Git from the command line and also from 3rd-party software

(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from Git Bash, the Command Prompt and the Windov PowerShell as well as any third-party software looking for Git in PATH.







Setup a GitHub account

In order to start version controlling your projects online or even start collaborating with others you will need a GitHub account.

- For this workshop you need a GitHub account.
- If you do not have one then you need to create an account at: <u>https://github.com/</u>

You do not need to download a software, however it's useful to know that there is a GitHub desktop version: https://desktop.github.com/



Setup a GitHub account

	Search GitHub		Sign in	Sign up
Username				
Email				
Password				
Make sure it's at least 15 ch number and a lowercase lef	aracters OR at least 8 characters inc tter. Learn more.	luding a		
Sig	n up for GitHub			
	Hub", you agree to our Terms of Se n casionally send you account related			

	Join GitHub	
Create your account		
	-	
Username *		
Email address *		
Password *		
Make sure it's at le Learn more.	ast 15 characters OR at least 8 characters including a number a	nd a lowercase le
Email preferenc	25	
Send me occ	asional product updates, announcements, and offers.	
Verify your ac	count	
	Please solve this puzzle so we know you are a real person	
	Create account	
	ount, you agree to the Terms of Service. For more information a se the GitHub Privacy Statement. We'll occasionally send you a	

It is a good idea to use a username that can easily be associated with you.

For example: first name initial + last name.... <u>n</u>icolas <u>attalides</u> **nattalides**





Create a GitHub repository

Once we have our GitHub account up and running the next step is to create a new repository. This is the space where your folder and files such R scripts are stored and version controlled. It is good practice for each project that you work on to have its own dedicated repository.

- Sign in to our GitHub account
- Create a new <u>public</u> repository





You can control who can access your project by creating a **private repository**, you can also convert a public to private and vice versa



Create a GitHub repository

Search GitHub 7 Sign in Sign up		
Username	Sign in to GitHub	
Email	Username or email address nattalides	https://github.com/nattalides
Password	Password Forgot password?	This is also the URL that
Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. Learn more.	Sign in	others can go to and see all of my public repositories!
By clicking "Sign up for GitHub", you agree to our Terms of Service and Privacy Statement. We'll occasionally send you account related emails.	New to GitHub? Create an account.	



Create a GitHub repository

Note: The homepage might look different for each user depending on recent activity or account overview. The important navigator is found on







Create a GitHub repository

Select the "+" symbol that we saw earlier and click on "New repository".





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

Great repository names are short and memorable. Need inspiration? How about upgraded-doodle?

Description (optional)

 \bigcirc

0

🔔 nattalides 🕶

Public Anyone on the internet can see this repository. You choose who can commit

Private You choose who can see and commit to this repository.

Initialize this repository with:

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

Add a README file

This is where you can write a long description for your project. Learn more.

Add .gitignore

Choose which files not to track from a list of templates. Learn more.

Choose a license

A license tells others what they can and can't do with your code. Learn more.

Create repository

This is the repository name. This is also the URL address that others can use to go straight into that project.

e.g. BarcelonaR_workshop_Introduction_to_GitHub_with_R

Select the repository type, in this workshop we will work with a public repository

It is good practice to add a README file as this is the first thing a new user will get to see/read about your project. In this workshop we leave these options unticked but it is worth coming back to explore these options!



Congratulations! You have just created a GitHub repository!

📮 nattalides / Ba	rcelonaR_worksho	p_Introduction_to_GitHub_with_R	💿 Unwat
<> Code () Issu	es 🕴 Pull requests	🕑 Actions 🔟 Projects 🖽 Wiki 🕕 Security 🖂 Insights 🕸 Settings	
		Quick setup — if you've done this kind of thing before	
		Set up in Desktop or HTTPS SSH git@github.com:nattalides/BarcelonaR_workshop_Introduction_to_GitHub_with_R.git	
		Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.	
		or create a new repository on the command line	
		echo "# BarcelonaR_workshop_Introduction_to_GitHub_with_R" >> README.md git init git add README.md git commit -m "first commit" git pranch -M master git remote add origin git@github.com:nattalides/BarcelonaR_workshop_Introduction_to_GitHub_with_R.git git push -u origin master	
		or push an existing repository from the command line	
		git remote add origin git@github.com:nattalides/BarcelonaR_workshop_Introduction_to_GitHub_with_R.git git branch -M master git push -u origin master	٢
		or import code from another repository You can initialize this repository with code from a Subversion, Mercurial, or TFS project. Import code	





Create a GitHub repository

	git@github.com:nattalides/BarcelonaR_workshop_Introduction_to_GitHub_with_R.git	
t started by creating a new file or uploading	an existing file. We recommend every repository include a README, LICENSE, and .gitignore.	
or create a new repository on	the command line	-
echo "# BarcelonaR_workshop_Introduct git init git add README.md git commit -m "first commit" git branch -M master git remote add origin git@github.com git push -u origin master	tion_to_GitHub_with_R" >> README.md :nattalides/BarcelonaR_workshop_Introduction_to_GitHub_with_R.git	This is important you will see later of why
or push an existing repository	from the command line	
git remote add origin git@github.com git branch -M master	:nattalides/BarcelonaR_workshop_Introduction_to_GitHub_with_R.git	



Connect repository with R project

In order to link up our GitHub account with RStudio (desktop) we need to establish a secure connection. This is done by creating and using an **SSH key** and it only needs to be done once.

SSH stands for Secure Shell and is a "cryptographic network protocol for operating network services securely over an unsecured network".

- Create an SSH key from your RStudio desktop
- Associate the SSH key with your GitHub account
- Clone a GitHub repository to an R project

For security **do not** share with others your SSH key



Create an SSH key from your RStudio desktop

Check for Package Updates Version Control Shell
Shell
Terminal
Jobs
Addins
Keyboard Shortcuts Help Alt+Shift+
Modify Keyboard Shortcuts
Project Options
Global Options

R General	 Enable version control interface for RStudi 	o projects	
ቬ Code	Git executable:		
Appearance	C:/Program Files/Git/bin/git.exe	Browse	
	SVN executable:		
Pane Layout	(Not Found)	Browse	
Packages			
R Markdown	SSH RSA key:	View public key	1
	C:/Users/nattalides-trainer/.ssh/id_rsa		
🕪 Sweave	Create RSA Key		
ABC Spelling	② Using Version Control with RStudio		
💕 Git/SVN			
9 Publishing			
Terminal	2		
🚺 Accessibility			



Create an SSH key from your RStudio desktop

	Create RSA Key
Create RSA Key	Generating public/private rsa key pair. Your identification has been saved in C:/Users/nattalides-trainer/.ssł
The RSA key will be created at: ⑦ SSH/RSA key management	Your public key has been saved in C:/Users/nattalides-trainer/.ssh/id The key fingerprint is:
C:/Users/nattalides-trainer/.ssh/id_rsa	SHA256:NfCeaKBW146EL/nkIkx76HrpyuMQF9tQxacwRNqmSPI nattalides-trainer@ The key's randomart image is:
Passphrase (optional): Confirm:	+[RSA 2048]+
	+0+ + 00+00 = 00 B 0.* * 0
	0 E = 0 S + 0 + 0 *
	. =.0 0
Create Cancel	Close
4	
A passphrase is like a password that you will be	

A passphrase is like a password that you will be prompted to supply. For simplicity we will leave this blank.



Create an SSH key from your RStudio desktop

Options			Public Key
R General	☑ Enable version control interface for RStudio projects		Press Ctrl+c to copy the key to the clipboard
Code	Git executable:		AAAAB3NzaC1yc2EAAAADAQABAAABAQDfkR03cGbSn
Appearance	C:/Program Files/Git/bin/git.exe Browse		n3Mqj6JxFG02003JJpOuRG1mdOubqgMdjzFxzS6fs
	SVN executable:		<pre>fm/mvQ11Ue79U4YYUFwiZTVOESJdKlCSw+ZJiBhc/ gHW1vkNomr+x3/I+CV5sjx5yJMgkY1h6vOOcA0mTw</pre>
Pane Layout	(Not Found) Browse		HYZ/fCJTk1hDN1uKFh1A2JxdEcAnK7Zvi7cwm/XsQ
Packages			1HXN26VpGS/0ApbNKMSL415sXdjDGXQvkbx9Z/Ec0
R Markdown	SSH RSA key: View public key C:/Users/nattalides-trainer/.ssh/id_rsa	- 5	L1NIF0FuYr5cKbPr0yM38tDx00tvBXJY+5uzC/+e+
Sweave	Create RSA Kev		t2yUTbGG78TfFrb9eG7tU6GfrGFMR/oP4LHOBk2J9 GaEyu617c252FMTokA8Cp9t5IrVsSEXRxT6j0Sjr6
			R75 nattalides-trainer@nattalides-xps
Spelling	⑦ Using Version Control with RStudio		
🎁 Git/SVN			
- Publishing			Close
Terminal		L	
🚺 Accessibility			
		6	Press Ctrl + c to copy the key to the clipboard
	OK Cancel Apply		



Associate the SSH key with your GitHub account



Personal settings		
Profile		
Account		
Account security		
Security log		
Security & analysis	New	SSH ke
Emails		
Notifications	4	
Billing		
SSH and GPG keys	(8)	
Blocked users		9
Repositories		
Organizations		
Saved replies		
Applications		
Developer settings		



Associate the SSH key with your GitHub account





Clone a GitHub repository to an R project

It is a good idea to learn how to write code within R projects. This is because by using R projects you can easily divide your work in a structured way. Each project has its own setup such as a working directory, workspace and history.

This is even better and more efficient when it is **version controlled**!

File	Edit	Code	View	Plots	Session	Build	D	
	New F	ile					•	
	New P	roject		- (1				
	Open	File			Ctrl+0			
	Recent Files						•	
	Open	Project						
	Open	Project ir	n New S	ession				
	Recent	t Projects	5				•	
	Impor	t Dataset					•	
	Save				Ctrl+S			
	Save A	\s						
	Save A	AII			Ctrl+Alt+S			
	Publis	h						
	Print							
	Close				Ctrl+W			
	Close	All			Ctrl+Shift	+W		
	Close	All Excep	t Currer	nt	Ctrl+Alt+	Shift+W		
	Close Project							
	Quit S	ession			Ctrl+Q			



Clone a GitHub repository to an R project





Clone a GitHub repository to an R project

Set up in Deskto	op or	HTTPS SSH	git@github.com:nattalio	des/BarcelonaR_workshop_Int	oduction_to_GitHub_with_R.git		Ľ	
started by creat	ting a new file	or uploading	an existing file. We recomme	end every repository include a	README, LICENSE, and .gitignore.			
N	ew Project Wizard				4 Copy the U	IRL path to		
	Back	Clone Git Re	pository		clipboard a	and paste to		
	+	Repository URL:			Repository	URL		
	Ġ	Project director	name:		New Project Wizard			
	U	Create project a	s subdirectory of:	Browse	Back Clone	e Git Repository		You can edit the
					git@g	tory URL: github.com:nattalides/BarcelonaR_worksh	op_	project directory
					Barcel	t directory name: elonaR_workshop_Introduction_to_GitHub	wit	name and the
	Open in new ses	sion	Create Projec	Cancel		project as subdirectory of:	Browse	folder where it wil
								live on your
								machine
					Open in new session	Create Proj	ect Cancel	



Congratulations! You have just connected a GitHub repository with an R

project!

ile Edit Code View Plots Session Build Debug Profile Tools Help ▶ • 🐼 💣 •	BarcelonaR workshop Introduction to GitHub with R
Console Terminal × Jobs ×	Environment History Connections Git
<pre>-/BarcelonaR_workshop_Introduction_to_GitHub_with_RV </pre> R version 3.6.3 (2020-02-29) "Holding the Windsock" Copyright (C) 2020 The R Foundation for Statistical Computing Platform: x86_64-w64-mingw32/x64 (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.	Import Dataset • Itst • • Global Environment • • Environment is empty
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. >	 elonaR_workshop_Introduction_to_GitHub_with_R Name Size Sigtignore BarcelonaR_workshop_Introduction 218 B







This tells us that we are now inside the R project: "BarcelonaR_workshop_Introduction_to_GitHub_with_R"

This is the "new" tab that should appear that tells you that your R project has added Git functionality, i.e. it is version controlled. This tab will be very useful as we develop and version control

This is a very useful (hidden) file that communicates to our GitHub repository what files/folders should be **ignored**... these files/folders **will not** be version controlled! More on this

This is the .Rproj file associated with this project





Basic version control processes

Now that our project is version controlled let's cover some of the basic processes to version control a file, keeping things as simple as possible. A file in our repository can be one of four statuses:

- 1. Untracked: The file is not being tracked, i.e. it is not version controlled
- 2. Unmodified: The file is tracked but has not changed
- 3. Modified: The file is tracked and has been changed
- 4. Staged: The file is tracked and the changes are ready to be version controlled







Basic version control processes



Image from: https://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository

Untracked File: We need to add the file to the "staging area"

Modified File: We need to stage (add) the file to the "staging area"

git add [file]

This is the command to add the file to the "staging area".



Basic version control processes



Image from: https://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository

Staged File: We need to commit the file and all of its changes with some useful documentation or comment about the changes.

git commit -m "your_comment_goes_here" [file]

This is the command to commit the file from the "staging area".





Basic version control processes

A file has been added to the staging area and the changes have been committed. This is "stored" locally (local repository) so the next step is to **push** these changes to our remote (online) repository in GitHub. Now the file has been version controlled!







Basic version control processes

Let's say a collaborator has a copy of the up-to-date remote repository on his/her machine and decides to add a new R script to our project. They follow the steps of *adding*, *committing* and *pushing* the file to the remote repository.

Our local repository is now "behind" the current version of the project so we need to **pull** the changes (i.e. the new file) into our local repository.







- 0

Basic version control processes

We can easily do all these version control actions in RStudio using the interactive git

RStudio: Review Changes

tab window...





RStudio: Review Changes		- 🗆 X	git	add				
Changes History (no branch) - Changes State of the state	tage 🔄 Revert 💿 Ignore	🖊 Pull 丨 👚 Push						
Staged Status A Path	Commit message							
2 2 .gitignore 3 3 BarcelonaR workshop Introduct				8 RStudio: Review C				- 🗆 ×
BarcelonaR_workshop_Introduct	c					Stage 🔈 Revert 🔘 Ignore	🖊 Pu	I 丨 👚 Push
				Staged Status	▲ Path .gitignore	Commit message		
					BarcelonaR_workshop_Introdu	ctic		
	Amend previous commit	Commit						
Show Staged Instaged Context 5	ine 🔻 🔲 Ignore Whitespace							
Show O Staged O Onstaged Context 5				4		Amend previous commit		Commit
						5 line 🔻 🔲 Ignore Whitespace	Unstage All	
					-0,0 +1,4 @@ roj.user			Unstage chunk
				2 .Rh	istory			
				3 . RDa	ata serdata			
				т.ки.	seruaca			
[_						
This tells us	what are the changes							
	_							
(green high	light is added changes,							
rod bigbligh	it is deleted changes)							
	it is deleted changes)							





Image: Bistory (no branch) ▼ Image: Stage Image: Bistory (no branch) ▼ Image: Stage Image: Bistory (no branch) ▼	- C × Pull Push	Insert the commit message
Staged Status Path Commit message	30 characters	for the file to be staged
Add .gitignore file to project Add .gitignore file to project Unstage All Amend previous commit Unstage All QQ -0,0 +1,4 QQ 1 .Rproj.user 2 .Rhistory 3 .RData 4 .Ruserdata	Commit Unstage chunk	Git Commit Git Commit Git Commit Git Commit Git Commit Git Commit Git Commit Git Commit Git Commit following command and follow the instructions in your edit your configuration file: git configglobaledit After doing this, you may fix the identity used for this git commitamendreset-author 1 file changed, 4 insertions(+) create mode 100644 .gitignore Some information about your commit



 RStudio: Review Changes Changes History master - C Stage Staged Status - Path ? BarcelonaR_workshop_Introductic 	Commit message	− □ × Pull Push	git push	We can now push the staged changes from our local to our remote
				repository
Show Staged Unstaged Context 5 lir	Amend previous commit e Ignore Whitespace Unstage All	Commit	Warning: Permanently add 4' to the list of known To github.com:nattalides, h_R.git	Close /bin/git.exe push origin HEAD:refs ed the RSA host key for IP address hosts. /BarcelonaR_workshop_Introduction_ D -> master
		Some	information abou	it your push.
A master brand	ch has been created			



 RStudio: Review Changes Changes History master G Stage Staged Status Path Path BarcelonaR_workshop_Introductic 	Commit message	- C X	git pull	We can now pull any new changes from our remote to our local repository
Show Staged Unstaged Context 5 lin	Amend previous commit Amend previous commit Ignore Whitespace Unstage All	Commit	Git Pull <pre>>> C:/Program Files/G Already up to date.</pre>	Close it/bin/git.exe pull
			Some informa	ation about your pull.



Congratulations! You have just added, committed and pushed a file into

a GitHub repository within an R project in RStudio desktop!

양 master → 양1 branch ⊙	0 tags	Go to file Add file -	⊻ Code -
unknown and unknown Ad	d .gitignore file to project	2be7902 15 minutes ag	o 🕲 1 commits
🗅 .gitignore	Add .gitignore file to project		15 minutes ago
Help people interested in this repos	itory understand your project by adding a README.		Add a README







Some further notes

During this workshop we have seen the basic version control actions. Using Git (and GitHub) can be a little frustrating at first but after a while you will get used to it and there is a definite reward to all of this hard work!

- There is a lot of useful resources online if you get stuck!
- Working in branches (we have not explored this) can improve the workflow and avoid headaches.
- Any sensitive (e.g. passwords) or large files (e.g. data) should not be version controlled... use the .gitignore file to "tell" Git to ignore these.
- A very good resource on Git and GitHub can be found at: <u>https://happygitwithr.com/</u>



Thank you to our sponsors and partners!







QUIZ TIME



https://ahaslides.com/GITHUB