



CONEXIÓN REPOSITORIO GIT - RSTUDIO

En caso de no tener Git instalado, descargarlo desde la web e instalarlo:

<https://git-scm.com/downloads>

En caso de no tener una cuenta en GitHub, crear cuenta y poner la licencia educativa con el correo electrónico de la universidad:

<https://github.com/>

Si es la primera vez que descargas Git, debes introducir tus credenciales. Para ello hay dos opciones:

1. Introducir las mediante la terminal/shell:

```
git config --global user.name 'GitHubUserName'
git config --global user.email 'xxxxx@example.com'
```

2. Introducir las con el paquete "usethis" de R

```
## install.packages("usethis") # en caso de no tener el paquete
library(usethis)
use_git_config(user.name="GitHubUserName", user.email=
'xxxxx@example.com')
```

Conexión con HTTPS o SSH

HTTPS

1. Generar token en GitHub <https://github.com/settings/tokens>
2. Copiar token generado
3. En RStudio: instalar paquetes `credentials` y `gitcreds` (`install.packages("credentials"), install.packages("gitcreds")`).
4. En RStudio ejecutar `credentials::set_github_pat()` y poner el token. Después, ejecutar `gitcreds::gitcreds_set()` poner la opción 2 y poner el token.

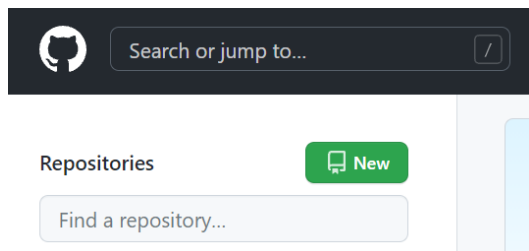
SSH

1. En Rstudio > Tools > Global Options > Git/SVN > Create SSH Key...
No poner nada en passphrase a no ser que se quiera escribir una contraseña cada vez que se haga pull/push
2. En el mismo sitio, hacer click en "View public key" y copiarla.
3. En GitHub > Foto perfil > Settings > SSH and GPG keys. Hacer click en "New SSH key". Copiar la clave pública en el recuadro y darle a "Add SSH key".



Enlazar repositorio

Comenzamos creando un repositorio en Git:



Hacemos click en “New” para crear un nuevo repositorio.

Le ponemos un nombre y una breve descripción. En este caso hacemos el repositorio público para que cualquier persona pueda acceder al material. Si se pone privado, sólo los colaboradores podrán acceder.



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 *

 carmenlancho / MDS 

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

Description (optional)

Material del Máster Data Science (URJC)

- ☒  **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.](#)

This will set  **main** as the default branch. Change the default name in your [settings](#).

Create repository



Ya tenemos el repositorio creado:

The screenshot shows a GitHub repository page for user 'carmenlancho'. At the top, it indicates 'main' branch, '1 branch', and '0 tags'. There are buttons for 'Go to file', 'Add file', and 'Code'. Below this, a commit history shows 'Initial commit' by 'carmenlancho' at '808d82d' 'now' with '1 commit'. A file list shows 'README.md' as the 'Initial commit' 'now'. The README content is displayed below, starting with 'MDS' and 'Material del Máster Data Science (URJC)'.

Le damos a “code”, seleccionamos HTTPS o SSH (en función de si tenemos la conexión hecha con tokens o con claves), y copiamos el enlace para conectarlo con R.

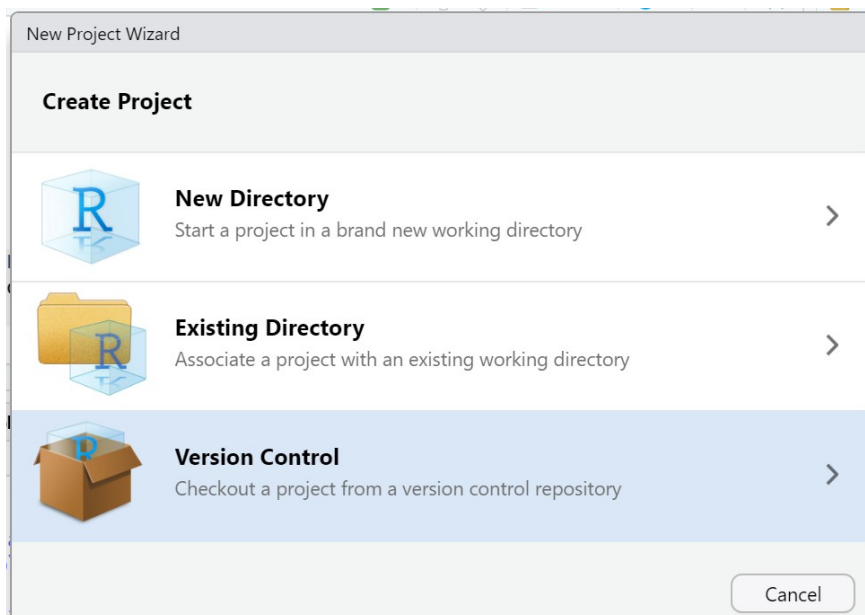
This screenshot shows the same GitHub repository page, but with the 'Code' button clicked. A dropdown menu is open, showing options to 'Clone' the repository. The 'HTTPS' option is highlighted with an orange box. Below it, the URL 'https://github.com/carmenlancho/MDS.git' is shown with a copy icon. Other options include 'SSH', 'GitHub CLI', 'Open with GitHub Desktop', and 'Download ZIP'.

Ahora vamos a conectarlo con R, para ello abrimos un nuevo proyecto:

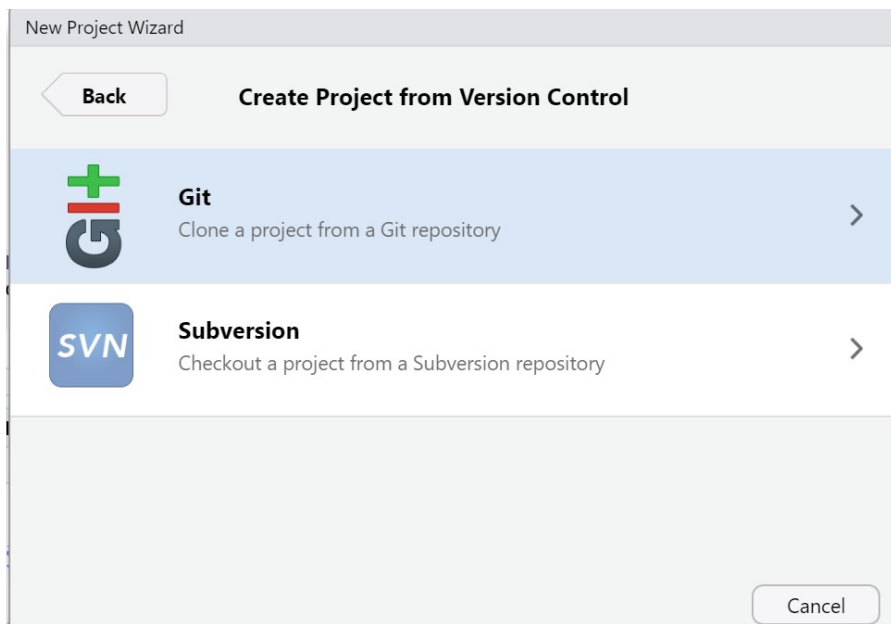
The screenshot shows the RStudio application window. The 'File' menu is open, and 'New Project...' is selected. Other visible menu items include 'New File', 'Open File...', 'Open File in New Column...', 'Reopen with Encoding...', and 'Recent Files'.



Le damos a “Version Control” puesto que lo vamos a importar de Git.



Seleccionamos Git:





En Repository URL pegamos el enlace de GitHub copiado previamente (nótese que en este caso el enlace corresponde a HTTPS). Hacemos click en “Browse” para seleccionar en qué lugar del ordenador guardamos el repositorio. Le damos a “Create Project”.

New Project Wizard

Back Clone Git Repository

Repository URL:
https://github.com/carmenlancho/MDS.git

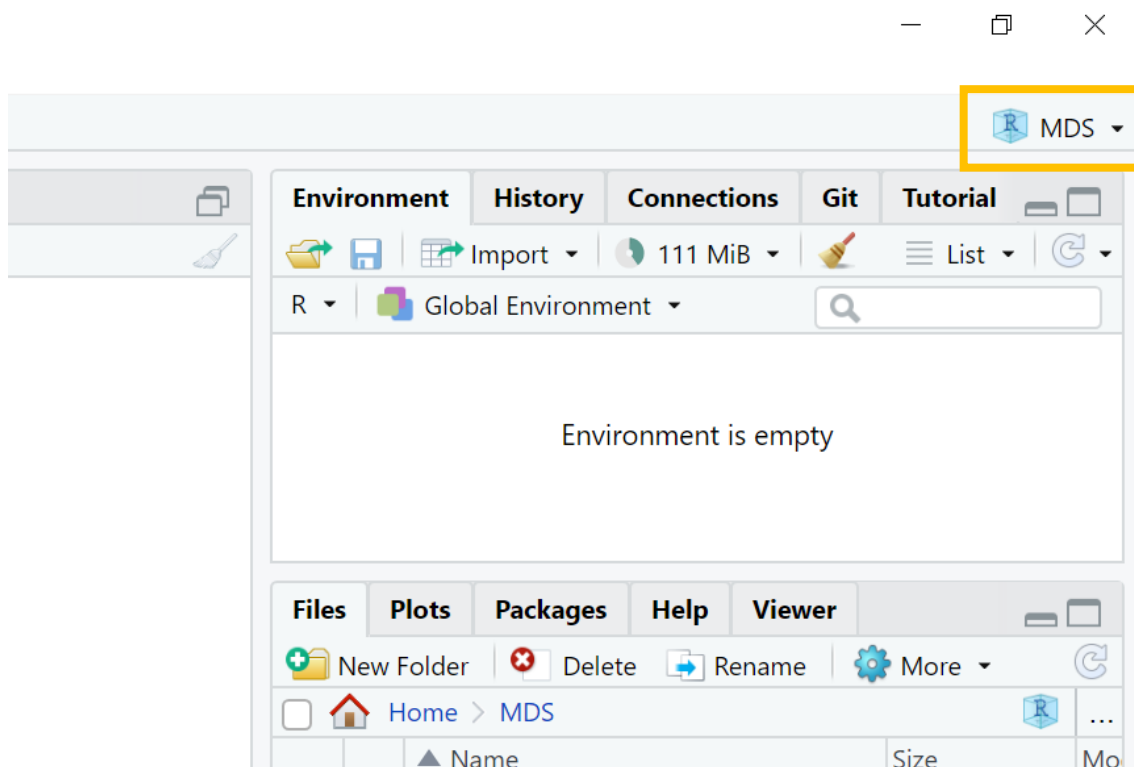
Project directory name:
MDS

Create project as subdirectory of:
~ Browse...

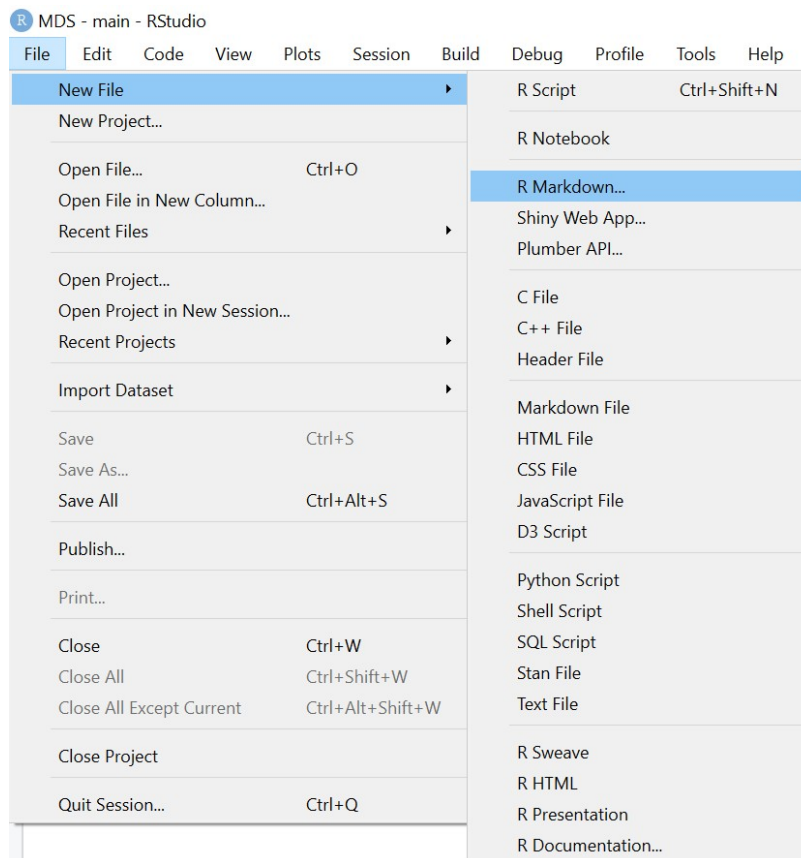
☐ Open in new session

Create Project Cancel

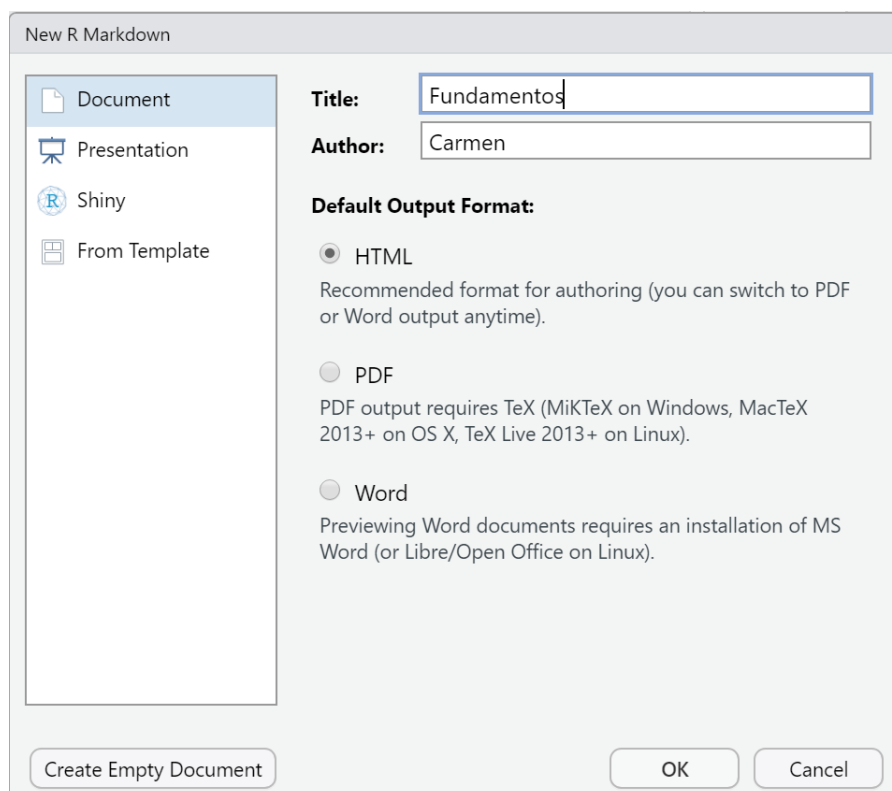
Automáticamente se abre el repositorio en RStudio:



Creamos un Rmd (RMarkdown):

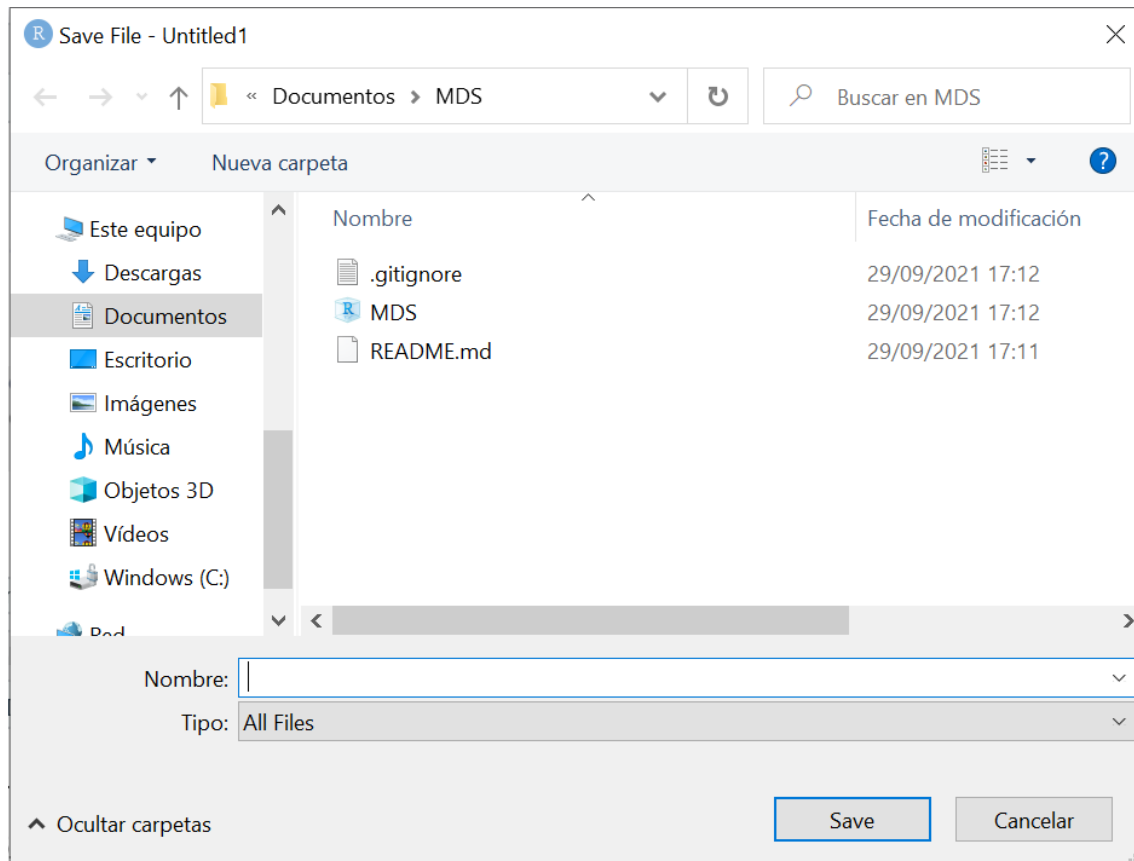


Le ponemos nombre y hacemos clic en “Ok”.

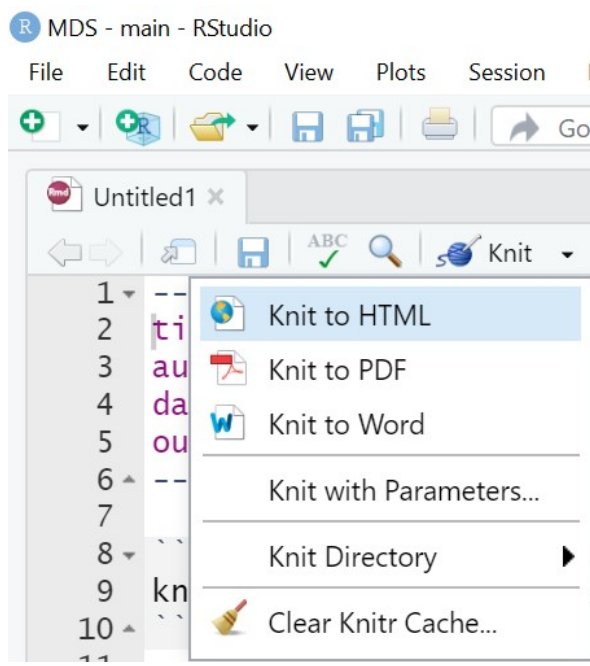




Automáticamente se crea un Rmd de ejemplo. Le damos a guardar y veremos que automáticamente se abre la carpeta del proyecto MDS:



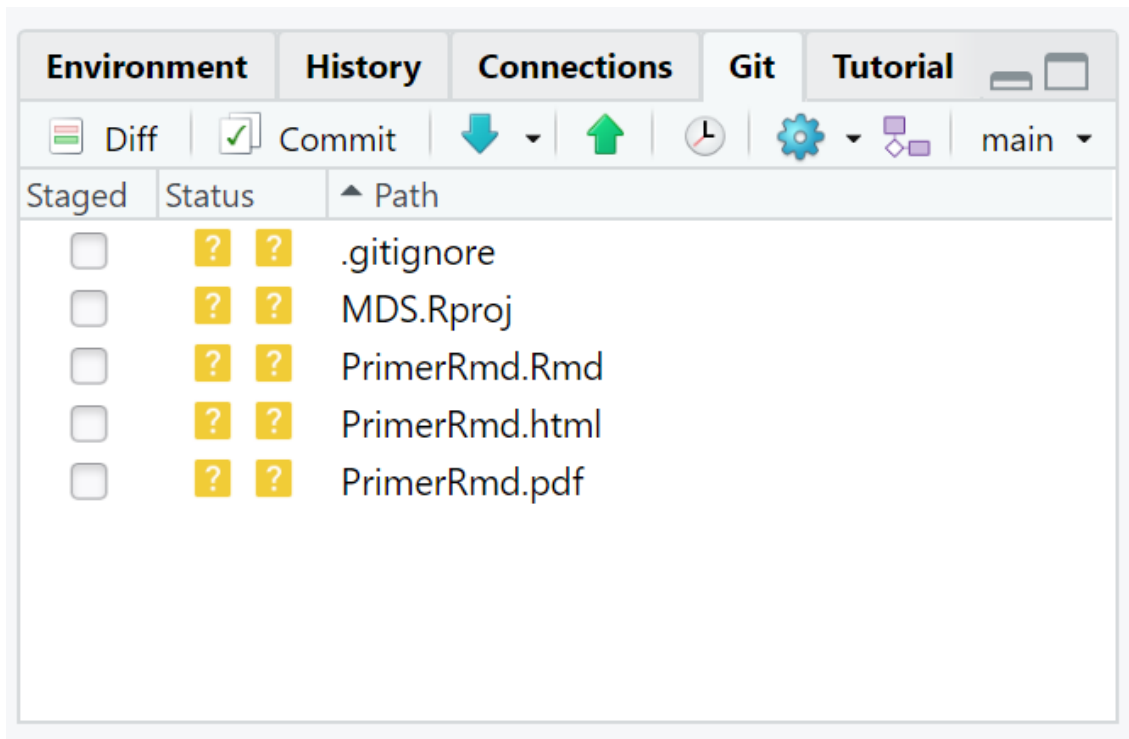
Para ejecutarlo, le damos a Knit. En función del output elegido, obtendremos un HTML, un PDF¹ o un Word. Esos archivos se irán generando también dentro de la carpeta del repositorio.



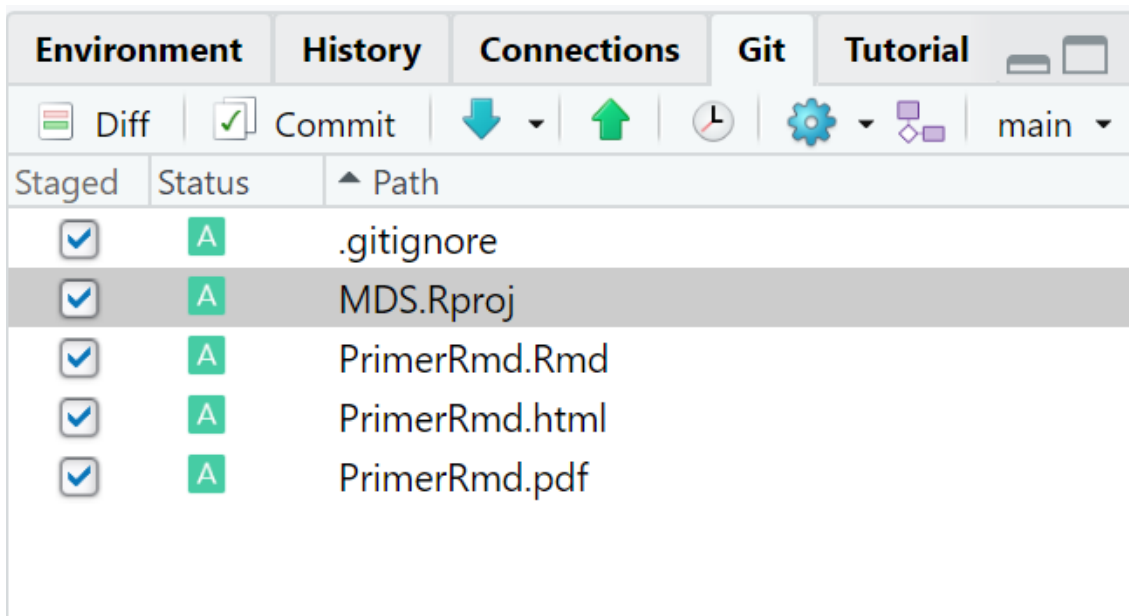
1 Para exportar un PDF es necesario tener instalado en el ordenador LaTeX. Para ello, en Rstudio debemos instalar el paquete tinytex con el comando `install.packages('tinytex')` y después ejecutar `tinytex::install_tinytex()`



Ahora vamos a subirlo a Git para guardar la versión actual. Hemos generado un PDF y un HTML de prueba. En la pestaña de Git (dentro de R) salen todos los cambios sin guardar.

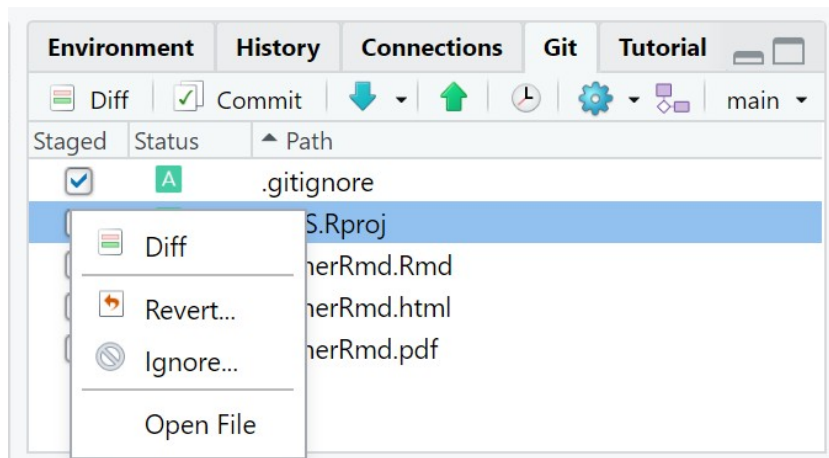


Los seleccionamos:

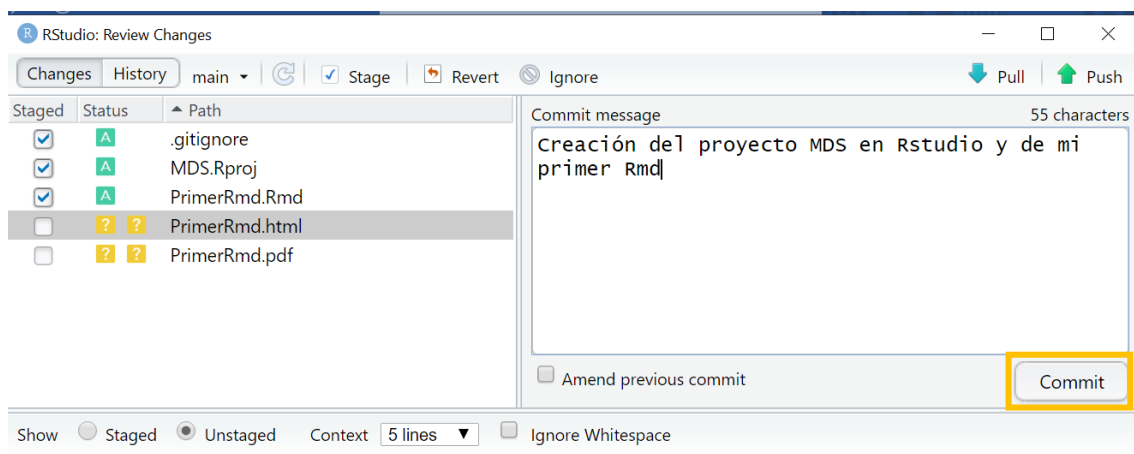




Si le damos con el botón derecho a uno de ellos nos salen las siguientes opciones: Diff, Revert, Ignore y Open File. Diff nos permite ver la diferencia entre la versión actual y la previa, revert nos permite volver a la versión anterior del archivo. Ignore es para que el archivo se meta en .gitignore.

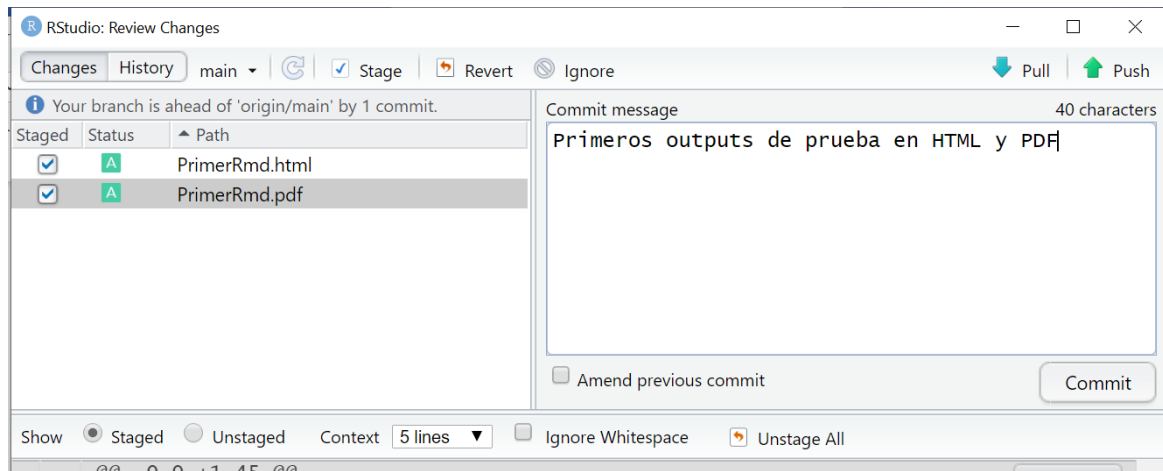


Comenzamos, como siempre, haciendo Commit:



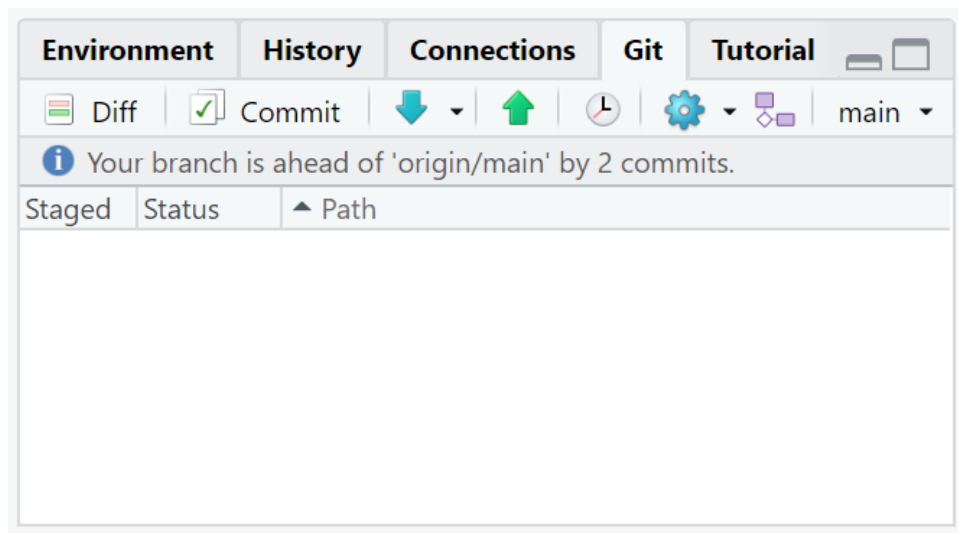


En lugar de hacer un único commit con todos los archivos, hacemos 2 commits separados para que el seguimiento del proyecto sea lo más informativo posible. Cuando más explícitos y detallados sean los commits, más fácil será el seguimiento del proyecto y volver a las versiones deseadas.



Una vez hechos los commits, ya está guardado en local.

Para subirlo a Git debemos hacer “Push” (siempre tras haber hecho commit). La flechita verde hacia arriba es “Push”, es decir, subirlo a Git. La flechita azul hacia abajo es “Pull”, es decir, bajarlo de Git.





Le damos a “Push”:

```
Git Push
>>> C:/Users/carmen.lancho/AppData/Local/Programs/Git/bin/git.exe push origin HEAD
```

Una vez aparezca lo siguiente, ya estará en el repositorio de Git actualizado.

```
Git Push
>>> C:/Users/carmen.lancho/AppData/Local/Programs/Git/bin/git.exe push origin HEAD
To https://github.com/carmenlancho/MDS.git
808d82d..179bdf2 HEAD -> main
```

Si nos vamos a Git, al repositorio MDS, vemos toda la información.

main


1 branch

0 tags

Go to file

Add file

Code




carmenlancho

Primeros outputs de prueba en HTML y PDF

179bdf2 6 minutes ago


3 commits



.gitignore

Creación del proyecto MDS en Rstudio y de mi primer Rmd


8 minutes ago



MDS.Rproj

Creación del proyecto MDS en Rstudio y de mi primer Rmd


8 minutes ago



PrimerRmd.Rmd

Creación del proyecto MDS en Rstudio y de mi primer Rmd


8 minutes ago



PrimerRmd.html

Primeros outputs de prueba en HTML y PDF


6 minutes ago



PrimerRmd.pdf

Primeros outputs de prueba en HTML y PDF

6 minutes ago



README.md

Initial commit

34 minutes ago

Antes de hacer cualquier modificación, se debe hacer Pull del repositorio para asegurar que vamos a trabajar sobre la última versión.



Trabajar de forma colaborativa

Si varias personas van a trabajar en un proyecto, se crea un único repositorio. Es decir, una de las personas crea el repositorio desde su cuenta de Git e invita al resto de componentes. Esto se hace de la siguiente manera. El creador debe irse al repositorio y darle a “Settings”.

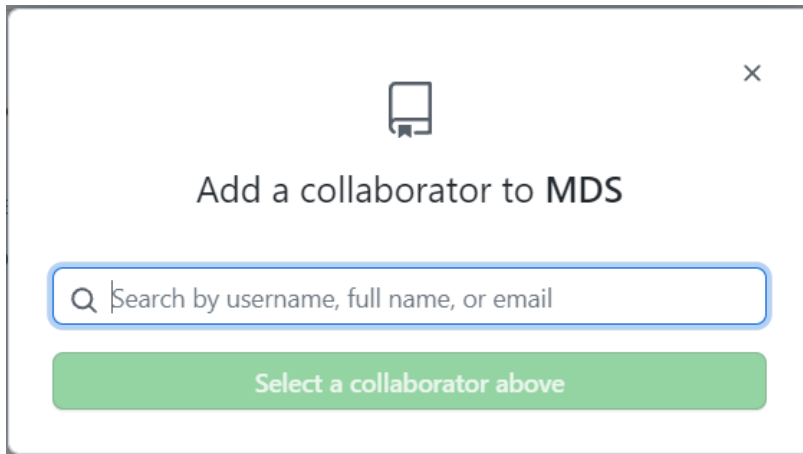
The screenshot shows the GitHub interface for a repository named 'MDS' by user 'carmenlancho'. The repository is public. The 'Settings' tab is highlighted in the top navigation bar. Below the repository name, there are buttons for 'Go to file', 'Add file', and 'Code'. A table lists the repository's files and their commit history:

File	Description	Commit Time
FAD	Rmd sin logos	4 hours ago
.gitignore	Creación del proyecto MDS en Rstudio y de mi primer Rmd	2 days ago
MDS.Rproj	Creación del proyecto MDS en Rstudio y de mi primer Rmd	2 days ago
README.md	Initial commit	2 days ago

Después hacer click en “Manage access” y en “Add people”.

The screenshot shows the 'Manage access' page in GitHub. On the left, a sidebar lists various settings, with 'Manage access' selected. The main content area is titled 'Who has access' and shows two sections: 'PUBLIC REPOSITORY' (which is public and visible to anyone) and 'DIRECT ACCESS' (which currently has 0 collaborators). Below these sections, there is a 'Manage access' section with a message: 'You haven't invited any collaborators yet'. A green 'Add people' button is visible in the top right corner of the 'Manage access' section.

Aparecerá la siguiente ventanita en la que se debe escribir el nombre del usuario de Git que queremos que colabore en el proyecto.



A dicho usuario le llegará un correo con la invitación. Una vez aceptada, le aparecerá el repositorio en su cuenta Git y podrá clonarlo.

Referencias

<https://happygitwithr.com/index.html>