SDG GitHub Repository

Quick user guide

Table of contents

[1. Introduction 3](#_Toc83289485)

[2. Getting started 3](#_Toc83289486)

[2.1. Create an account in GitHub 3](#_Toc83289487)

[2.2. Download GitHub desktop 3](#_Toc83289488)

[2.3. Download Git 3](#_Toc83289489)

[2.4. Create a new repository 3](#_Toc83289490)

[3. Integrating GitHub and RStudio 3](#_Toc83289491)

[3.1. Create a new project in R Studio 3](#_Toc83289492)

[3.2. Create a SSH RSA key 4](#_Toc83289493)

[4. Working simultaneously with RStudio and GitHub 4](#_Toc83289494)

[4.1. Create and save a new script within the current project 5](#_Toc83289495)

[4.2. Create a new branch 5](#_Toc83289496)

[4.3. Commit and push your changes 5](#_Toc83289497)

[4.4. Create a pull request 5](#_Toc83289498)

[4.5. Merge a pull request 6](#_Toc83289499)

[5. Uploading and synchronising files 6](#_Toc83289500)

[5.1. Move your file 6](#_Toc83289501)

[5.2. Synchronise your file 7](#_Toc83289502)

1. Introduction

The aim of this document is to offer a step-by-step guide to quickly get started with GitHub and its integration with RStudio. GitHub is the most popular code hosting platform for version control and collaboration. GitHub allows to track changes when multiple people collaborate on a project. This platform can also be integrated with other tools, such as RStudio. RStudio provides an interface for version control using Git. When working with Git, it is important to differentiate between the remote repository, which is essentially the web version of GitHub and the local repository, which is the repository that is stored on the computers of various team members and that can also be visible in the desktop version of GitHub.

1. Getting started
   1. Create an account in GitHub

The first thing to do is to create an account in the main [website](https://github.com/) of GitHub.

* 1. Download GitHub desktop

The desktop version allows to update the remote repository when adding files in the local repository. The GitHub desktop version can be downloaded [here](https://desktop.github.com/).

* 1. Download Git

Download Git to connect GitHub with RStudio. Git can be downloaded [here](http://git-scm.com/).

* 1. Create a new repository

Once you have created your GitHub account, you can create a repository (in the web version). In a repository a variety of projects and files can be stored either to collaborate with or to make their content available to a larger audience. A step by step guide to create a repository can be found [here](https://docs.github.com/es/get-started/quickstart/create-a-repo). Please note that it is considered a good practice to add README.me file to describe the project and the files that are in the repository.

1. Integrating GitHub and RStudio
   1. Create a new project in R Studio

Choose File > New project and then choose Version Control > Git. In the Repository URL paste the link of the repository. To copy the link of the repository go the web version, select the repository and then click on the green button ‘Code’ and copy the HTTPS link. Afterwards, choose a folder in which you want to save your local repository by clicking on the Browse button (this will automatically create a clone folder, that will be visible in GitHub desktop). Lastly, click on Create Project.

|  |  |
| --- | --- |
|  |  |
|  |  |

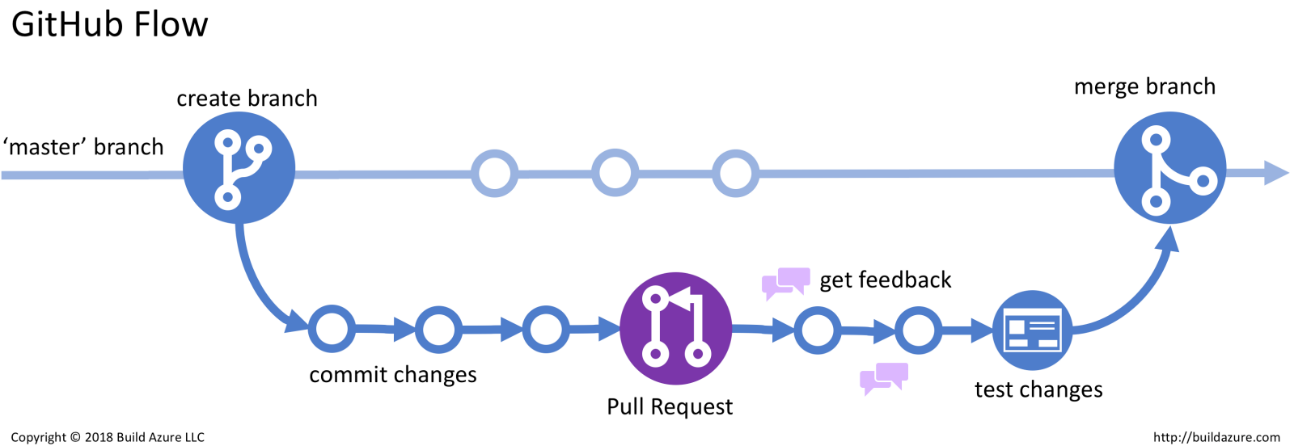
* 1. Create a SSH RSA key

In RStudio go to Tools > Global Options > SSH and GPG keys and click on Create RSA key and then Create (no need to set a passphrase) and copy the key. Afterwards, in the web version of GitHub go to Account settings > SSH and GPG Keys and copy the Key, add a title (e.g. RStudio) and click on Add SSH key.

|  |  |
| --- | --- |
|  |  |

1. Working simultaneously with RStudio and GitHub

When team members need to work individually in a R script, they need to create new branches in order to avoid a modification of the master branch. When changes are made in new branches, they need to be committed and pushed in order to create a pull request, so that they can be merged with the master branch. The diagram bellow shows a simplified work flow of the branching system of GitHub. The following steps demonstrate how to save an R script and make changes in a secondary branch.

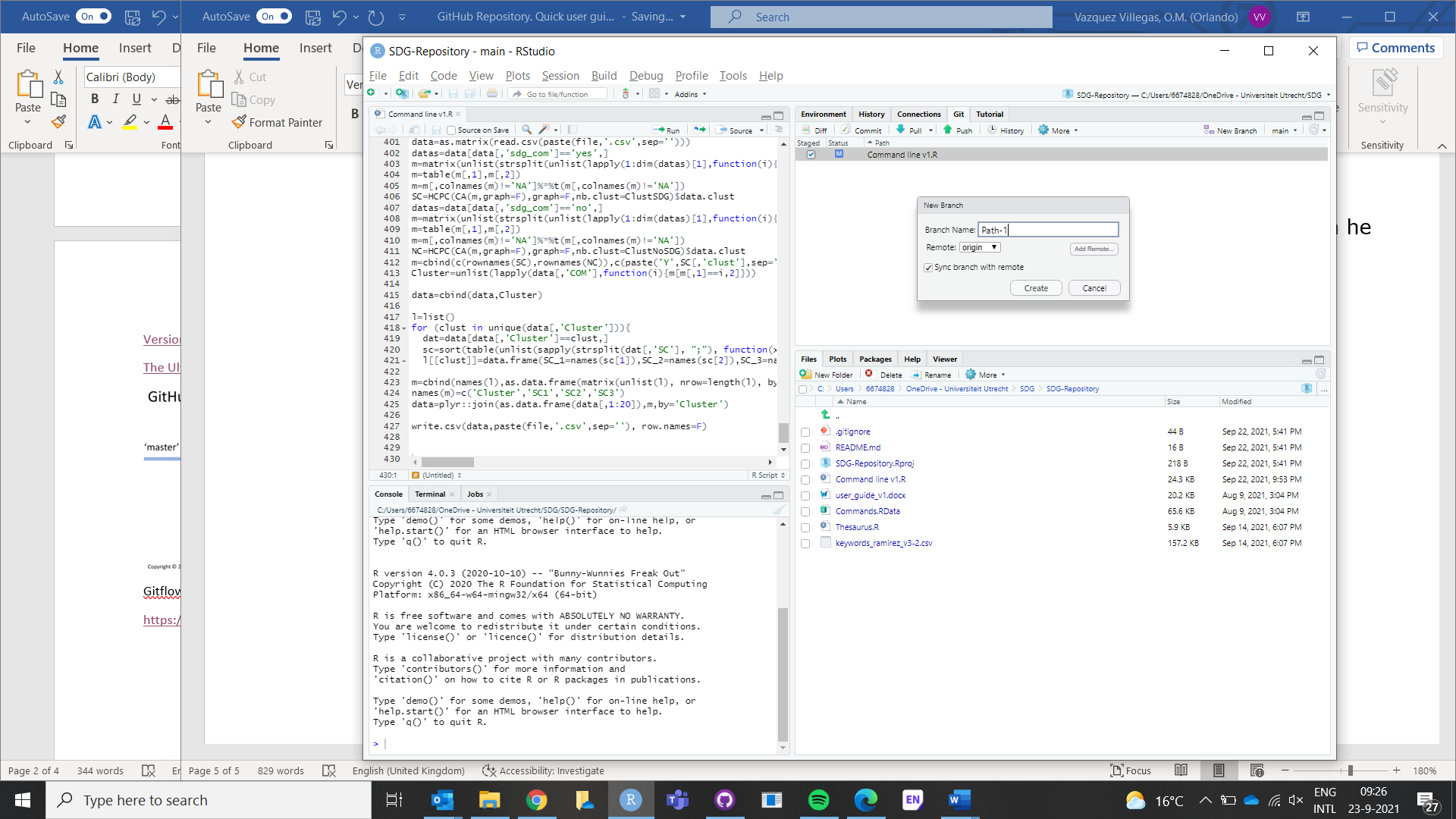
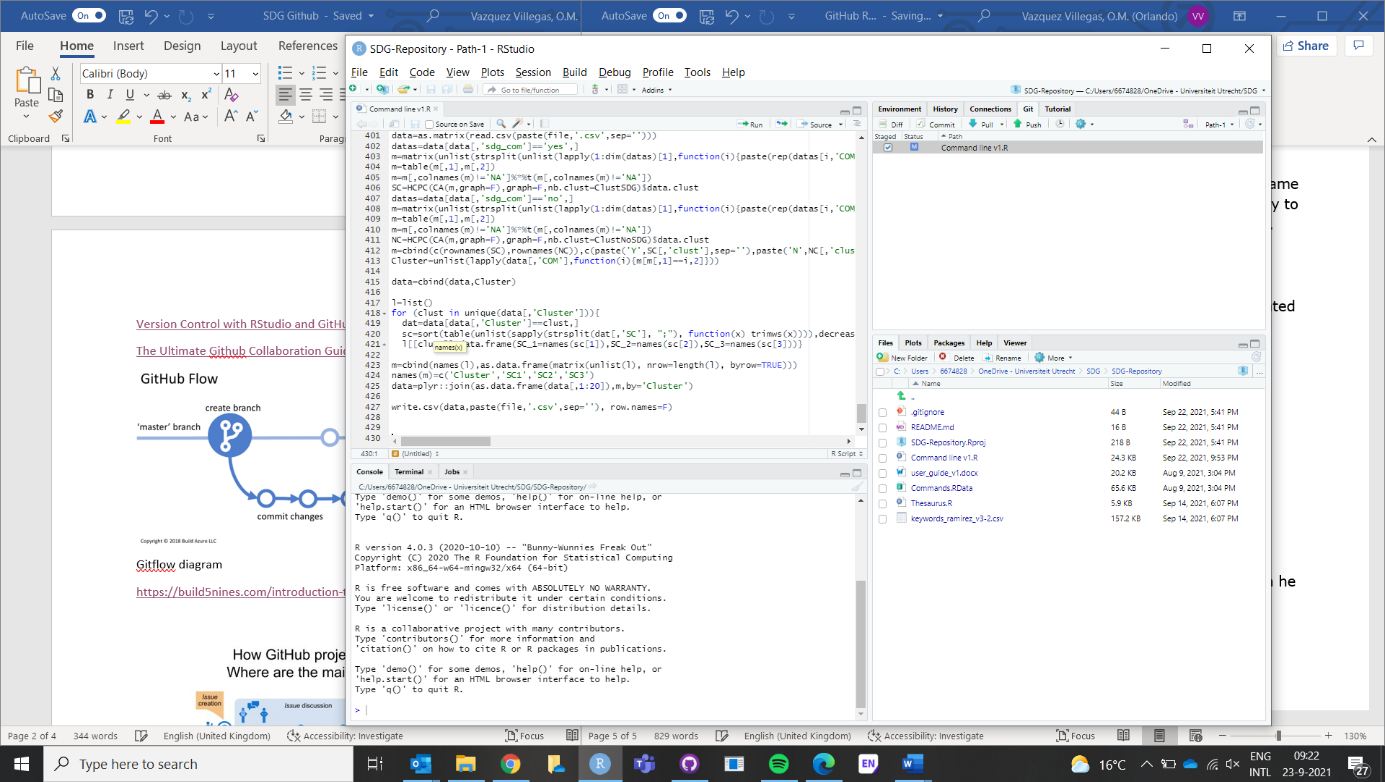


* 1. Create and save a new script within the current project

Script files that are create within the current project can be saved in the folder (holding the name of the repository) that was automatically created in step 3.2. To save the script, it is necessary to give a name to the file and either press the save button or go the File menu and choose Save.

* 1. Create a new branch

When a script file needs to be modified, it is recommended to create a new branch in order to commit and push the new changes. When you create a branch off the master branch, you’re making a copy, or snapshot, of master as it was at that point in time. To create a new branch press the New Branch button located in the Git panel. When the new branch is created, its name appears on the top-left side of the RStudio window.

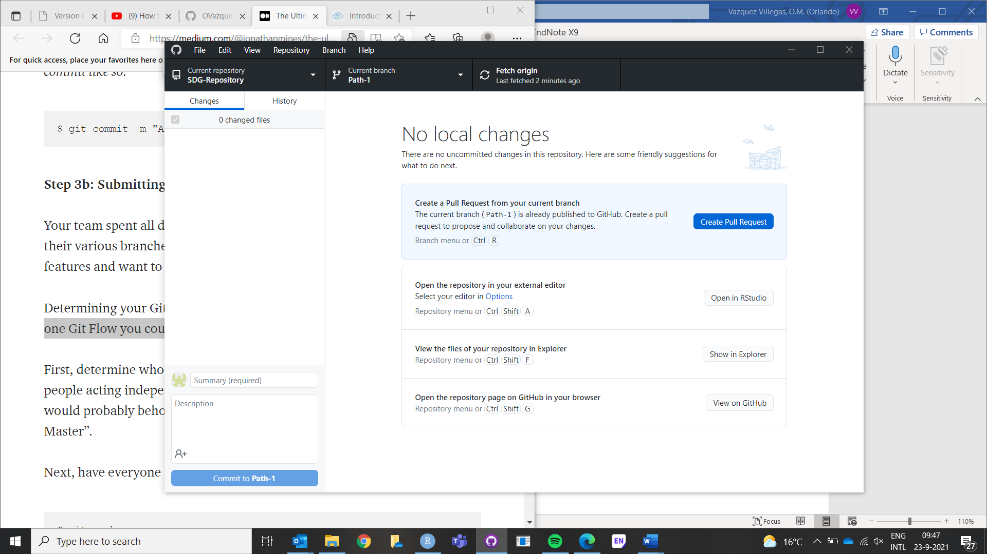


* 1. Commit and push your changes

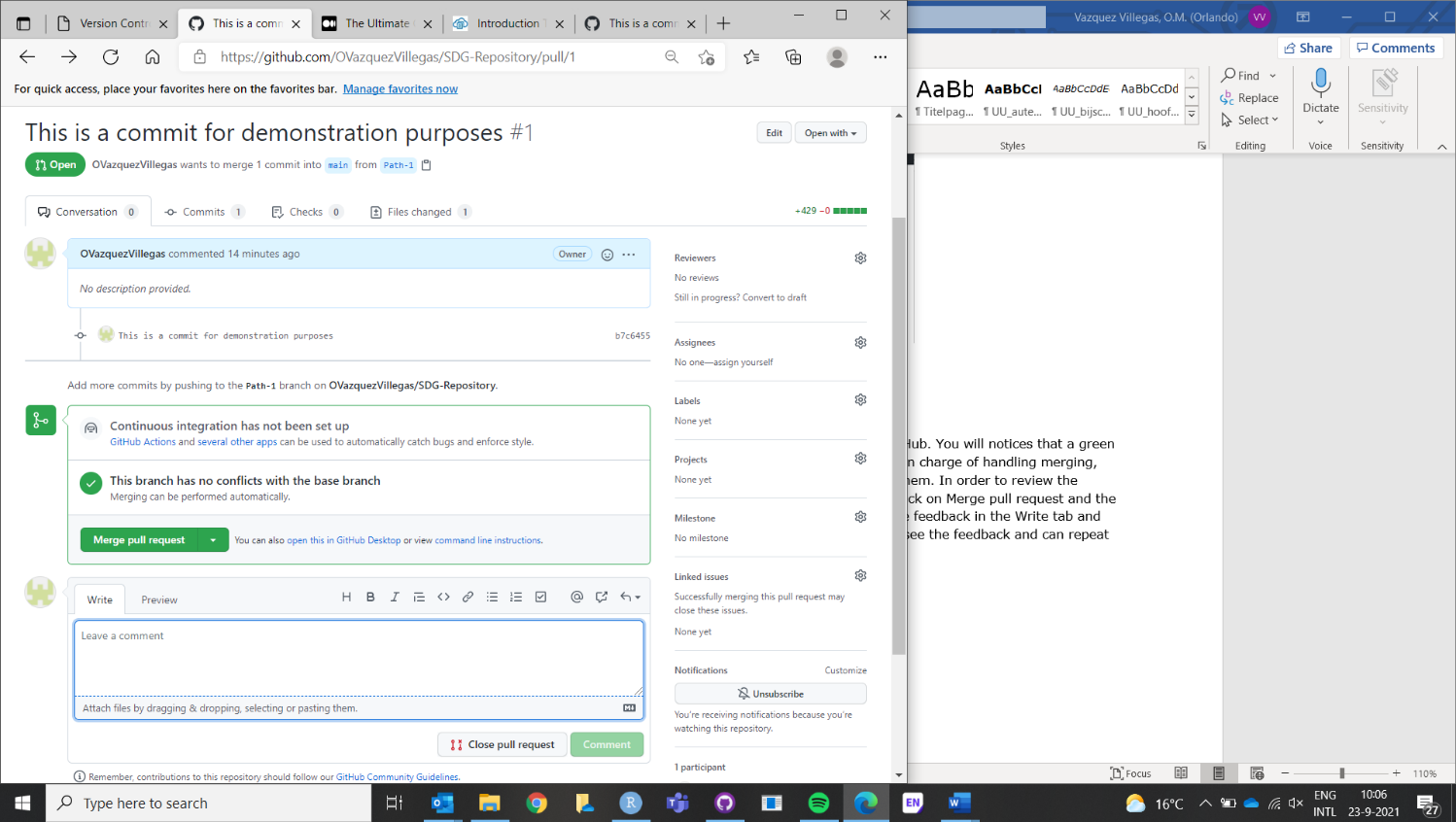
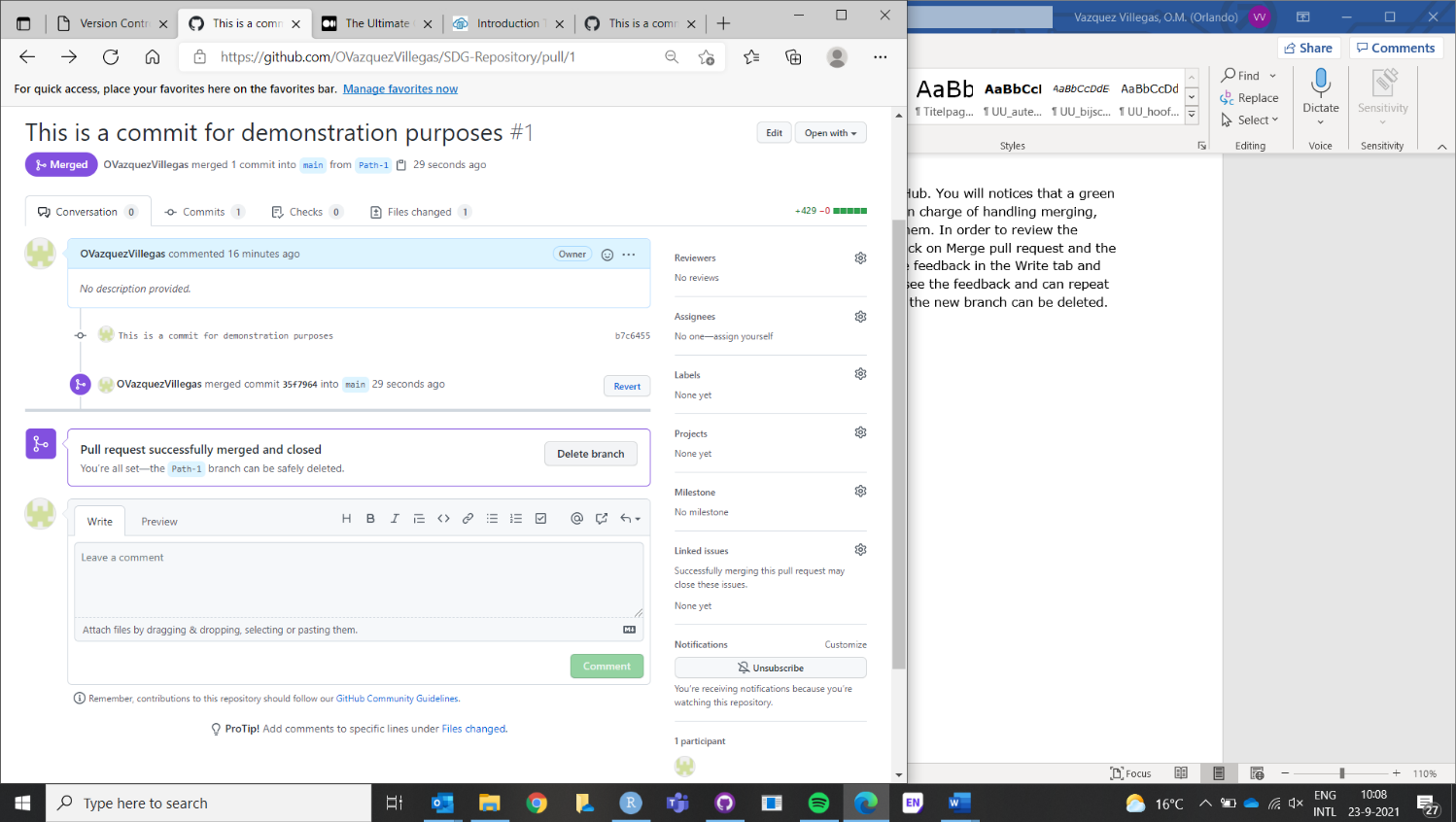
Having made in changes in the script file, you need to save your script and the changes will be listed in the Git panel. In order to commit those changes, push the Commit button and a new windows will pop up with the new changes. Write any comment related to the new changes and press the Commit button again. In order to make this changes visible in the desktop version of GitHub, press the push button.

|  |  |
| --- | --- |
|  |  |

* 1. Create a pull request

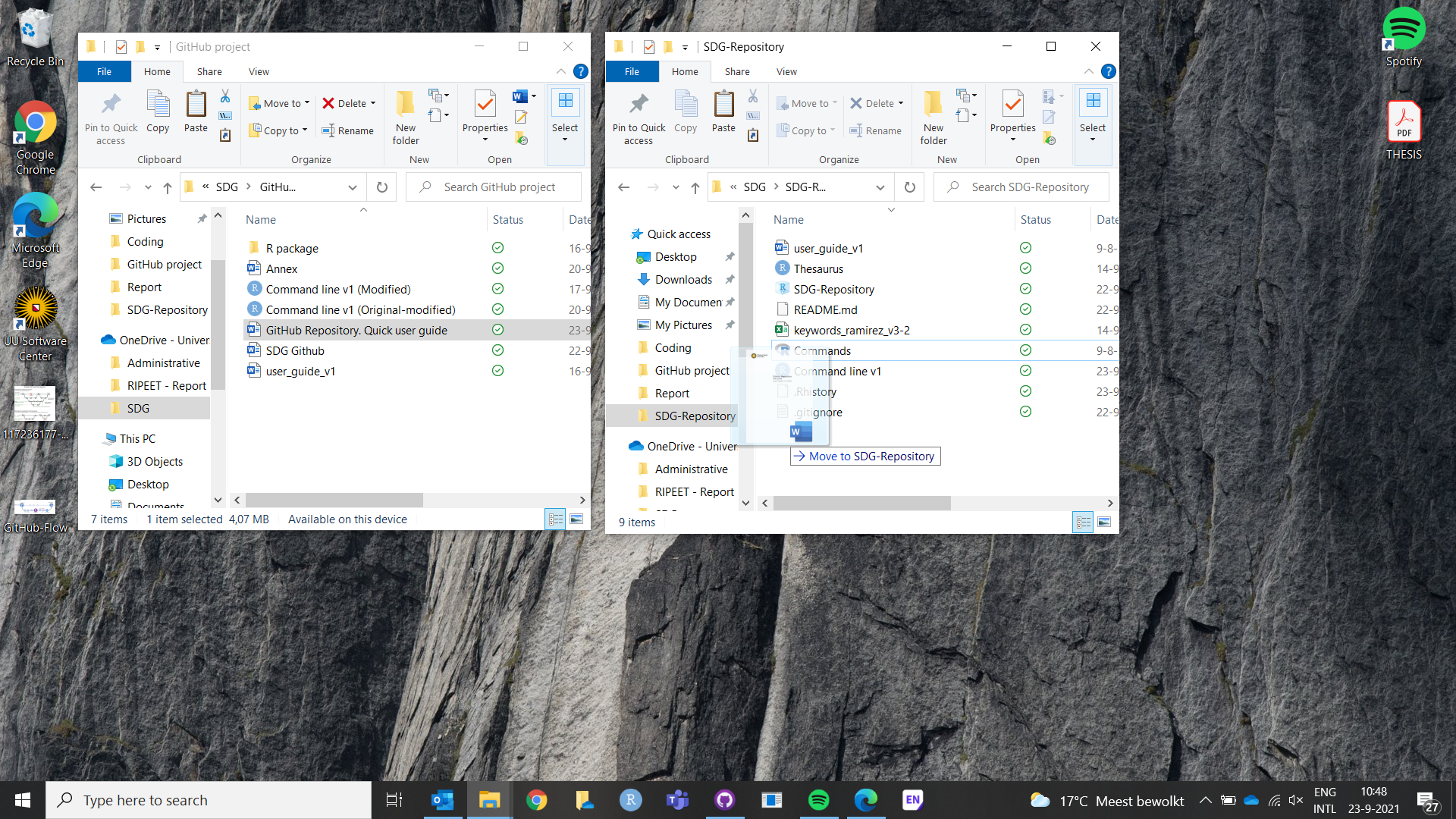
Once the changes have been pushed, they will be visible in GitHub desktop and the next step is to push the button Create a Pull Request. In doing so, the person in charge of handling merging will be able to see your pull request and approve or request your changes to be merged with the master branch.

* 1. Merge a pull request

Now go back to your remote repository in the web version of GitHub. You will notices that a green button saying Compare & pull request appears. Here the person in charge of handling merging, needs to check the changes, and either approve them or reject them. In order to review the changes, click on the Files changed tab. If the changes are ok, click on Merge pull request and the click in Confirm merge. If you can’t approve the changes, provide feedback in the Write tab and click on Comment. Then, the person that made the changes will see the feedback and can repeat steps 4.3 and 4.4. When the pull request has been merged then, the new branch can be deleted.

1. Uploading and synchronising files
   1. Move your file

Having made in changes in the script file, you need to save your script and the changes will be When new files (no RStudio files) need to be stored in the repository, they can be added directly to the local repository (the local folder that was originally created in step 3.1).



* 1. Synchronise your file

Once the file has been moved to your local repository, open GitHub Desktop and you will notice that a message saying ‘1 changed file’ appears in the left top corner. Press in Commit to main and add a comment, so that the file can be available in the remote repository. When a change is made in a file (e.g. a Word document), GitHub Desktop will automatically detect his change and it has to be committed to update the file of the remote repository.

