**Linking Rstudio and GitHub/Git**

ALL PLAGIARISED

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

In that same RStudio option window, click “Create RSA Key” and when this completes, click “Close.”

Following this, in that same window again, click “View public key” and copy the string of numbers and letters. Close this window.

Graphical user interface, application

Description automatically generated

Note: ignore SVN for now

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

GitHub and RStudio are now linked. From here, we can create a repository on GitHub and link to RStudio.

**Create a Personal Access Token (PAT)** – not plagiarised

At some point we will need to push our file to the repository. Doing so, GitHub used to ask for your GitHub password when authenticating Git operations. Today, GitHub asks for your PAT. We can create a PAT by following this gist (Gists are guides hosted on GitHub that help clarify a concept):

<https://gist.github.com/Z3tt/3dab3535007acf108391649766409421>

The code presented in the gist we will type in the Rstudio console…

Configure git with Rstudio:

##Set your GitHub username and email:

### usethis::use\_git\_config(user.name = "YourGitHubName", user.email = "yourGitHubEmail@mail.com")

### Example: usethis::use\_git\_config(user.name = "akis-h", user.email = "ioakeim.h@gmail.com")

### ## create a personal access token (PAT) for authentication:

### usethis::create\_github\_token()

### This will open a new window on GitHub where you will create your PAT. After you have created it, save it somewhere for future reference. Back to R.

## set personal access token:

credentials::set\_github\_pat("YourPAT")

Once you have run this code, you will be asked to provide your PAT. Paste it into the window that has opened and click ok.

Create a new repository and edit it in RStudio

On GitHub, create a new repository (github.com > Your Profile > Repositories > New). Name your new test repository and give it a short description. Click Create repository. Copy the URL for your new repository.

Graphical user interface, application, Teams

Description automatically generated

**Location of the “Repositories” link on your profile**

Graphical user interface, text, application

Description automatically generated

**Note:** when you create the repository switch from HTTPS to SSH.

**Graphical user interface, text

Description automatically generated**

**Creating a new repository on GitHub**

In RStudio, go to File > New Project. Select Version Control. Select Git as your version control software. Paste in the repository URL from before, select the location where you would like the project stored. When done, click on “Create Project”. Doing so will initialize a new project, linked to the GitHub repository, and open a new session of RStudio.

Graphical user interface, application

Description automatically generated

**Creating a version controlled project on RStudio**

Graphical user interface, application, Teams

Description automatically generated

**Cloning your Git repository to RStudio**

Create a new R script (File > New File > R Script) and copy and paste the following code:

print("This file was created within RStudio")

print("And now it lives on GitHub")

Save the file. Note that when you do so, the default location for the file is within the new Project directory you created earlier.

Graphical user interface, text, application

Description automatically generated

**Saving your first script for this project**

Once that is done, looking back at RStudio, in the Git tab of the environment quadrant, you should see your file you just created! Click the checkbox under “Staged” to stage your file.

Graphical user interface, application

Description automatically generated

**All files that have been modified since your last pull appear in the Git tab**

Click “Commit”. A new window should open, that lists all of the changed files from earlier, and below that shows the differences in the staged files from previous versions. In the upper quadrant, in the “Commit message” box, write yourself a commit message. Click Commit. Close the window.

Graphical user interface, application

Description automatically generated

**Commiting your R Script to the repository!**

So far, you have created a file, saved it, staged it, and committed it. If you remember your version control lecture, the next step is to push your changes to your online repository. Push your changes to the GitHub repository.

Graphical user interface, application

Description automatically generated

**How to push your commit to the GitHub repository**

Go to your GitHub repository and see that the commit has been recorded.

You’ve just successfully pushed your first commit from within RStudio to GitHub!