**Github info**

Contents:

[1.Simple SSH key check 2](#_Toc503199024)

[2.Git respository 3](#_Toc503199025)

[3.Main commands 4](#_Toc503199026)

[4.Set up git from scratch 5](#_Toc503199027)

# 1.Simple SSH key check

In order to verify your hardware (read: laptop) you have to generate a set of keys and then share the public key with your github account. The keys are defaulted in this folder:

C:/Users/THINK\_SUBJECT22/.ssh

C:/Users/COMPUTER\_NAME/.ssh

Note: if you change your computer name, or if you upgrade your windows version the ssh keys will no longer work, and you will have to create new ones.

https://help.github.com/articles/reviewing-your-ssh-keys/

1 Check the existing keys: in settings on github

2 if they do not match generate new ones: activate ssh-agent:

eval "$(ssh-agent -s)"

3 generate new keys:

ssh-keygen -t rsa

4 key is generated in default folder: C:/Users/COMPUTER\_NAME/.ssh

5 go to this folder and copy the file in id\_rs.pub

1. click new ssh- key in github + paste the id\_rsa.pub in the field ‘Key’

Now the new key should be set up, following steps are to check if it worked:

1. in git bash:

ls -al ~/.ssh

1. check if you can push and pull to git from the device, from a git folder (see next chapter how to set up git repository)

check via windows shell outside of git bash:

cd C:/Users/THINK\_SUBJECT22/.ssh

(to list the files in the directory):

dir

simply visually check if this files is present: **“id\_rsa.pub”**

# 2.Git respository

In order to get started with Git, a folder on your local machine will have to be connected to git. You can approach this from two sides: either create the folder locally and push to a repository on your github account: or create the folder on the github account and pull it into your local environment, this guide will follow first approach.

NOTE: slashes are annoying: in git bash they have to be forward slashes: from windows they have to be change from back to forward

1. Create folder on desktop you want to connect to git
2. Copy the path to use in git bash, navigate here
3. this initializes the local folder to become a repository:

Git init

1. In git bash add the remote origin to the local folder using (make sure you are in the repository to which you want to add the repository)

git remote add origin [git@github.com:brabagaza/new\_repos\_name.git](mailto:git@github.com:brabagaza/new_repos_name.git)

1. **NOTE: ‘origin’ is now the name of the remote path: this is important in syncing the local and remote copies**
2. Create the repository on your github account with the same name
3. It should now be set up: try adding a file, commiting and pushing

(if error origin already exists: check origin:

git remote –v

git remote rm origin

Add the correct remote path using code above ^^

The other way around:

**Pull an existing repository into your local machine**

# 3.Main commands

There are three main commands to update the remote repository.

Git add name\_of\_file.txt

Git commit –m ‘some message for the commit’

Git push origin master

This all occurs while you are in the correct repository and via git bash

If the file does not exist, check which files are there with

ls

go to a subfolder by using

cd sub\_folder

go to a folder back using

cd ..

NOTE: there is a space between the cd and the dots, otherewise it does not work

In order to check which changes are staged for commiting + pushing:

Git status

this line pushes all the work from local machine to the git hub online repository:

git push -u origin master

this will force the local copy onto the remote (github hosted) counterpart: -- **so you will lose commits for the online respository:**

git push -f origin master

*this ^^ may be needed in the beginning if the repository is created and the files are not merged.*

NOTE: If git says up to date: it means up to date with the branch that it is tracking: this might mean the local branch on your laptop: This means that the remote version could be different: but git only reports on the status of you local copy of the branch.

see:

https://stackoverflow.com/questions/27828404/why-does-git-status-show-branch-is-up-to-date-when-changes-exist-upstream

in order to check with remote branches:

git fetch

git status

hmm, what is this?:

git fetch origin master

git diff

show all changes in the repository

output looks like this:

diff --git a/report.txt b/report.txt

index e713b17..4c0742a 100644

--- a/report.txt

+++ b/report.txt

@@ -1,4 +1,4 @@

-# Seasonal Dental Surgeries 2017-18

+# Seasonal Dental Surgeries (2017) 2017-18

TODO: write executive summary.

a --> first version

b --> second version

lines added prefixed with '+'

lines removed prefixed with '-'

@@ - shows where changes have been made

@@ -1,4 + 1,4 @@ = lines 1 to 4 have been replaced by different lines

now it shows the line by lines changes

NOTE: git diff file\_name.csv --> show specific change of this one file

# 4.Git and R

<http://r-bio.github.io/intro-git-rstudio/>

In r studio: tools--> project options change version control to git

--> now we have a git pane on the top right

http://r-pkgs.had.co.nz/git.html

# 4.Set up git from scratch

# 5.Special problems (e.g force push)