

**Let's get started with Git!**

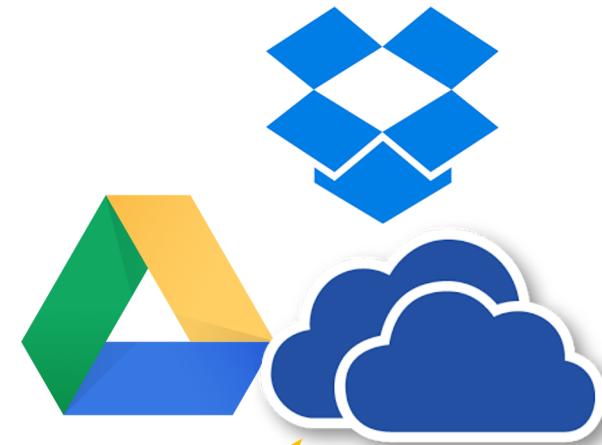
# To contribute to a GitHub project, follow these 5 steps

1. Add (existing or new) project to Git on your local computer
2. Committing to your local Git repository
3. Create a new repository on GitHub
4. Setup credentials to connect GitHub to your local computer
  - a. Use password
  - b. Generate RSA key
5. Push commits from local Git repository to GitHub



# Git and Dropbox are not best friends

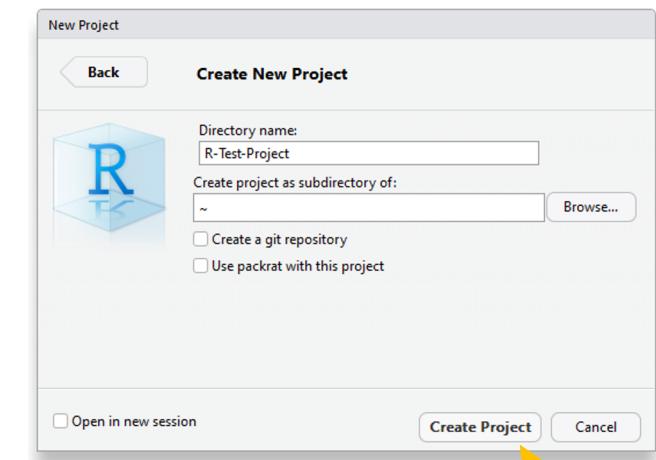
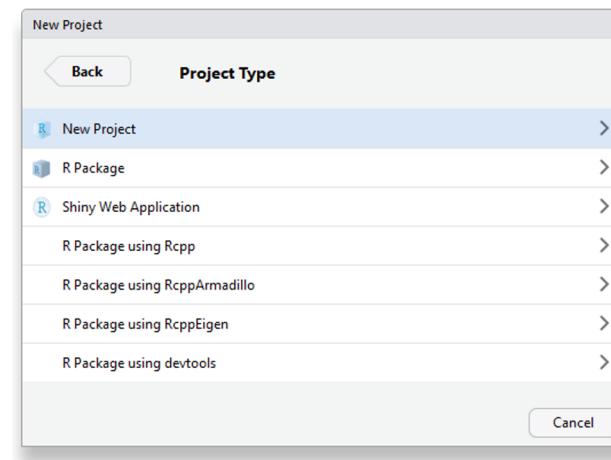
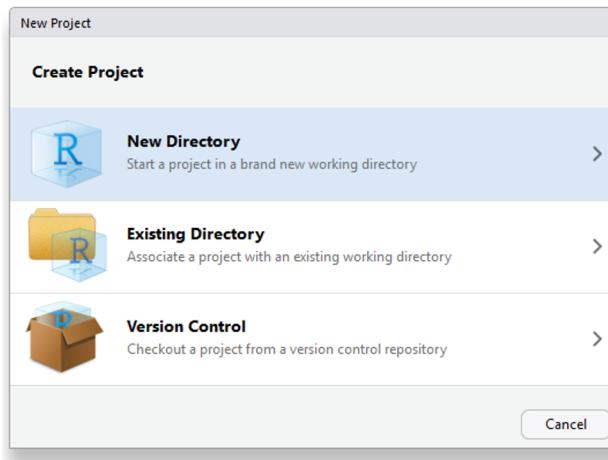
Do not place Git repositories on Dropbox, Google Drive, OneDrive, ...



Document clouds  
will destroy Git  
repos.

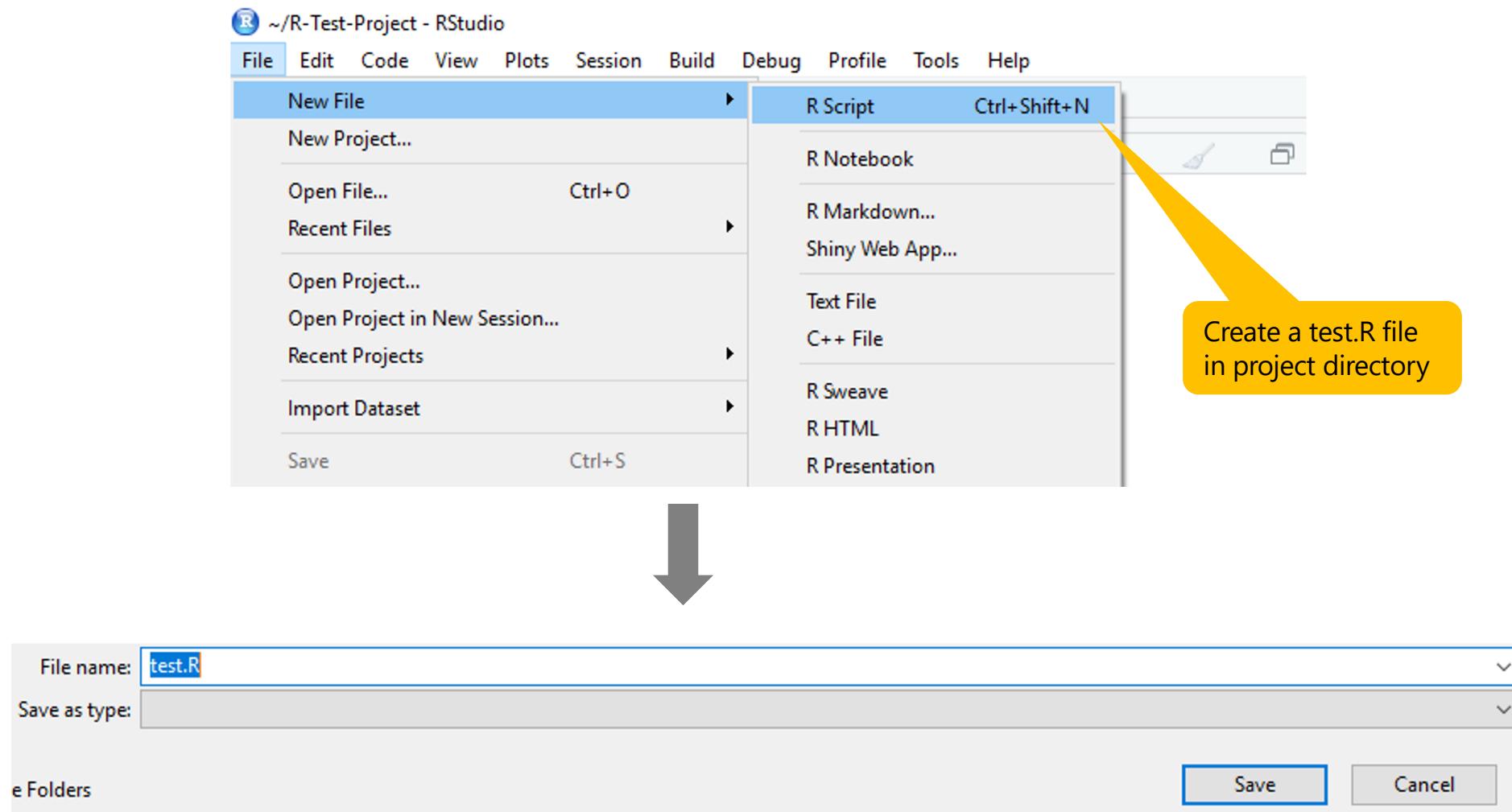
# Step 1: Add project to Git on your local computer (1/3)

Create a New Project in RStudio ("File" -> "New Project"):



Then click "Create Project"

# Step 1: Add project to Git on your local computer (2/3)



# Step 1: Add project to Git on your local computer (3/3)

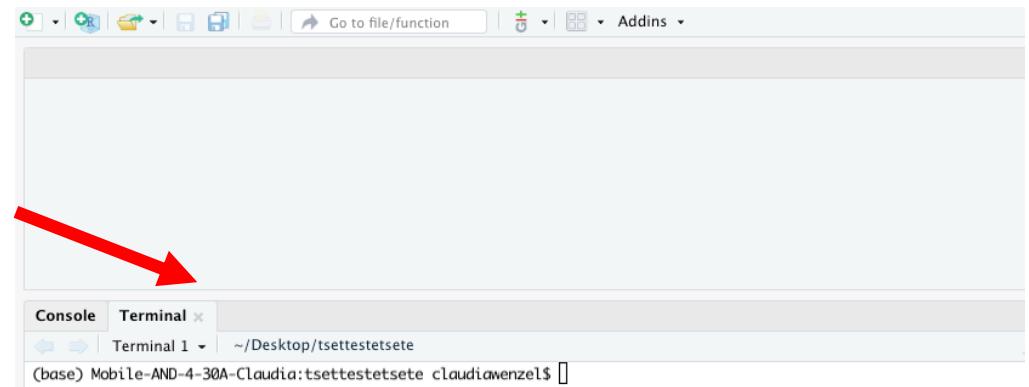
Make sure to run the command line within your project directory

1. Navigate to the project directory via the command line.

```
lweibel@lweibel-PC MSYS /c/Users/lweibel/Documents/R-Test-Project
$
```

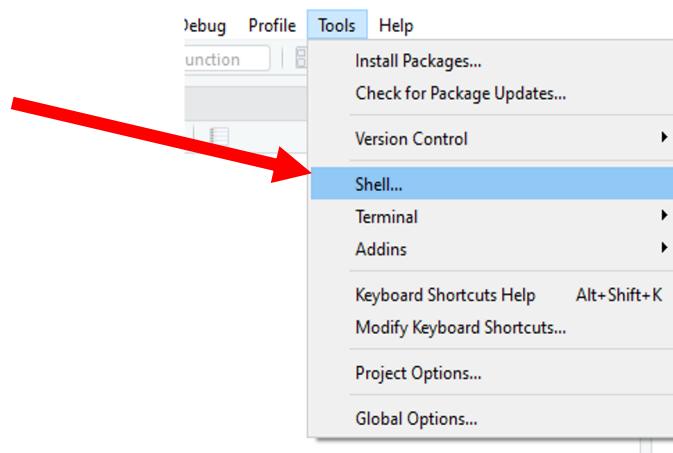
OR

2. Go to Terminal next to your Console



OR

3. Go to Tools -> Shell...



# Step 1: Add project to Git on your local computer (3/3)

Navigate to the project directory via the command line.

Initialize a local Git repository within your created project.

1. `git init`

Make sure the command line is running in your project directory before executing the commands!

Add files to your local repository (can add new files individually by name or all at once with ".").

1. `git add test.R` (OR) `git add .`

# Step 1: Add project to Git on your local computer (3/3)

Navigate to the project directory via the command line.

Initialize a local Git repository within your created project.

1. `git init`

Make sure the command line is running in your project directory before executing the commands!

Add files to your local repository (can add new files individually by name or all at once with ".").

1. `git add test.R` (OR) `git add .`

# Step 1: Add project to Git on your local computer (3/3)

Navigate to the project directory via the command line.

Initialize a local Git repository within your created project.

1. `git init`

Make sure the command line is running in your project directory before executing the commands!

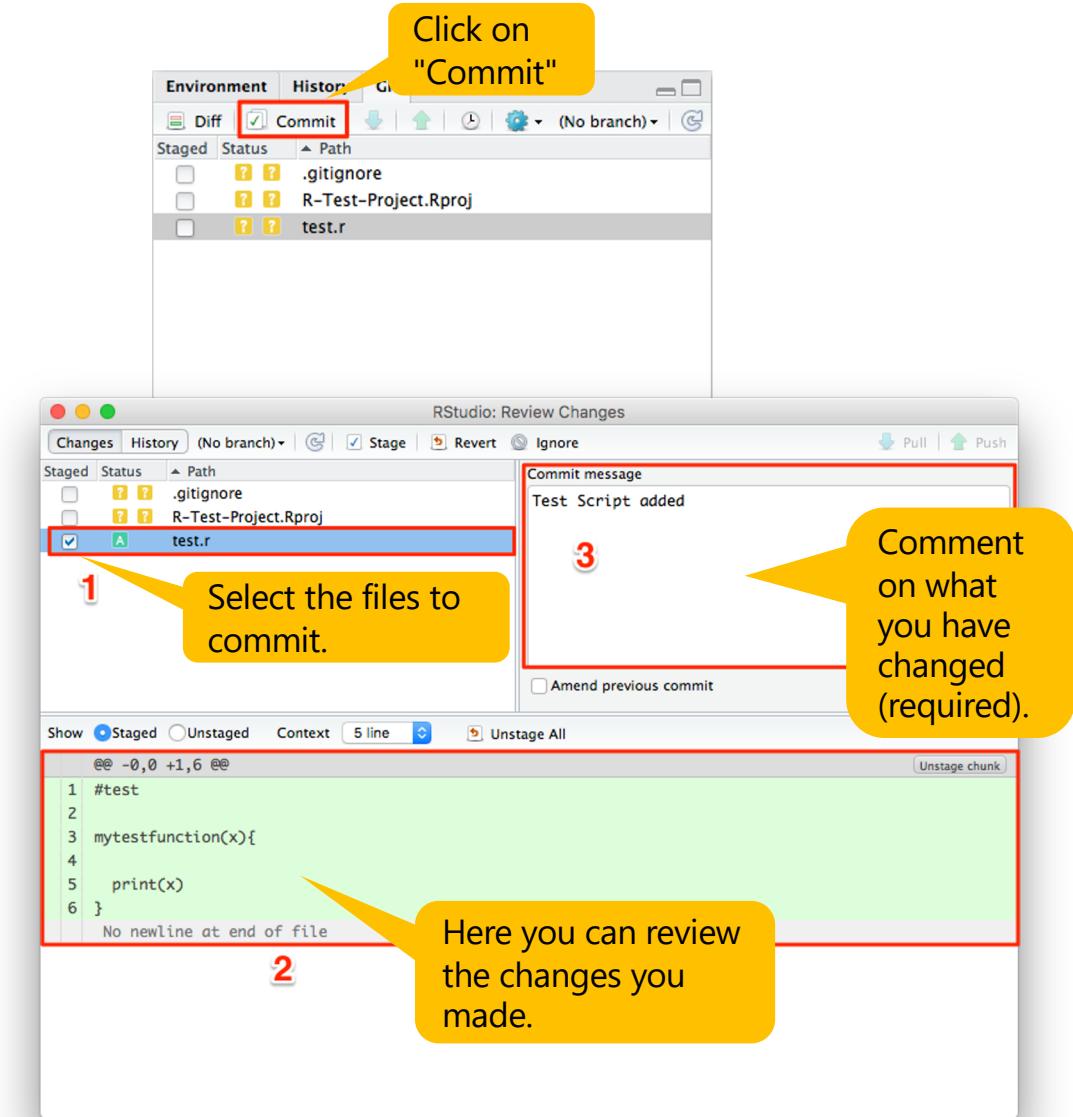
Add files to your local repository (can add new files individually by name or all at once with ".").

1. `git add test.R` (OR) `git add .`

## Step 2: Committing to your local Git repository

- When you want to save files that you have newly added/changed in your Git repository, you "commit" them
- Each commit needs a message describing what is in the commit, e.g. "Removed menu bar"
- You can commit either with the command below or using the RStudio GUI on the right

```
git commit -m "first commit"
```



## Step 3: Create a new repository on GitHub (1/3)

The screenshot shows the GitHub homepage. At the top, there is a search bar labeled "Search GitHub", navigation links for "Pull requests", "Issues", and "Gist", and a user profile icon. Below the header, there are two recent activity feeds. On the right side, a dropdown menu is open from the user profile icon, listing options: "New repository" (which is highlighted with a red box and has a yellow callout bubble pointing to it), "Import repository", "New gist", "New organization", and some other partially visible options. A yellow callout bubble also points to the "New repository" option with the text "Click on \"New Repository\"". At the bottom right of the page, there is a section titled "Repositories you contribute to" with one item listed: "mmeierer/CLVTools" with a star count of "0".

# Step 3: Create a new repository on GitHub (2/3)

## Create a new repository

A repository contains all the files for your project, including the revision history.

Owner  Repository name  

Make sure your names describe your project!

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

Description (optional)

 Public  
Anyone can see this repository. You choose who can commit.

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

Attention! Everything you upload is publically accessible!

Initialize this repository with a README  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore:  Add a license:  

**Create repository**

# Step 3: Create a new repository on GitHub (3/3)

The screenshot shows a GitHub repository page for 'bachmannpatrick / R-Test-Project'. The top navigation bar includes 'Unwatch' (1), 'Star' (0), and 'Fork' (0) buttons. Below the navigation, there are links for 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. A yellow callout box points to the URL field in the 'Quick setup' section, which contains the HTTPS URL: <https://github.com/bachmannpatrick/R-Test-Project.git>. The URL field is highlighted with a red box and has a red arrow pointing to it from the left.

**Quick setup — if you've done this kind of thing before**

[Set up in Desktop](#) or **HTTPS** <https://github.com/bachmannpatrick/R-Test-Project.git>

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

**...or create a new repository on the command line**

```
echo "# R-Test-Project" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/bachmannpatrick/R-Test-Project.git
git push -u origin master
```

**...or push an existing repository from the command line**

```
git remote add origin https://github.com/bachmannpatrick/R-Test-Project.git
git push -u origin master
```

**...or import code from another repository**

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

**Write this URL down.**

## Step 4: Setup credentials to connect GitHub to your local computer

To check your user email and user name use:  
`git config user.email`  
`git config user.name`

Type in:

```
git config --global user.email "youremail@uzh.ch"
```

```
git config --global user.name "your-github-username"
```

This step will help your colleges to easily identify your code when working in teams.

# Setup credentials to connect GitHub to your local computer

When connecting with GitHub, you can either use your password to authenticate or an RSA key.

## A. Password

- no additional setup required (easy) but have to type in every time (annoying)

## B. RSA key

- requires additional setup but only have to do once

## Step 4: Setup credentials to connect GitHub to your local computer

When connecting with GitHub, you can either use your password to authenticate or an RSA key.

### A. Password

- no additional setup required (easy) but have to type in every time (annoying)

### B. RSA key

- requires additional setup but only have to do once

## Step 4: Setup credentials to connect GitHub to your local computer

When connecting with GitHub, you can either use your password to authenticate or an RSA key.

### A. Password

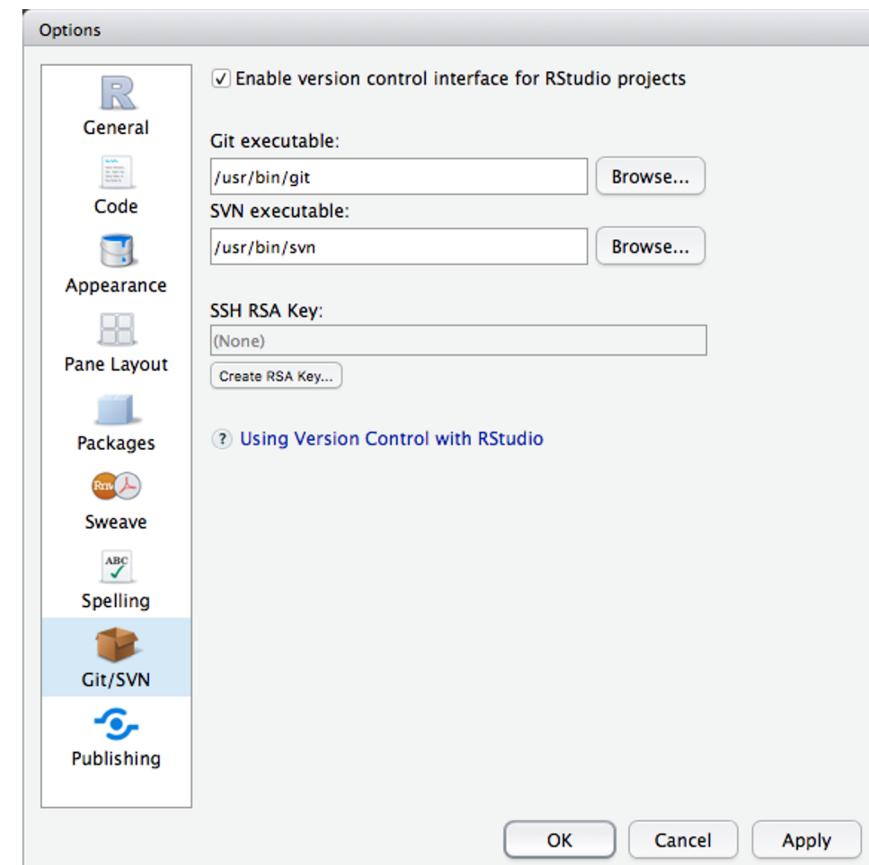
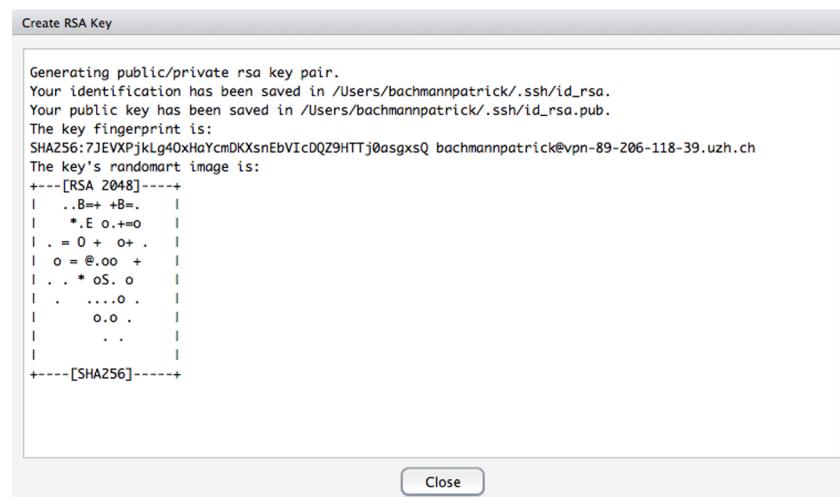
- no additional setup required (easy) but have to type in every time (annoying)

### B. RSA key

- requires additional setup but only have to do once

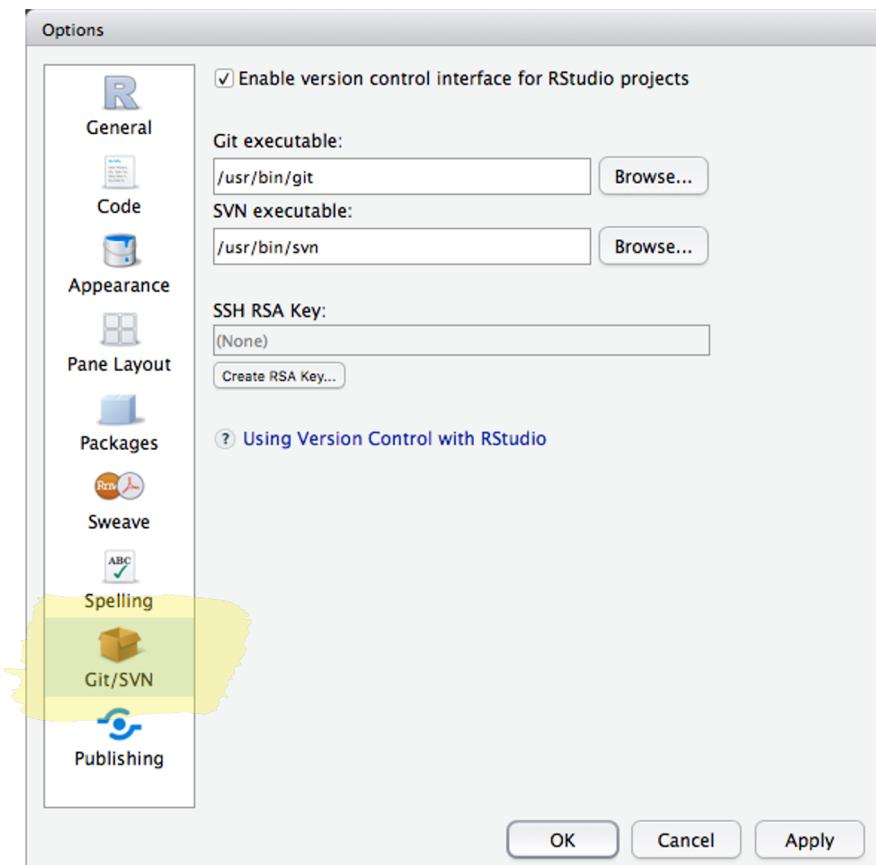
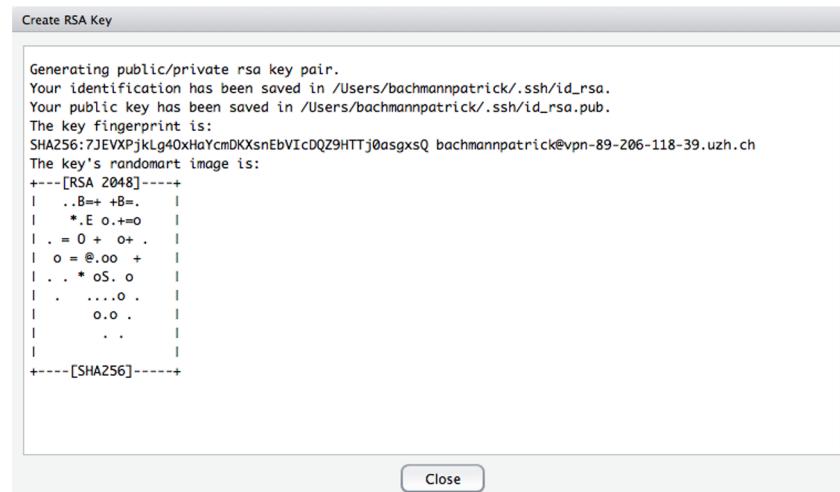
## Step 4 (B): For your convenience: Create an RSA key

1. In RStudio go to Tools -> Global Options ...
2. Click on Git/SVN.
3. Generate an RSA Key.
4. Do not use a passphrase for the key.



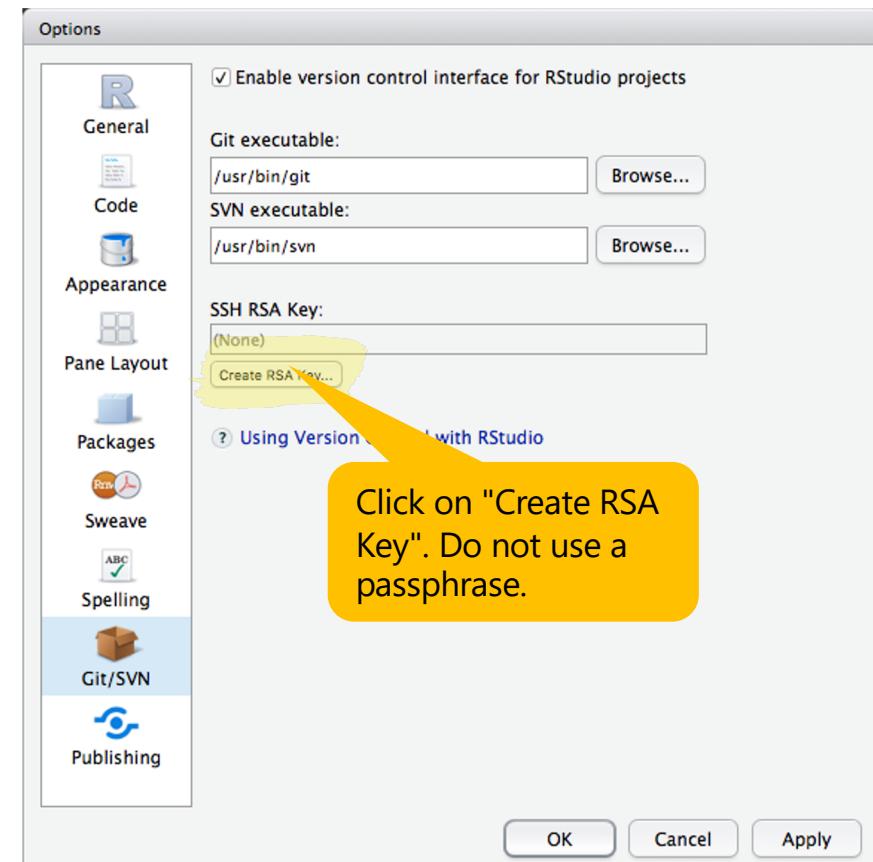
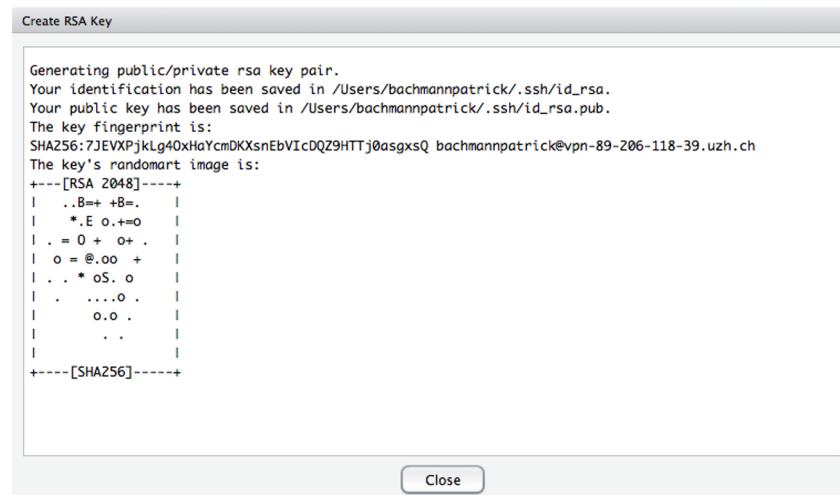
## Step 4 (B): For your convenience: Create an RSA key

1. In RStudio go to Tools -> Global Options ...
2. Click on Git/SVN.
3. Generate an RSA Key.
4. Do not use a passphrase for the key.



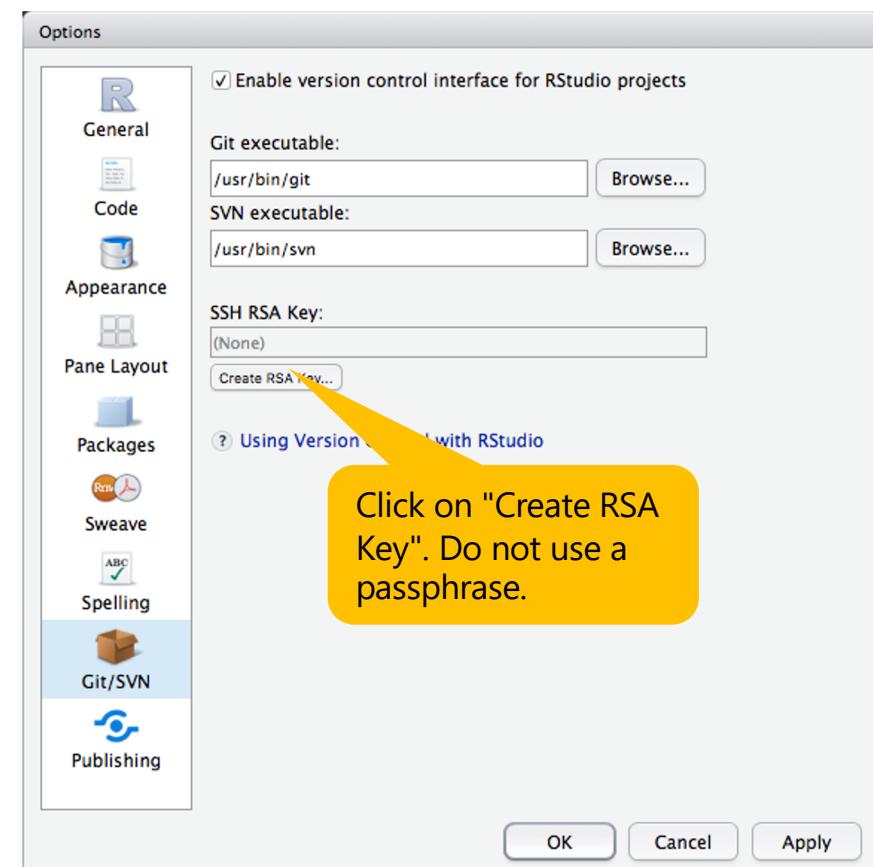
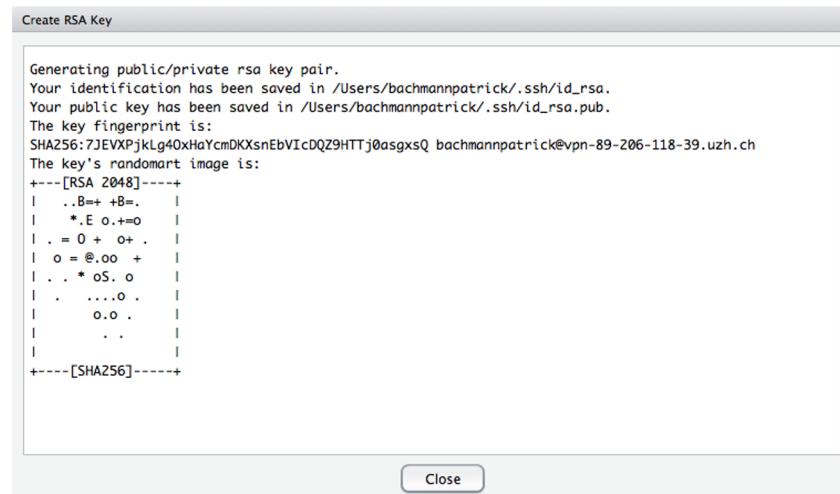
## Step 4 (B): For your convenience: Create a RSA key

1. In RStudio go to Tools -> Global Options ...
2. Click on Git/SVN.
3. Generate an RSA Key.
4. Do not use a passphrase for the key.

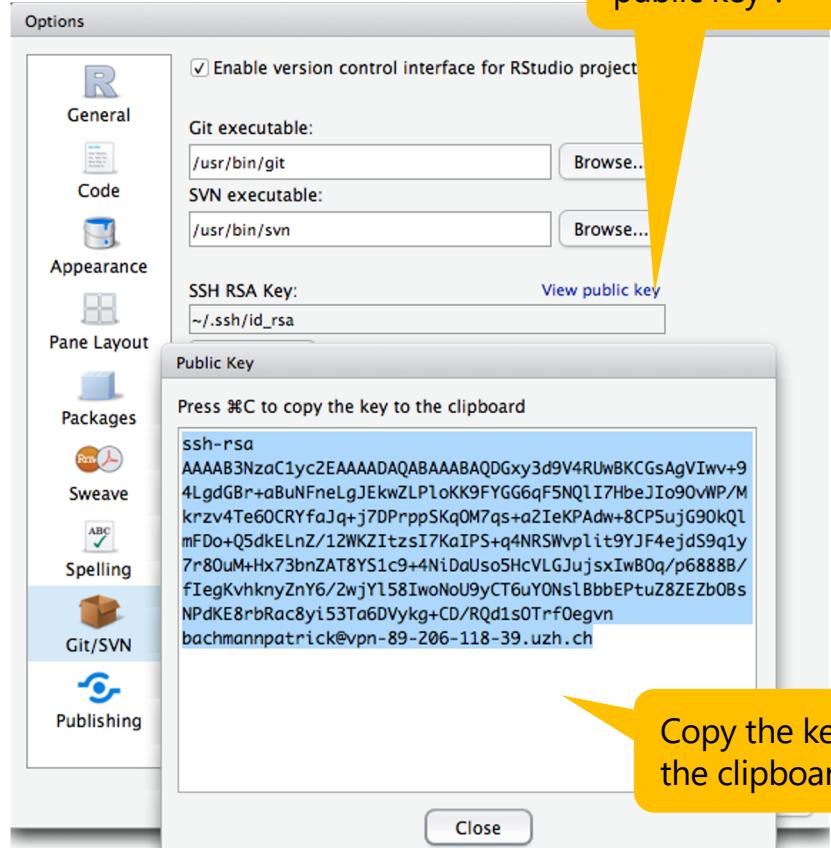


# For your convenience: Create a RSA key

1. In RStudio go to Tools -> Global Options ...
2. Click on Git/SVN.
3. Generate an RSA Key.
4. Do not use a passphrase for the key.



## Step 4 (B): For your convenience: Create an RSA key



Go to  
Github.com

The screenshot shows the GitHub "SSH keys" settings page. On the left, a sidebar lists "Personal settings" (Profile, Account, Emails, Notifications, Billing) and "SSH and GPG keys" (selected). A red arrow labeled "1" points to the "SSH and GPG keys" link. In the main area, a list of existing SSH keys is shown with a red arrow labeled "2" pointing to the "New SSH key" button. The "Key" input field at the bottom contains the copied RSA key, indicated by a red box and a red arrow labeled "3". A red arrow labeled "4" points to the "Add SSH key" button at the bottom right.

# Step 5: Push local Git repository to GitHub

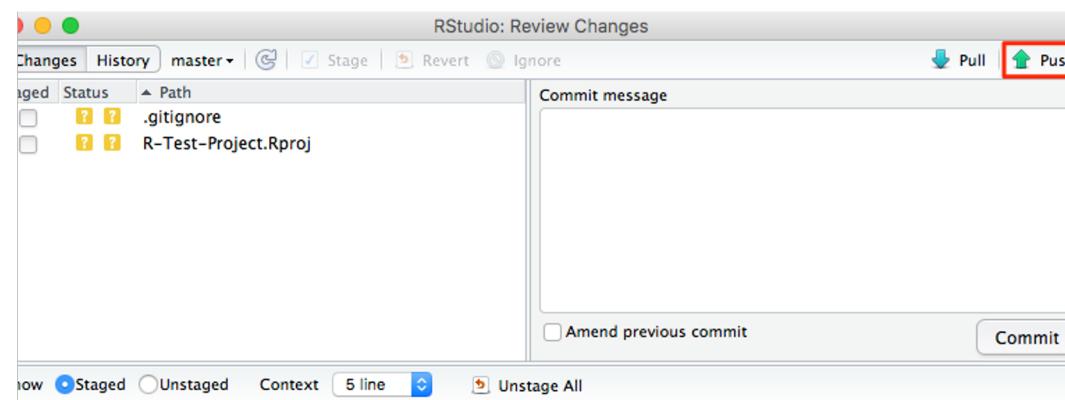
First you must add the GitHub repository you created earlier as a remote repository (This has only to be done once!):

```
git remote add origin https://github.com/bachmannpatrick/R-Test-Project.git
```

Enter your GitHub repository URL

Then you can push all the commits you have made to the remote GitHub repository:

```
git push -u origin master
```



Alternatively, press "Push" to publish your commits to the remote repository.

# Step 5: Push local Git repository to GitHub

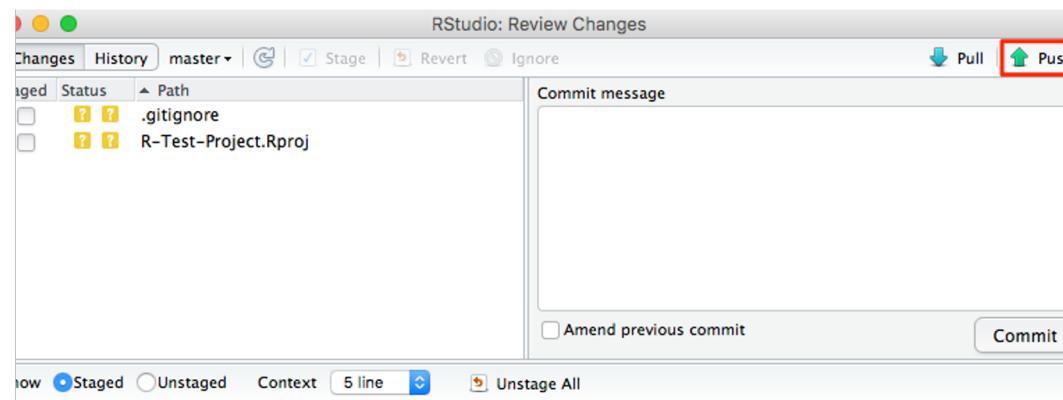
First you must add the GitHub repository you created earlier as a remote repository (This has only to be done once!):

Enter your GitHub repository URL

```
git remote add origin https://github.com/bachmannpatrick/R-Test-Project.git
```

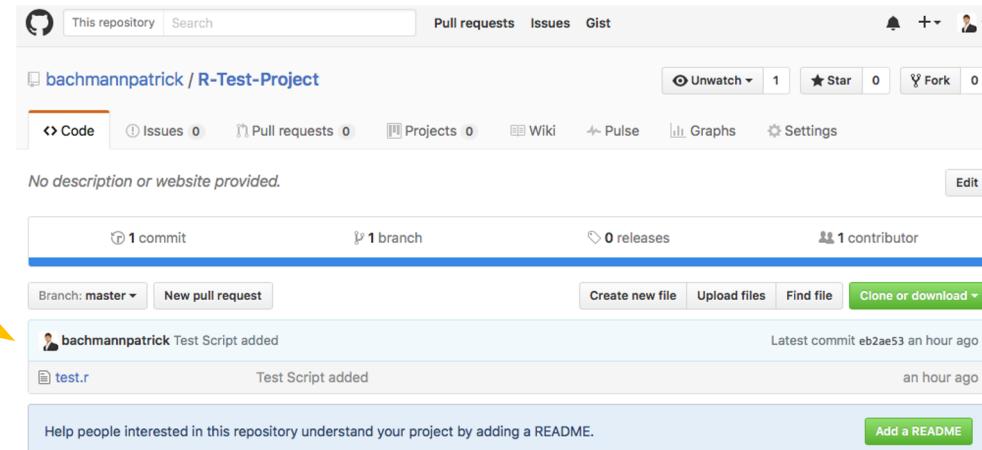
Then you can push all the commits you have made to the remote GitHub repository:

```
git push -u origin master
```



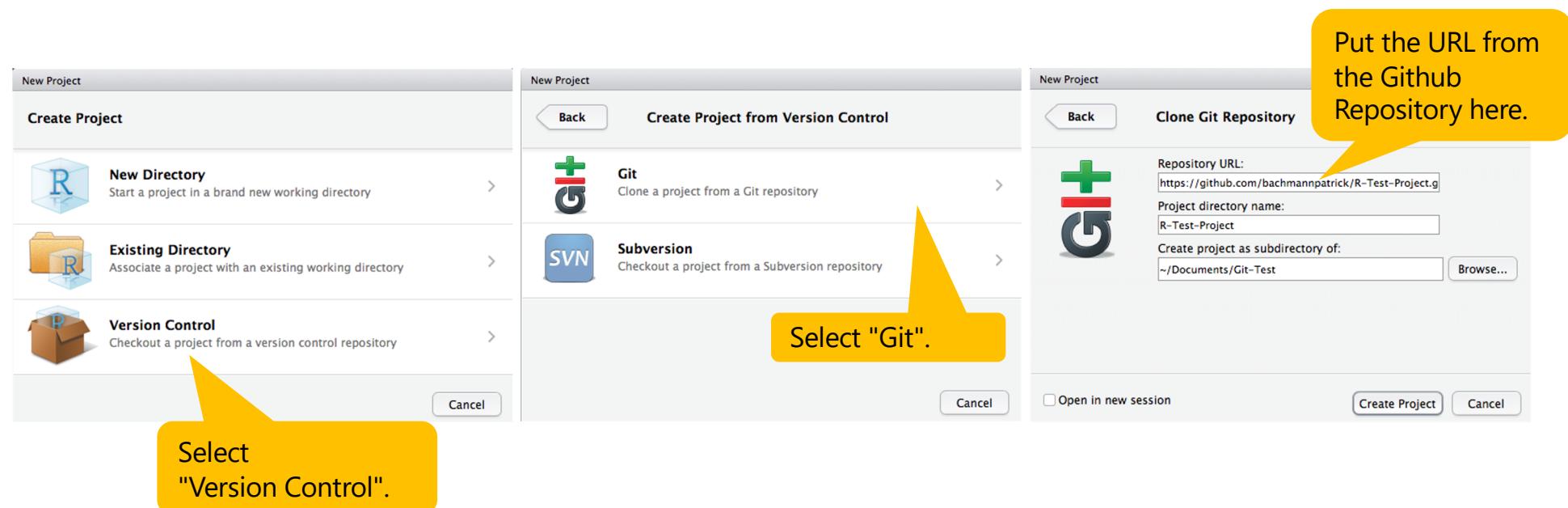
Alternatively, press "Push" to publish your commits to the remote repository.

# Check the changes you made on github.com



# Alternative for linking the repository

Create a New Project in RStudio ("File" -> "New Project"):



**Now it's your turn!**