Step-by-step of how to create your personal website using R blogdown, Hugo, and GitHub.

By: Monsuru Adepeju (www.madepeju.com)

Updated on: 12/02/2019.

1. Sign up on GitHub:

(i) Sign up on **'GitHub.com'** (if you don't already have one): https://github.com/join?source=header-home

2. Installation

(i) Install **Desktop GitHub** *software (if you don't already have it).* After the installation, you can set it up (i.e. log into it) with your GitHub account details (above). Check your '*My Document*' folder on your local PC, you should see that a folder named '*GitHub*' has been created.

3. Create a GitHub Repository

(i) Sign into your GitHub account (in your web browser).

(ii) At the top right-hand corner of the page, click the + sign and select 'New *repository*'.

(iii) For illustration, I created a repo called '*myWebsite*' ('repo' means 'repository). This repo will hold all the contents of our website. You can type a short description if you want. Check the 'Public' option (for now), and also check '*Initialise this repository with a README*' option.
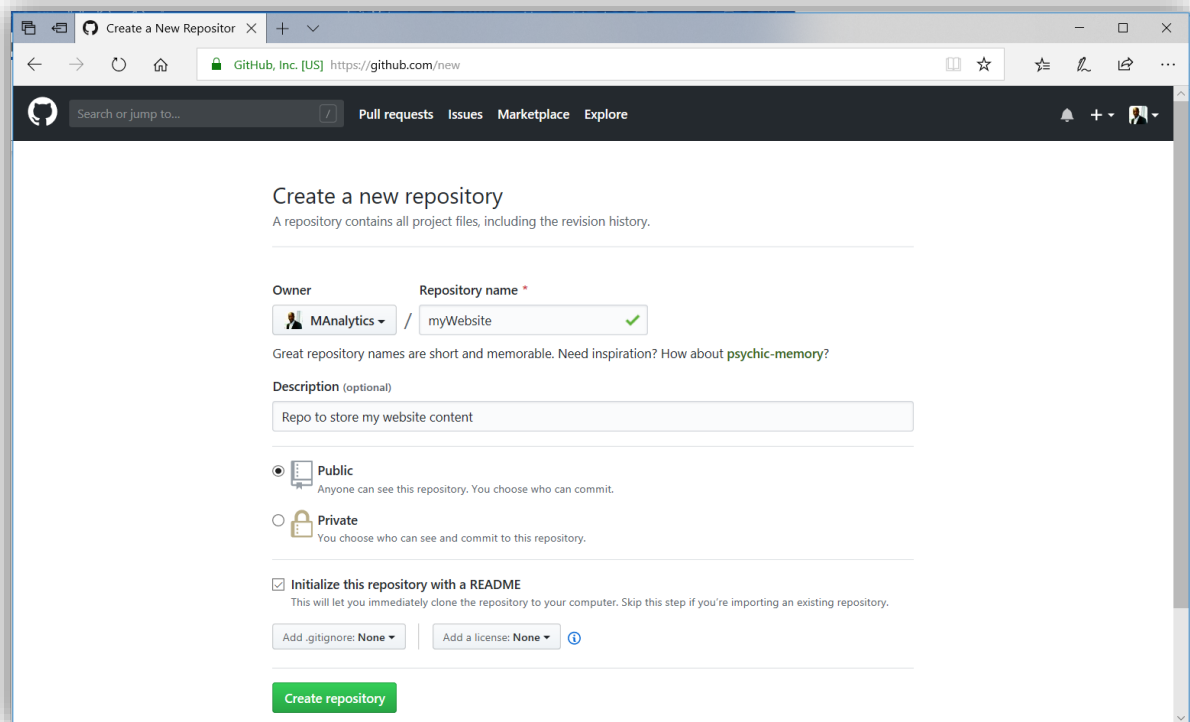
4. Cloning your repo onto your local machine

(i) Make sure you are signed in to your GitHub account (web browser).

(ii) Click the profile pic at the right-hand corner of the GitHub page, and select '*Your Repositories*', to show all repositories in your account. Find and click on the new repository that you created (i.e. '*myWebsite*').

(iii) Click on '*Clone or download*' button.

(iv) Click on '*Open in Desktop*' (This means that you are instructing GitHub to clone this repo onto your local PC and at the same time open it using the '***Desktop GitHub'*** software that you installed.
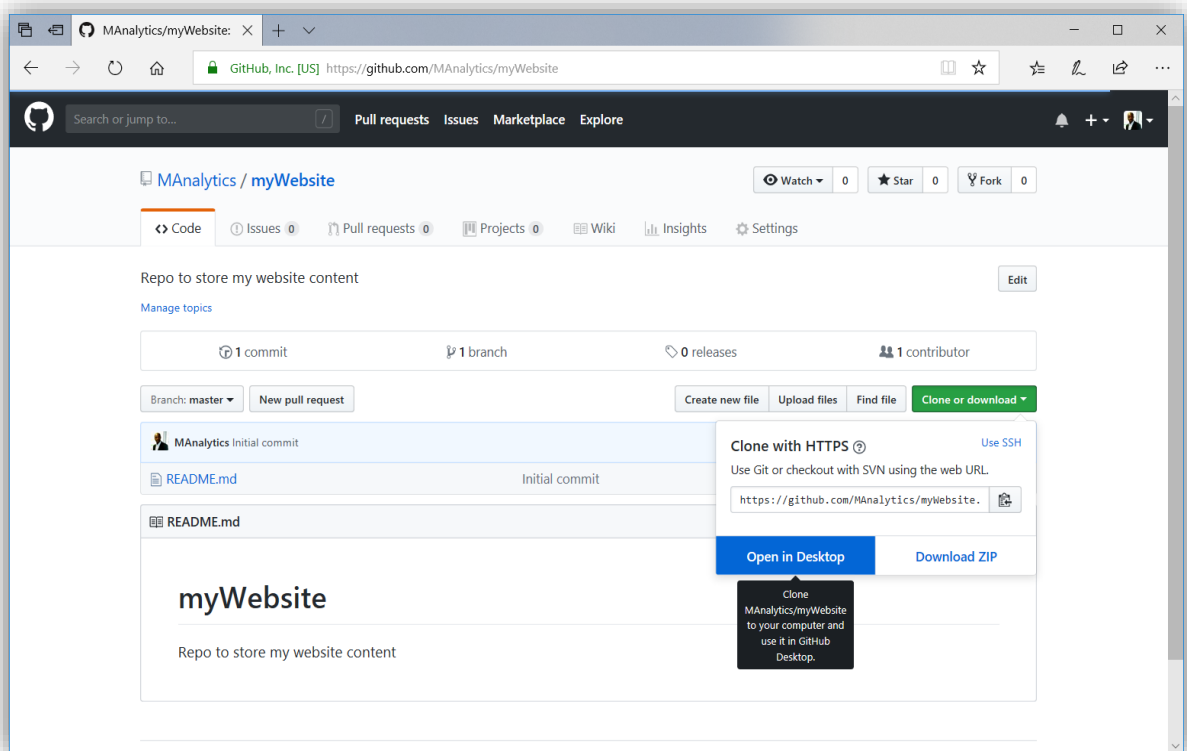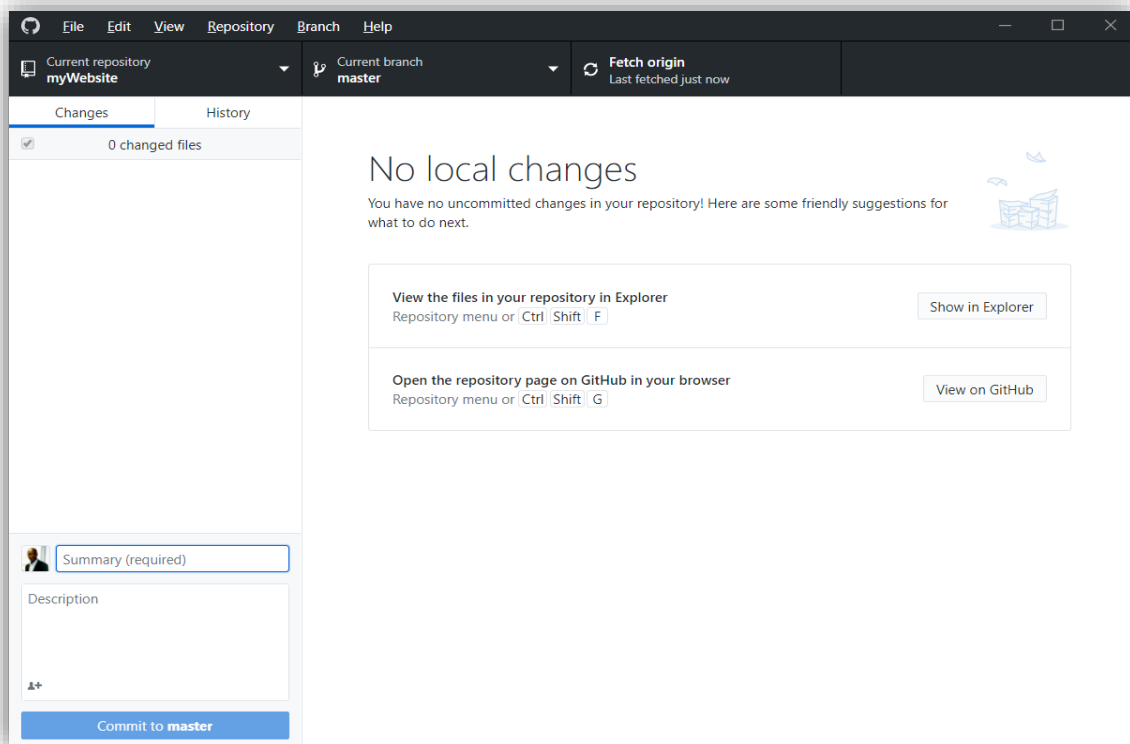

(v) Your ***Desktop GitHub*** software will be loaded automatically, and you should see a similar window as below. Check that the '*Local path*' box ends with this: "*….\GitHub\myWebsite*"
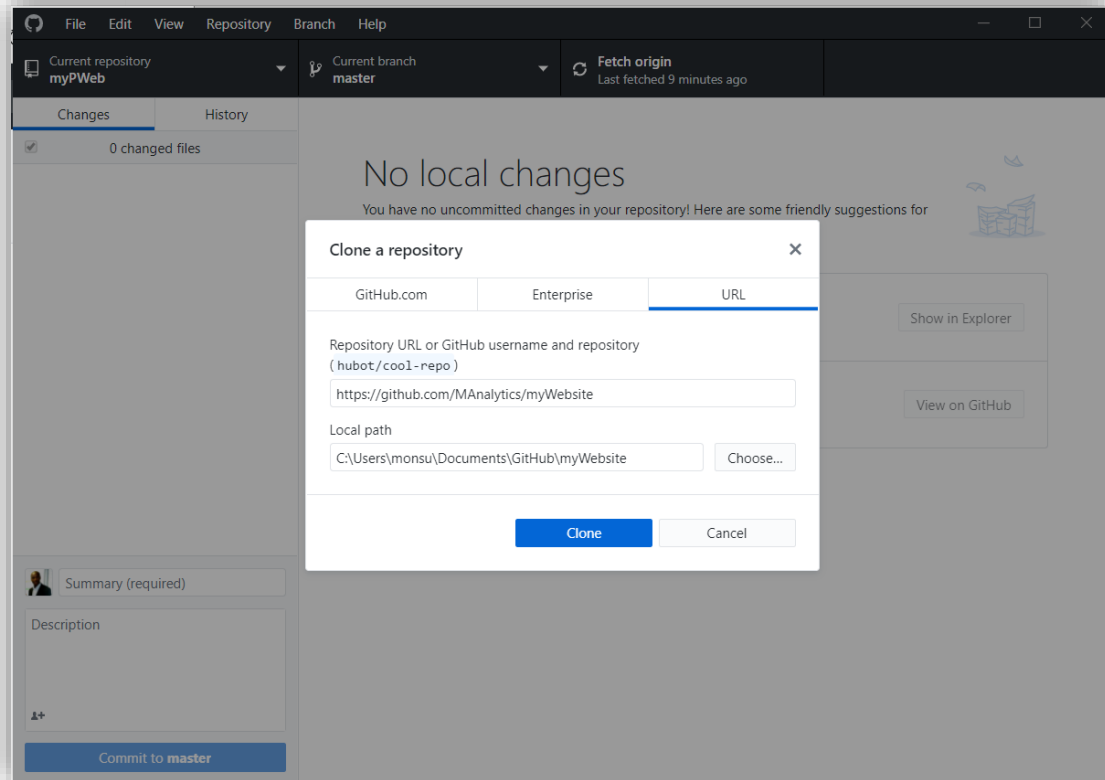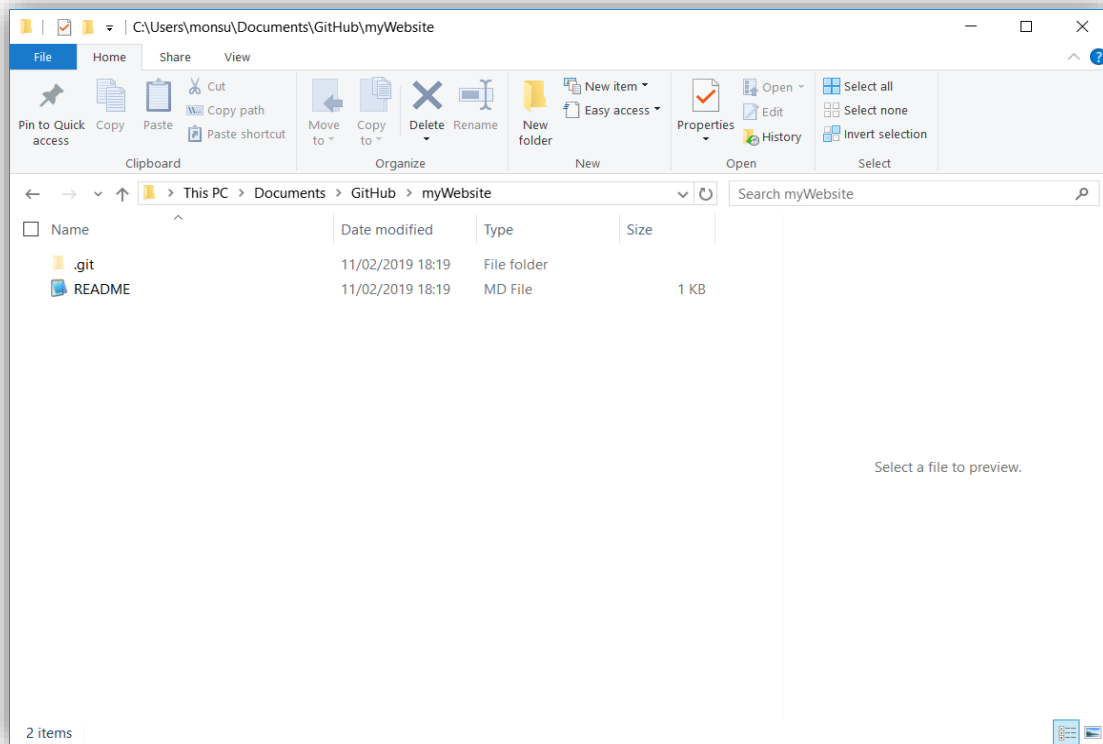
(vi) Click on '*Clone*' button.

(vii) Now that you have successfully cloned the repo onto your PC, navigate to *"….\My Document\GitHub"* and delete the *README.md* file. Don't add anything else to this folder just yet, *blogdown* requires that the directory is empty (However, do not delete the *.git* folder that appears).

(viii) Check the '**Desktop GitHub**' program, you will see that the software is tracking the changes that you have made to the folder. That is, it detects the deletion you made from '*myWebsite*' folder. This means that any time you make any changes, whatsoever, inside this folder, the software will detect it.
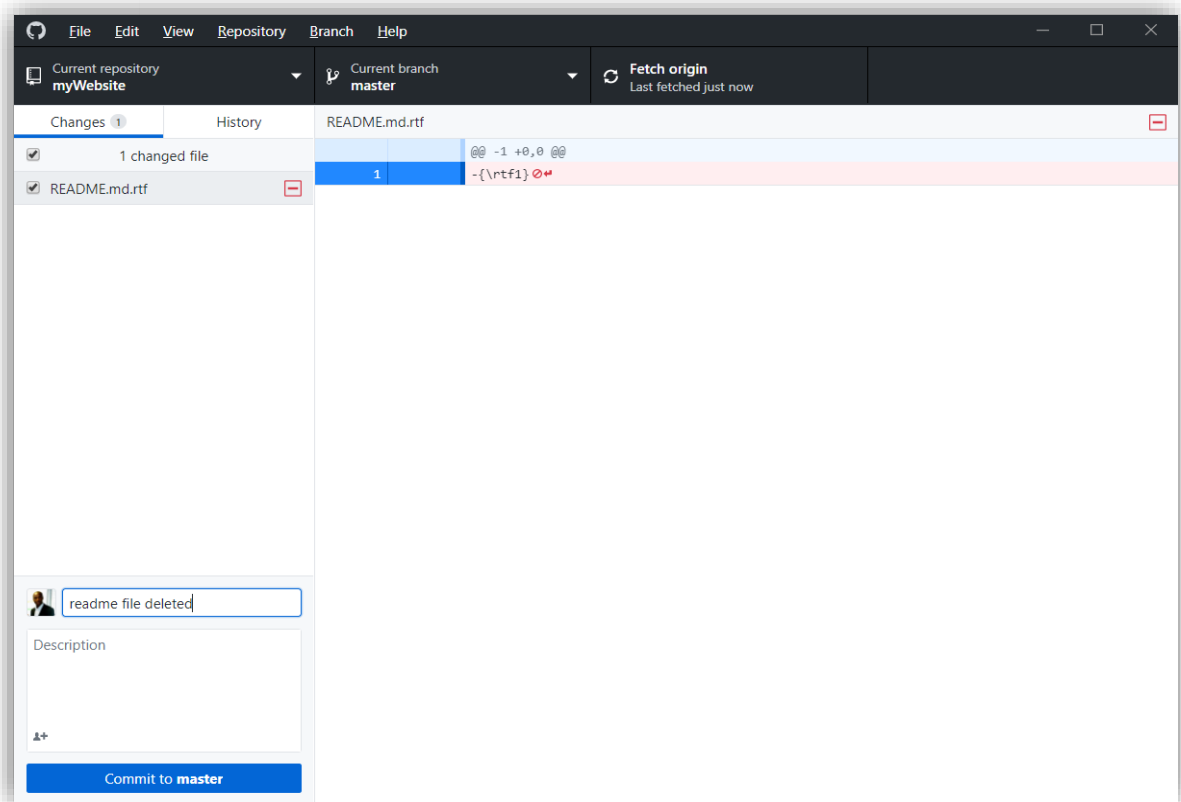
(ix) Next, you now need to update the corresponding '*myWebsite'* repo on your GitHub account. This can be done as follows:

- o Type a short description of the changes in the small description box beside your profile pic e.g. 'readme file deleted'.
- o Click "*Commit to master*" button, and then click "*Push origin*" button. (Congratulations! You have just updated your GitHub '*myWebsite'* repo from your local machine)

FYI: This is one of the effective ways of updating the contents of a GitHub account.

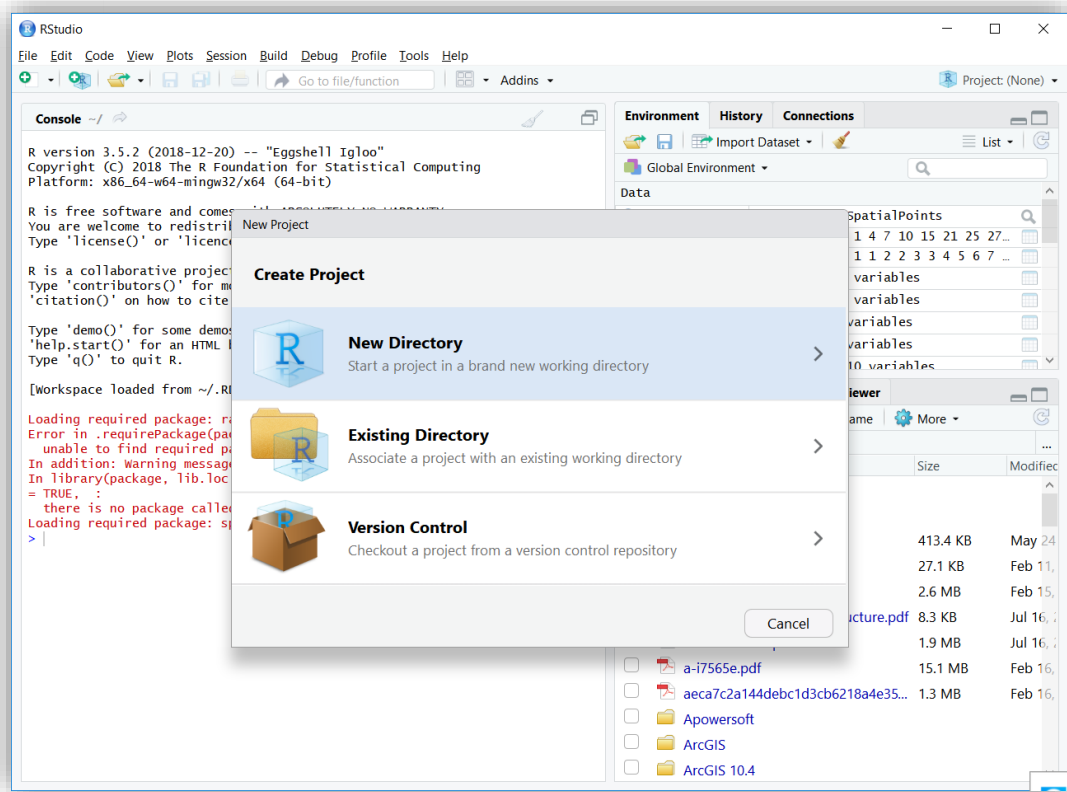## 5. Downloading the website template using RStudio

(i) Load **RStudio**.
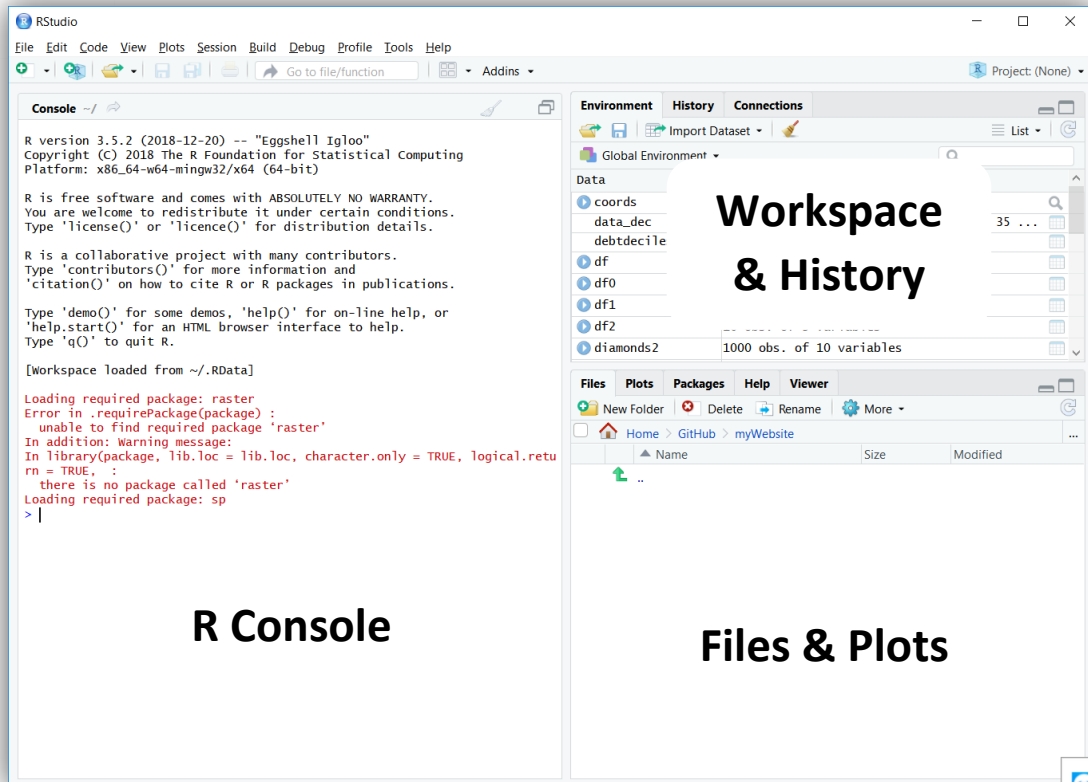
(ii) Download the '*blogdown*' package by typing the code:
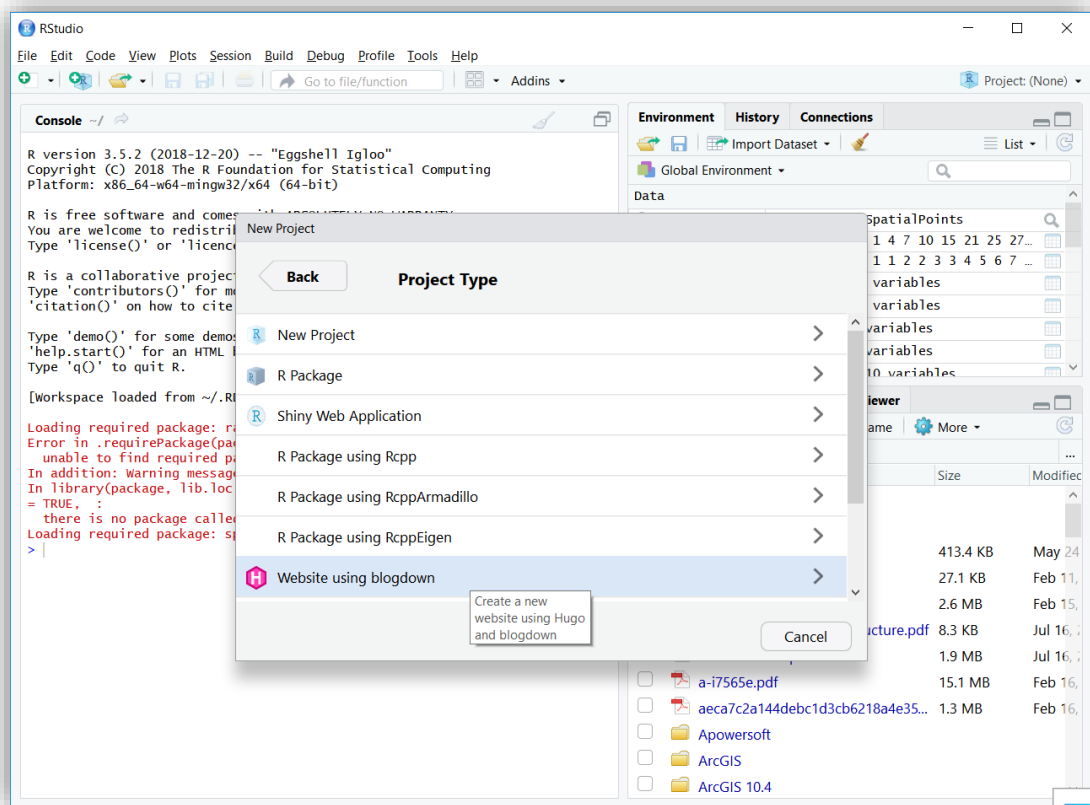
> *install.packages("blogdown")*

(iii) When successfully installed, re-load **RStudio**. Under the '*Files & Plots*' section, select '*Files*' tab and navigate to '*myWesite*' folder (it should be empty).

(iv) Click on '*File*' menu, select '*New Projects…*", and select '*New Directory*'.

(v) Select '*Website using blogdown*'

(vi) In the '*Directory name:*' box, enter a name. Let's say: '*all_web_content*'. Ensure that you have the path suffix "…/GitHub/*myWebsite*" under the '*Create project as subdirectory of*:'

(vii) In the '*Hugo theme:*' box, enter the "*GitHub_Username/templateRepo*" of the Hugo template that intend to use. For example, to use the hugo-academic template, you have "*gcushen/hugo-academic*"**. *I recommend that you experiment with this template first*. The collection of the hugo templates can be found here: *https://themes.gohugo.io/*. You can *also try: "manalytics/mywebTemplate"* in order to use the template of my website: www.madepeju.com which was also based on the original hugo-academic template.

Then, click 'Create Project'.
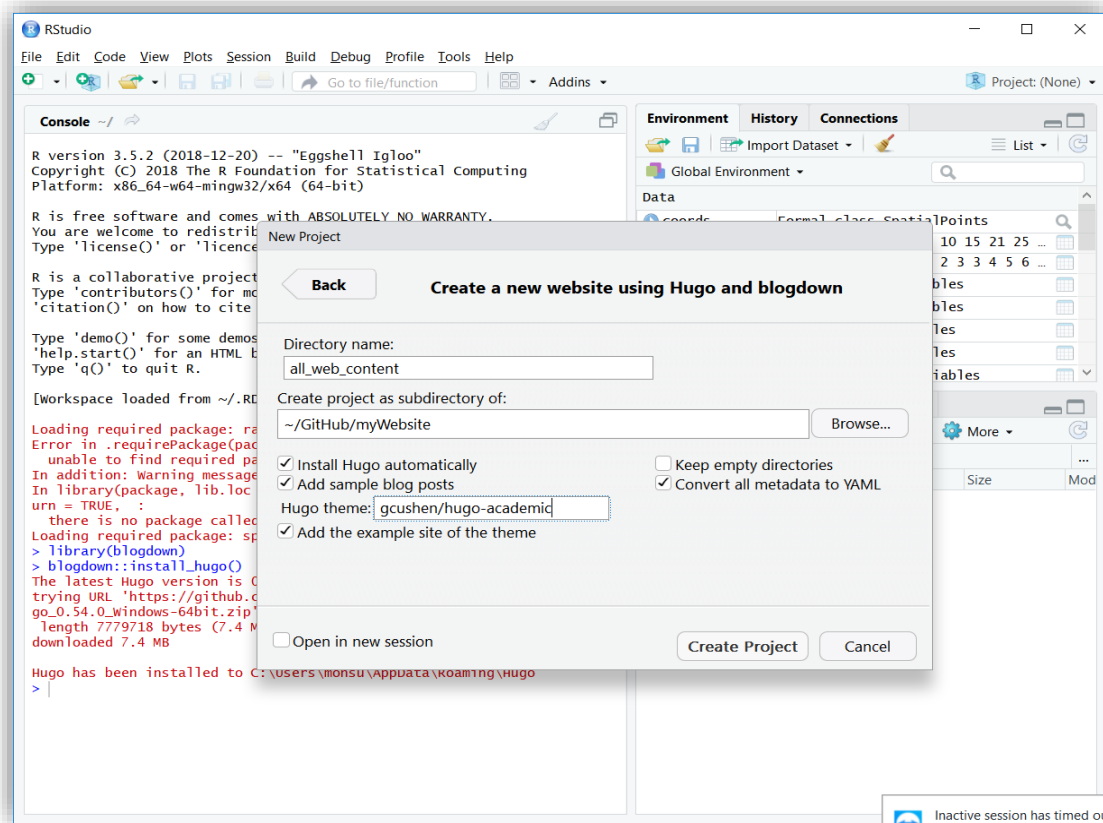
(viii) You might get some error messages, just ignore them. From the '*Files & Plots*' section, navigate to confirm that an '*all_web_content*' folder has been created as a subdirectory of *myWebsite* folder. The *all_web_content* folder will contain all the template files you need to build a website.

(ix) Set the *all_web_content* folder as your working directory using the *setwd()* command.

> setwd("C:/Documents/GitHub/myWebsite/*all_web_content*") #note the use of forward slash "/" instead of "\".

(x) Loading the libraries and installing hugo.

> *library(blogdown)*

> *blogdown::install_hugo() # installing hugo – a static site developer.*

> *blogdown::update_hugo() # make sure that you run this regularly too in order to keep hugo updated.*

> *blogdown::hugo_version()* *#It is important that you take note of the current version of the hugo using this same command and monitor whether it has changed. Why do you need to be doing this? Because you will need to enter the version of hugo in the Environmental Variables settings when deploying your website. I recommend using Netlify (www.netlify.com) to deploy your website because of its great compatibility with GitHub*

(xi) Next, select '*Addins*' and click '*Serve Site*' to build the website.
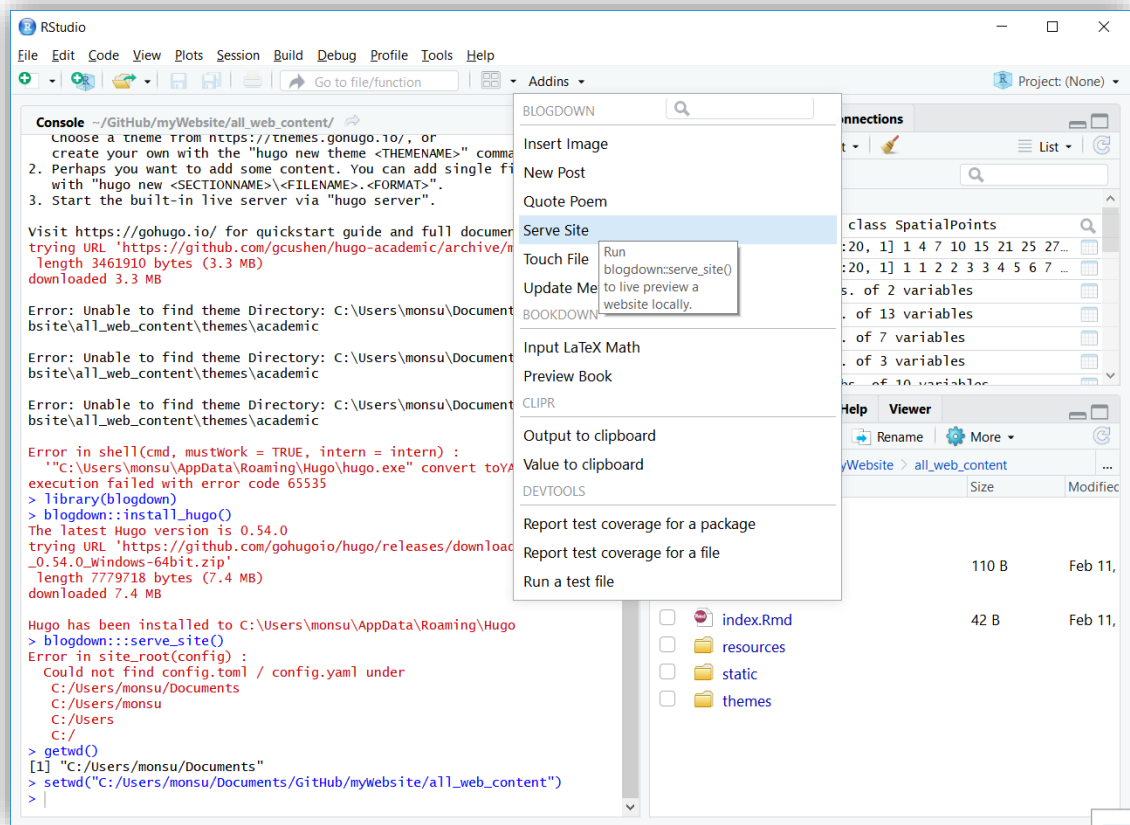
(xii) If the website is built successfully, you will see a preview in the '*Files & Plots*' section of the **RStudio**. You can also preview it in a web browser by clicking the '*Show in New Window*' icon of the '*Files & Plots*' section.

(xiii) You should have the same preview as below:
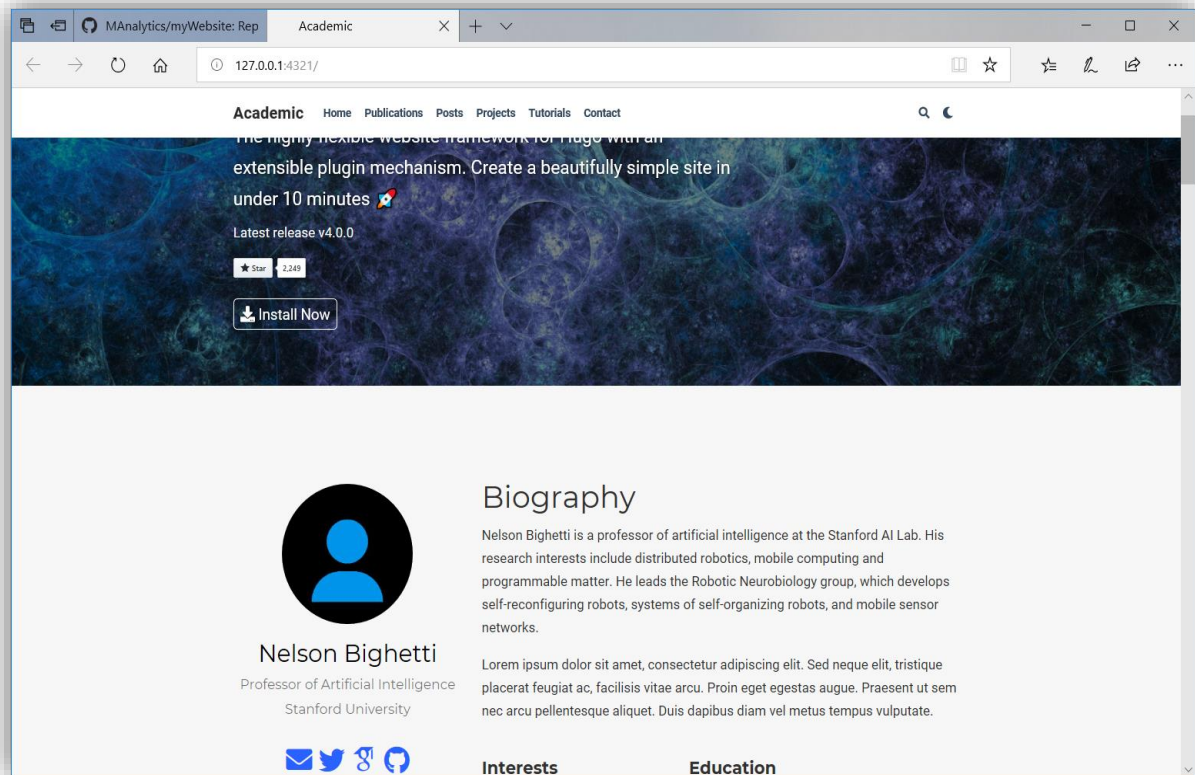
6. Editing the website

(i) The first step towards editing this website (i.e. the content of the folder: *"…\My Document\GitHub\myWebsite"*) is to first locate the '*config.toml*' file and open it in **RStudio**. The '*config.toml*' file contains basic settings of your website such as the *theme* and most of the *home page* settings. All files with the extensions **.md* and **.Rmd* can be edited using RStudio while other files such as images can be added or deleted directly through the window explorer.
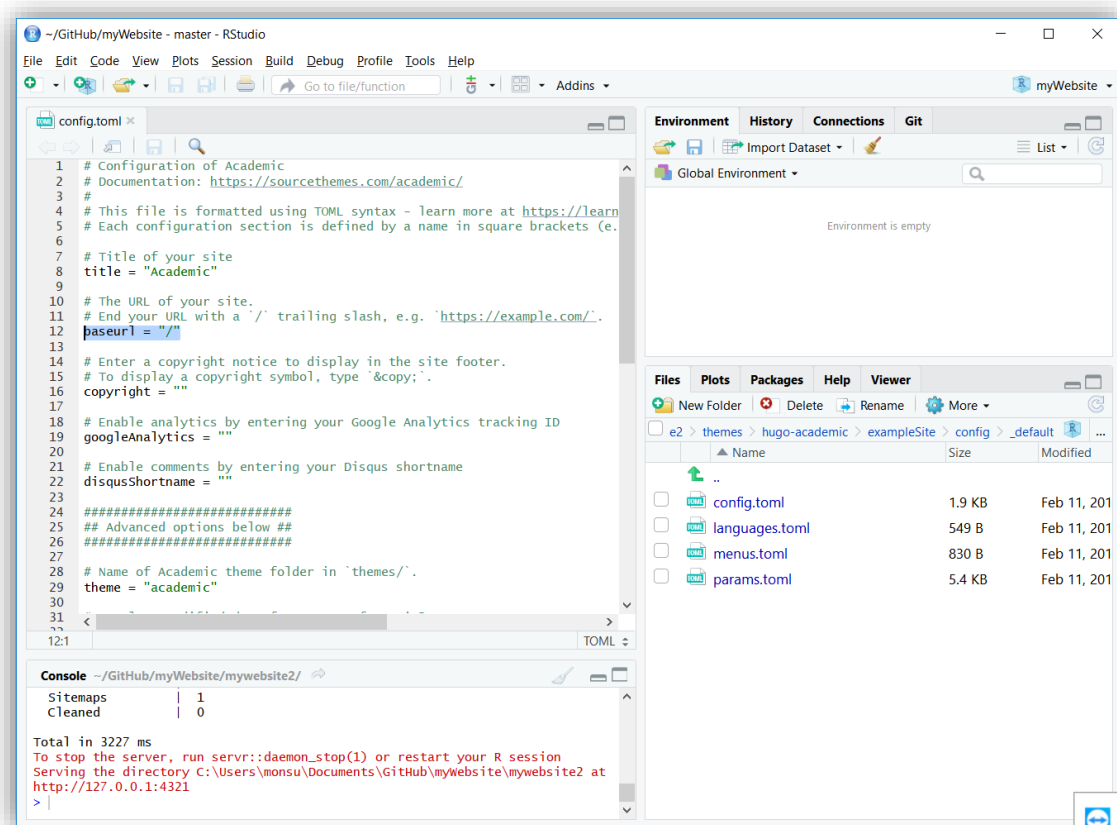

But remember that every time you make any changes, you have to carry out step 5(xi) to re-build the website. After the website has been built successfully, you then go to the 'Desktop GitHub' software to push the changes onto your GitHub account (i.e. update the "myWebsite" repo) using step 4(ix). Note that it is "myWebsite" repo that you are going to be deploying repeatedly as your actual website.

7. Deploying your website

At this point, you can set up a Netlify account (it's free!). Through Netlify you will be able to connect your "*myWebsite*" repo and deploy it as a live website. This part is very easy. The *step-by-step* of how to do this can be found here: https://www.netlify.com/blog/2016/09/29/a-step-by-step-guide-deploying-on-netlify/. At first, Netlify will use the 'public' folder to serve your website (i.e. you will have something like *something-something.netlify.com*). Note: you will also have to specify your Hugo version in the *Environmental Variables* section (see step 5(ix)). Also note that a part of connecting your "*myWebsite*" repo include editing the 'baseurl = "/"' line of the *config.toml* file to baseurl = "*something-something.netlify.com*" in order for your website to be able to go live.

When the website is finally up and running, you can create a customised domain name for a token of $9.99 per year. That is, you will have the opportunity to remove the '.netlify' from your '*something-something.netlify.com*'.
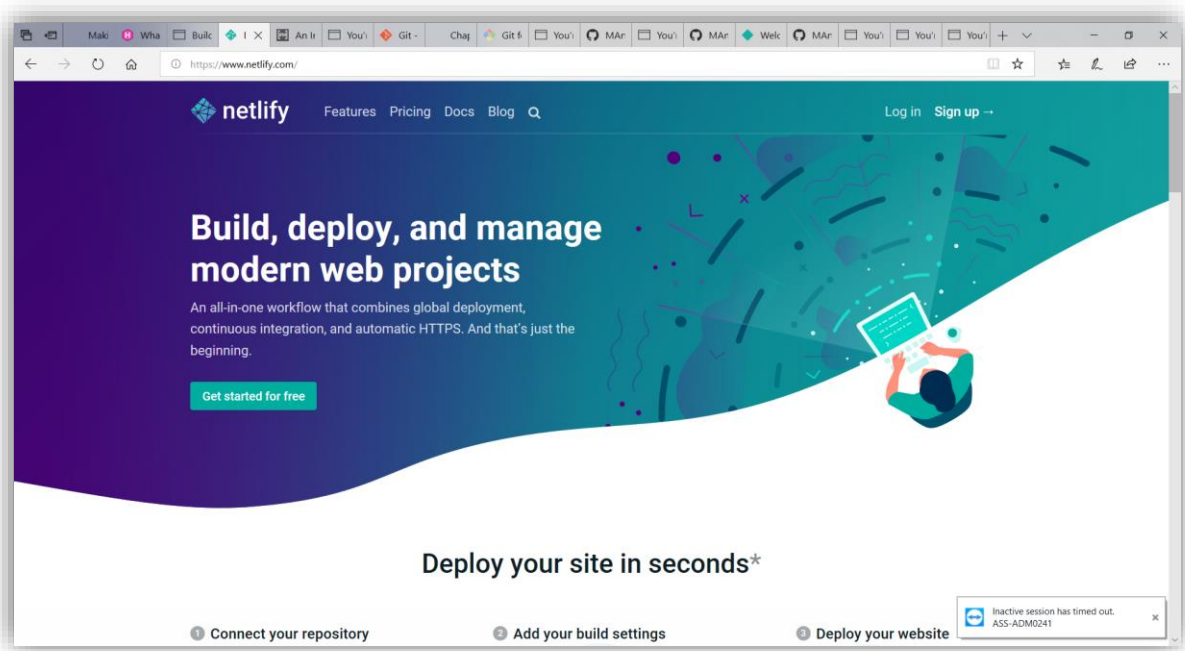
When the website is finally up and running, you can create a customised domain name for a token of $9.99 per year. That is, you will have the opportunity to remove the '.netlify' from your '*something-something.netlify.com*'.

What next?

Congratulations!

The next thing you will be doing now is to keep repeating the following 4 steps:

(i) Edit, remove or add a files in "C:....\My Document\GitHub*myWebsite*" on your local PC using **RStudio** or directly from the **window explorer**

(ii) Re-build the website in **RStudio** by clicking the 'Addins > Serve Site'

(iii) Push the changes to your "*myWebsite*" repo using the **Desktop GitHub**.

(iv) Deploy on *Netlify.com*.