- DANL 310: Data Visualization and Presentation -

4. Getting Started with a GitHub website and RMarkdown

Byeong-Hak Choe

SUNY Geneseo

February 8, 2022

Personal website on GitHub

- We have a class website on GitHub (https://bcdanl.github.io/index.html).
- This website is built by `git` and `R Markdown` using `R Studio` and `terminal` (`Git Bash` for Windows users).

Personal website on GitHub

Step 0

- Step 0.
  1. Install `git`.
     - For Windows, go to https://gitforwindows.org. Make sure you have “Git Bash” on your laptop.
     - For mac, go to http://git-scm.com/downloads.

     In `Terminal` or `Git Bash`, ...

     1.1 `git config -list`
     to check a list of git users in your computer.

     1.2 `git config -global user.email "YOUR_GITHUB_EMAIL_ADDRESS"`

     1.3 `git config -global user.name "YOUR_GITHUB_USERNAME"`

Personal website on GitHub

Step 0

- Step 0.
  2. Obtain a personal access token (PAT) for `github`. In RStudio Console, ...
     - 2.1 `install.packages("usethis")`.
     - 2.2 `usethis::create_github_token()`.

     Then, click “Generate token” in the pop-upped web browser. Then, copy the generated PAT to your clipboard.
  3. Set the GitHub credential using the PAT. In RStudio, ...
     - 3.1 `install.packages("gitcreds")`.
     - 3.2 `gitcreds::gitcreds_set()`.

     Then, paste your PAT to the RStudio Console.

Personal website on GitHub

Step 1

- Step 1.
  1. Create your GitHib account and login to it.
  2. From https://github.com, click the plus (+) icon in the upper right corner and select “New repository”.
  3. Name this repo `YOUR_GITHUB_NAME.github.io`, which will be the domain for your website.
  4. Copy the address of your git repo

     `https://github.com/YOUR_GITHUB_NAME/YOUR_GITHUB_NAME.github.io`

Personal website on GitHub

Step 1

- Step 1.
    5. Find the path name of the folder where you want to keep the files for your website on your laptop.
    6. If you have the folder, `YOUR_USERNAME.github.io`, change its name to anything (e.g., `tmp`).
    7. For mac users, open `terminal`. For Windows users, open `Git Bash`.
    8. In `Terminal` or `Git Bash`, ...

- `cd PATH_NAME_OF_THE_PARENT_FOLDER_FOR_THE_WEBSITE_FOLDER` for terminal.
- `cd "PATH_NAME_OF_THE_PARENT_FOLDER_FOR_THE_WEBSITE_FOLDER"` for Git Bash.

Personal website on GitHub

Step 1

- Step 1.
  9. Follow the steps in https://pages.github.com.
     - 9.1 If you have an error for the line `git push -u origin main`, try `git push -u origin master`.
  10. Close your `terminal` or `Git Bash`.
  11. Go to https://YOUR_USERNAME.github.io.
  12. Change the folder name, `YOUR_USERNAME.github.io` to anything (e.g., `tmp2`).

Personal website on GitHub

Step 2

- Step 2.
  1. Open R Studio.
  2. Select “New Project” under the “File” menu.
  3. Click “Version Control”.
  4. Click “Git”.
  5. Paste the address of your git repo in the “Repository URL” dialogue box:
     `https://github.com/YOUR_GITHUB_NAME/YOUR_GITHUB_NAME.github.io`
  6. Choose the “Project directory name” as `YOUR_GITHUB_NAME.github.io`.
  7. Select a location of the files for your personal website in your laptop and copy its path name to “Create project as subdirectory of:”.
  8. Click “Create Project”.

Personal website on GitHub

Step 3

- Step 3.
  1. Let's update our rmarkdown package to make sure we actually have the version that supports R Markdown websites.
     - `install.packages("rmarkdown", type = "source")`.
  2. From our Canvas, download the website files—(1) `index.Rmd`, (2) `about.Rmd`, and (3) `_site.yml`—and save them inside your repository folder, `YOUR_GITHUB_NAME.github.io`, in your laptop.

Personal website on GitHub

Step 4

- Step 4.
  1. Run the following within your R Studio project.

```
rmarkdown::render_site()
```

     - Now if everything has gone according to plan, you should get a bunch of output followed by the message : `Output created: index.html`.

  2. Open the repository folder with Finder (Mac) or File Explorer (Windows) in your laptop.
  3. Open the file, `index.html` to see if the website is rendered.

Personal website on GitHub

Step 5

- Step 5.
  1. Open `Terminal` or `Git Bash`.
  2. `cd PATH_NAME_OF_THE_FOLDER_FOR_THE_WEBSITE` for `Terminal`.

     `cd "PATH_NAME_OF_THE_FOLDER_FOR_THE_WEBSITE"` for `Git Bash`.
  3. `git add --all` or `git add .`
  4. `git commit -m "update"`
  5. `git push -u origin main` or

     `git push -u origin master`

Personal website on GitHub

Step 6

- Step 6: Return to your GitHub repository in the browser.
  - Go to your personal website:
    `https://YOUR_GITHUB_NAME.github.io`

Personal website on GitHub

Updating your website

1. Modify `about.Rmd` as follows:

```
---
title: "About Me"
---

- __Name:__ YOUR_NAME
- __Ocupation:__ "Student"
- __Hobbies:__ Learning data analytics.

Here is a super cool photo.

![](FILE_NAME_OF_YOUR_PHOTO_IN_THE_REPO_FOLDER.jpg){
    width=50%}
```

2. Rebuild your site by running `rmarkdown::render_site()` in console.

3. Open `index.html` again to see if it worked.

Personal website on GitHub

Modifying your website

1. From our Canvas, download the website file—`style.css`—and save them inside your repository folder, `YOUR_GITHUB_NAME.github.io`, in your laptop.

- Cascading Style Sheets (CSS) is used to format the layout of a webpage (color, font, text size, background, display, etc.).
  - HTML will format the architecture of the house;
  - CSS will be the carpet and walls to decorate the house;
  - JavaScript adds interactive elements in the house, such as opening doors and lighting.

Personal website on GitHub

Modifying your website

2. Add the following lines to your `_site.yml` file to apply this css to your site.

```
name: "YOUR_NAMEs-website"
output_dir: "."
navbar:
  title: "YOUR_NAME's Website"
  left:
    - text: "Home"
      href: index.html
    - text: "About Me"
      href: about.html
output:
  html_document:
    theme: default
    css: style.css
```

3. Run `rmarkdown::render_site()` to checkout how things have changed.

Personal website on GitHub

Modifying your website

On `Terminal` or `Git Bash`,

4. Change the directory to your website folder:
   - `cd PATH_NAME_OF_THE_WEBSITE_FOLDER` for `Terminal`.
   - `cd "PATH_NAME_OF_THE_WEBSITE_FOLDER"` for `Git Bash`.
5. `git add --all` or
   `git add .`
6. `git commit -m "update your website"`
7. `git push -u origin main` or
   `git push -u origin master`

Personal website on GitHub

Modifying your website

The color scheme that we are currently using can be seen under the `theme` section of the `_site.yml` file.

There are others built in:

- `default`
- `cerulean`
- `journal`
- `flatly`
- `readable`
- `spacelab`
- `united`
- `cosmo`
- `lumen`
- `paper`
- `sandstone`
- `simplex`
- `yeti`

In the `_site.yml` file, you can replace "`default`" theme with other theme listed above.

Personal website on GitHub

Adding projects/other links

1. To add `project` page to your website, edit the `_site.yml` file as follows:

```
name: "YOUR_NAME's-website"
output_dir: "."
navbar:
  title: "YOUR_NAME's Website"
  left:
    - text: "Home"
      href: index.html
    - text: "About Me"
      href: about.html
    - text: "Project"      ##### the new
      href: project.html   ##### stuff
output:
  html_document:
    theme: default
    css: style.css
```

Personal website on GitHub

Adding projects/other links

2. From our Canvas, download the website file—`project.Rmd`—and save them inside your repository folder, `YOUR_GITHUB_NAME.github.io`, in your laptop.

Personal website on GitHub

Adding projects/other links

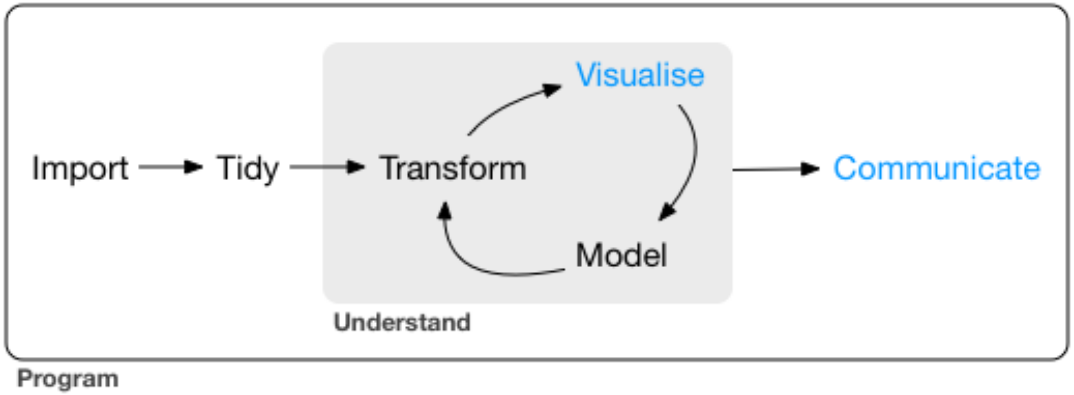
2. Inside the chunk of ```r, ... ``` in the `project.Rmd` file, you can put any other R code you want:

```
library(tidyverse)
ggplot(mpg) + geom_point(aes(x = cty, y = hwy))
```

3. Run `rmarkdown::render_site()`.
4. On `Terminal` or `Git Bash`,
   - `cd PATH_NAME_OF_THE_WEBSITE_FOLDER` for `Terminal`.
   - `cd "PATH_NAME_OF_THE_WEBSITE_FOLDER"` for `Git Bash`.
5. `git add --all` or `git add .`
6. `git commit -m "update your website"`
7. `git push -u origin main` or `git push -u origin master`

R Markdown

- You know the tools to get your data into R, tidy it into a form convenient for analysis, and then understand your data through transformation, visualization and modeling.



- R Markdown provides an unified authoring framework for data science, combining your code, its results, and your prose commentary.
- R Markdown documents are fully reproducible and support dozens of output formats, like HTML, PDF, docx, ppt, slideshows, and more.

R Markdown

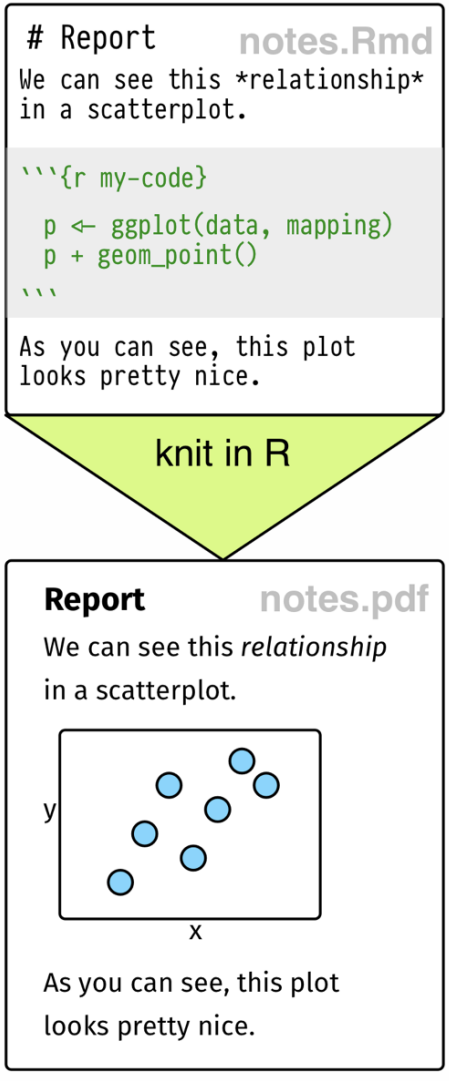
- R Markdown files are designed to be used in three ways:
  - For communicating to decision makers, who want to focus on the conclusions, not the code behind the analysis.
  - For collaborating with other data scientists (including future you!), who are interested in both your conclusions, and how you reached them (i.e. the code).
  - As an environment in which to do data science, as a modern day lab notebook where you can capture not only what you did, but also what you were thinking.

R Markdown

- Use R Markdown during the class and throughout the semester.
- Keep "*R Markdown Reference Guide*" close to hand (This PDF file is on Canvas).
- Must-know shortcuts:
  - **option+command+I** or **Alt+Ctrl+I** : to create a R chunk
  - **command+shift+return** or **Ctrl+Shift+Enter** : to run the code in the R chunk
  - **command + shift + K** or **Ctrl + Shift + K**: to knit the R Markdown file
  - **command + shift + C** or **Ctrl + Shift + C**: to (de-)comment out a line in the R Markdown file (`<!- YOUR_CODE ->` is used for commenting out `YOUR_CODE`.)

R Markdown

- An (optional) YAML (yet another markup language) header surrounded by ---s.
  - It is commonly used for configuration files (e.g., title, author, date, ...).
- Chunks of R code surrounded by ```.
- Text mixed with simple text formatting like # heading and _italics_.



R Markdown

- When you knit the document, R Markdown sends the `.Rmd` file to `knitr`, which executes all of the code chunks and creates a new markdown (.md) document which includes the code and its output.
- The markdown file generated by `knitr` is then processed by `pandoc`, which is responsible for creating the finished file.

- I suggest you to first create the project by selecting **File > New Project... > Existing Directory** in the menu bar.
- Then get started with your own `.Rmd` file by selecting **File > New File > R Markdown...** in the menu bar.