SM-2302 Software for Mathematicians

R0 & Git: Getting started

Dr. Haziq Jamil
Mathematical Sciences, Faculty of Science, UBD
https://haziqj.ml

Semester I 2023/24

Learning objectives

- Introduction to R, RStudio and learn the difference between the two
- Familiarisation with RStudio layout and customising appearance
- Using the help (?) function
- `base` package and installing other packages
- Setting up working directory and project area
- Using R script files vs working through the console
- Importing data

Highly recommended book:
`https://rstudio-education.github.io/hopr/index.html`

Why learn R?

https://towardsdatascience.com/a-complete-guide-to-learn-r-29e691c61d1

Before we start

Preamble

Before proceeding, some best practices on how to properly conduct data analysis:

1. Keep all files in one folder (working directory), including data file, R scripts, etc.
2. When working with large amounts of files, perhaps better to organise into sub-folders (e.g. folders for code, figures, data, etc.)
3. Use simple naming conventions for files and variables (no spaces, no caps, no special characters, etc.)
4. Create an RStudio project file so that the working directory, environment, code history, etc. is preserved
5. Collect all your R code into R scripts. Don't rely on the console.

Hello, World!

```
my_string <- "Hello, World!"
print(my_string)
```

```
## [1] "Hello, World!"
```

Titanic data analysis

Create an R Project containing the the files in the `R Demo` folder from Canvas. You may run the code line by line.

Observe the way the code is written and formatted, as well as where comments are placed.

`https://style.tidyverse.org/`

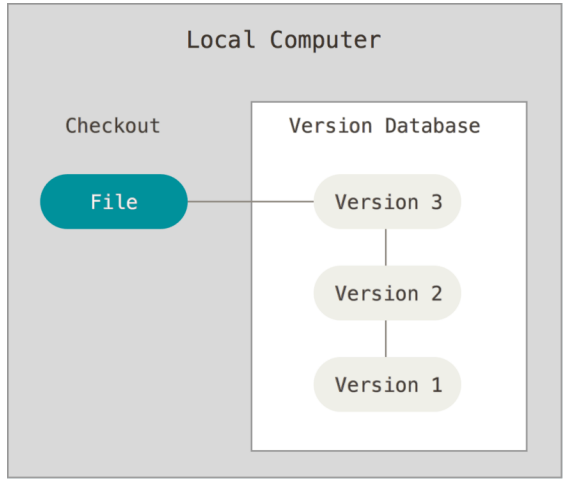
R

Git and GitHub

- Version control
- Configure Git
- Workflow and best practices

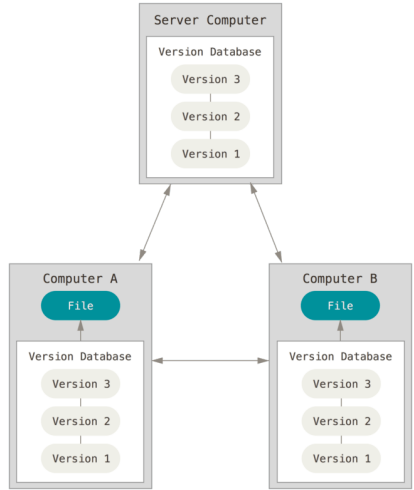
Why version control?

- Simple formal system for tracking changes to a project over time
- Time machine for your projects
  - Track blame and/or praise
  - Remove the fear of breaking things
- Learning curve can be a bit steep, but when you need it you *REALLY* need it



Why Git?

- Distributed
  - Work online or offline
  - Collaborate with large groups
- Popular and Successful
  - Active development
  - Shiny new tools and ecosystems
  - Fast
- Tracks any type of file
- Branching
  - Smarter merges



Veryifying git installation

Git should already be installed in the lab PCs. Verify by launching the terminal and typing

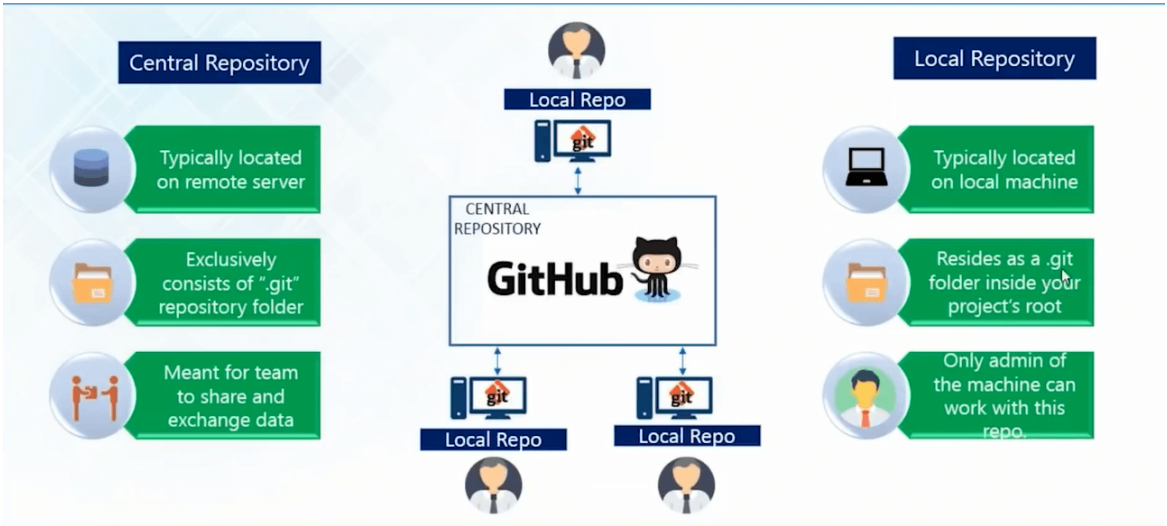
```
haziqj@Naqiyyah-MBP ~ % git --version
git version 2.32.1 (Apple Git-133)
```

On your own PCs, you can install git by following the directions in Happy Git and GitHub for the useR.

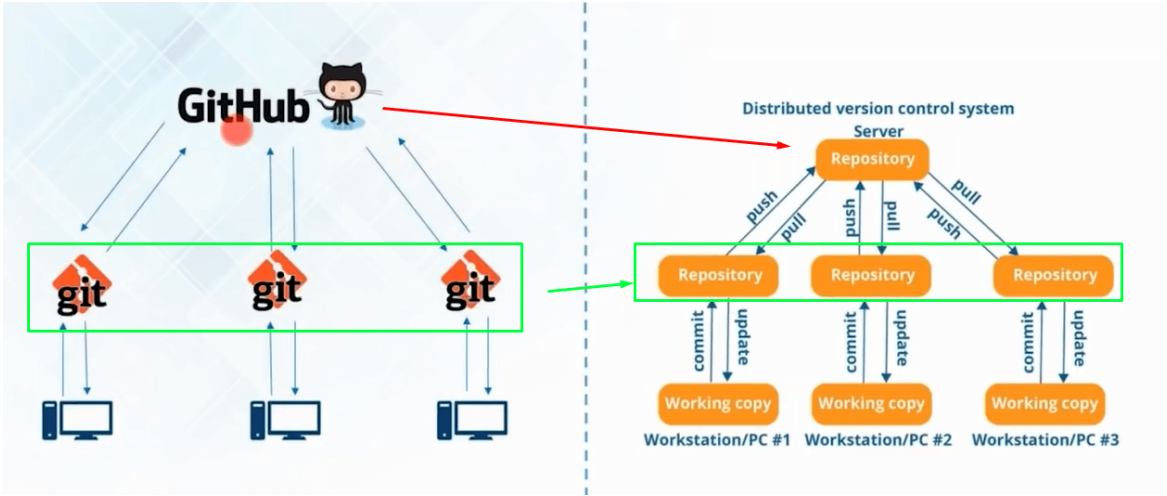
Git sitrep

```
usethis::git_sitrep()
## Git config (global)
## • Name: <unset>
## • Email: <unset>
## • Global (user-level) gitignore file: <unset>
## • Vaccinated: FALSE
##   See `?git_vaccinate` to learn more
##   Defaulting to 'https' Git protocol
## • Default Git protocol: 'https'
## • Default initial branch name: <unset>
## GitHub
## • Default GitHub host: 'https://github.com'
## • Personal access token for 'https://github.com': <unset>
## • To create a personal access token, call `create_github_token()`
## • To store a token for current and future use, call `gitcreds::gitcreds_set()`
##   Read more in the 'Managing Git(Hub) Credentials' article:
##    https://usethis.r-lib.org/articles/articles/git-credentials.html
## Git repo for current project
##   No active usethis project
```

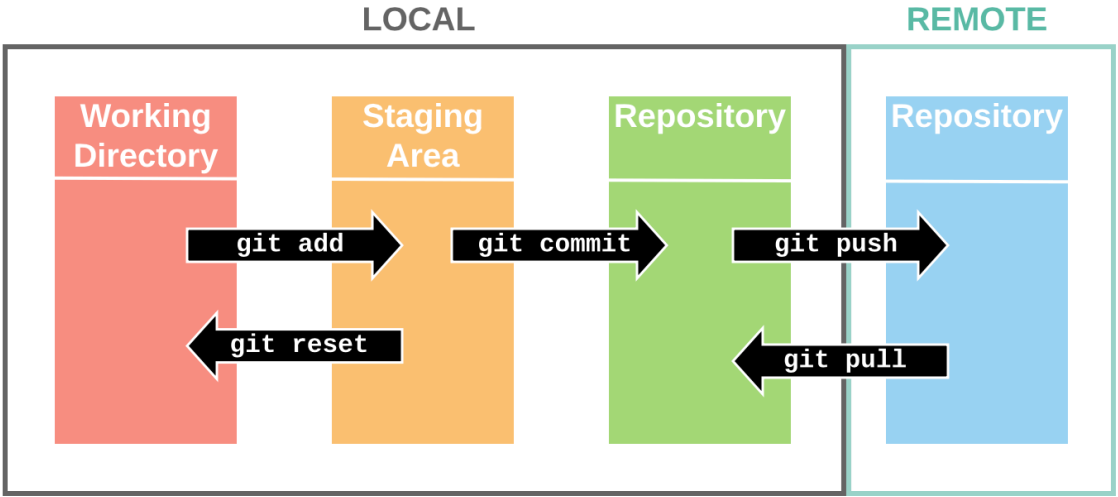
Git vs GitHub



Git vs GitHub (cont.)



Git in a nutshell



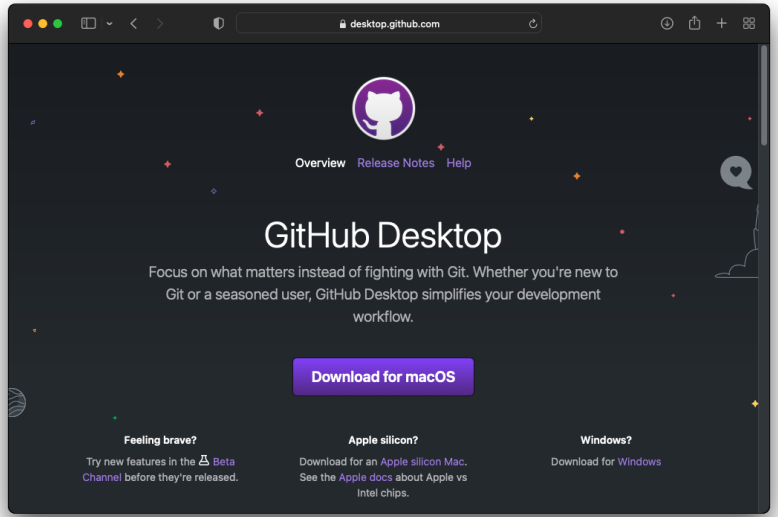
https://support.nesi.org.nz/hc/en-gb/articles/360001508515-Git-Reference-Sheet

Quick tutorial

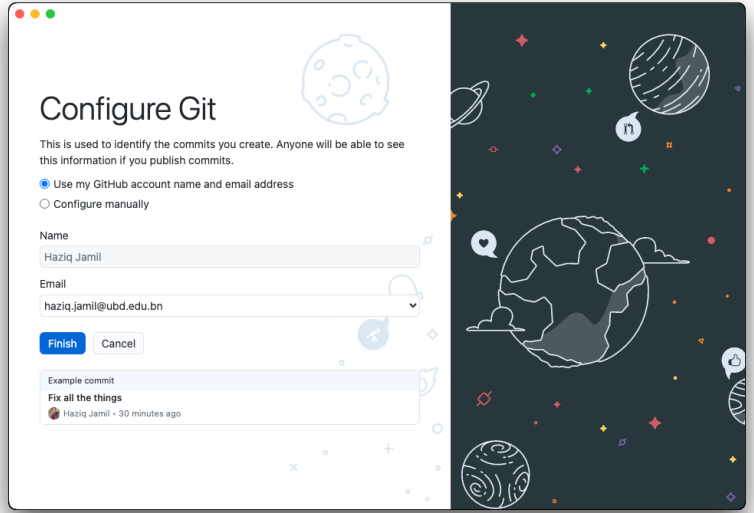
Follow along

`https://docs.github.com/en/get-started/quickstart/hello-world`

Install GitHub Desktop



Configure Git



Configure Git (cont.)

Remark

Do this only if sitrep still shows wrong info.

The following will tell Git who you are, and other common configuration tasks.

```
usethis::use_git_config(
  user.name = "Haziq Jamil",
  user.email = "haziq.jamil@ubd.edu.bn"
  # push.default = "simple",
  # pull.rebase = FALSE
)
```

This can also be done via the terminal with,

```
$ git config --global user.name "Haziq Jamil"
$ git config --global user.email "haziq.jamil@ubd.edu.bn"
$ git config --global push.default simple
```

Typical GitHub workflow

I want to start a project that involves some code.

1. Go to GitHub.com and create a new repo.
   - Can initialise accordingly (.gitignore and/or README)
2. Clone to local repo
3. Add code inside
4. Commit and push

   Not really going to spend much time branching and creating pull requests

Version control best practices

- Commit early, often, and with complete code.
- Write clear and concise commit summary messages.
- Test code before you commit.
- Use branches.
- Communicate with your team.

Git and GitHub resources

- Git's Pro Git book, Chapters Getting Started and Git Basics will be most useful if you are new to Git and GitHub
- Git cheatsheet by Atlassian
- GitHub's interactive tutorial
- Free online course from Udacity
- Happy Git with R by Jenny Bryan