RStudio, Git and GitHub

Version Control

Introduction

"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.
CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.
corrections9.MORE.30.doc
FINAL_rev.22.comments49.
corrections.10.#@$%WHYDID
ICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012
WWW.PHDCOMICS.COM

Fig 1: Source - “notFinal.doc” by Jorge Cham, https://www.phdcomics.com

- **Version Control** can be thought of as a recording of your progress: you can rewind to start at the base document and play back each change you made, eventually arriving at one of your more recent versions.
- In other words, we can “play” the different *versions* of the document of interest i.e. version control system is a tool that keeps track of our changes, effectively creating different *versions* of our files.
- Version control is useful, whether you are working alone or in a team.

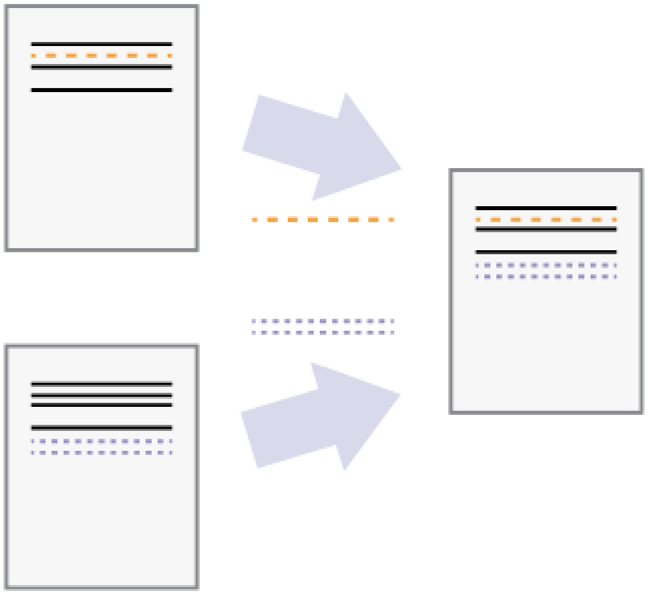


Fig 2: Let us say that two users make independent sets of changes on the same document. Unless there is a conflict, you can incorporate two sets of changes into the same base document.

- A **conflict** is a change made by one user of version control system that is incompatible with changes made by other users.
- Version control system requests human assistance when two people make conflicting edits.
- Version control allows us to decide which changes will be made to the next version, and keeps useful metadata about them.
- For any part of a document, you can determine when, why, and by whom it was ever edited.
- Each person edits his or her own copy of the documents and chooses when to share those changes with the rest of the team. Thus, temporary or partial edits by one person do not interfere with another person’s work.

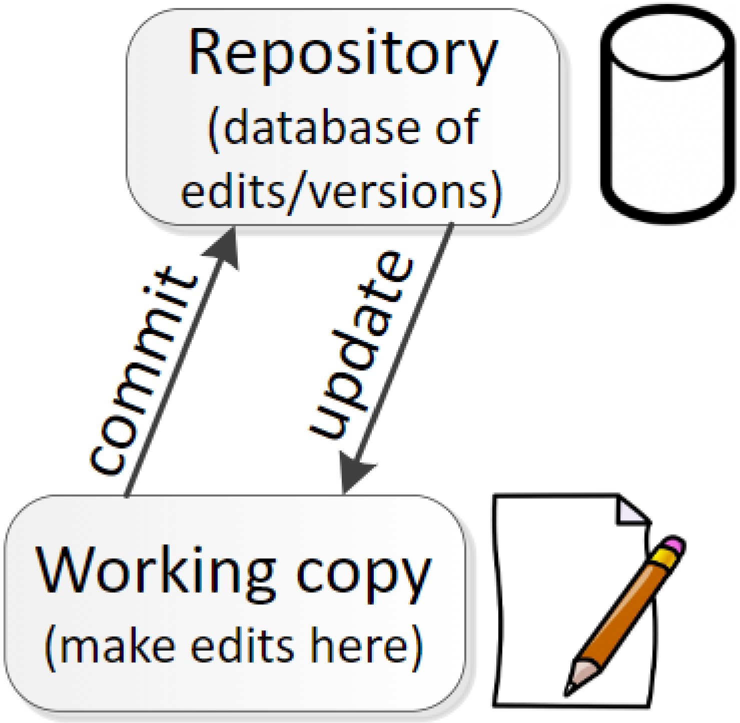
Summary

1. Version control is like an unlimited ‘undo’.
2. Version control also allows many people to work in parallel.

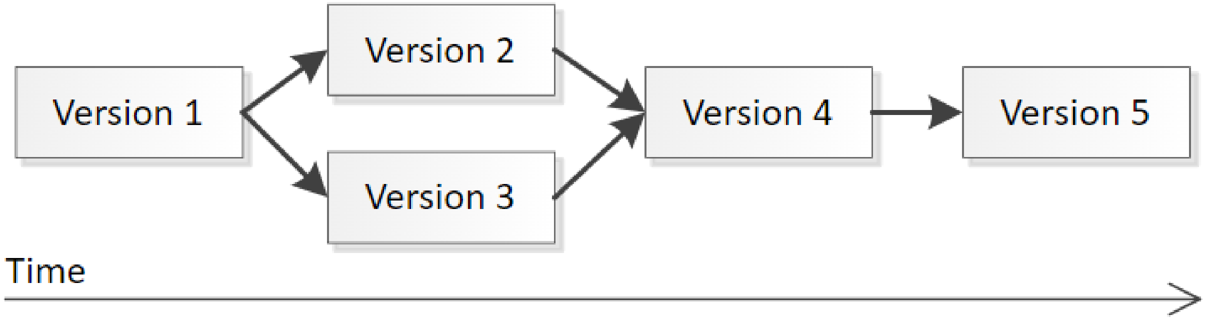
Git is a kind of version control system. Some other examples of version control are Subversion, Mercurial etc. In this class, we focus on Git.

Repositories and working copies

Version control uses: - **Repository**: database of all edits or versions of your project. - **Working copy**: A personal copy of all the files in the project. You can make arbitrary edits to this copy, without affecting your teammates.

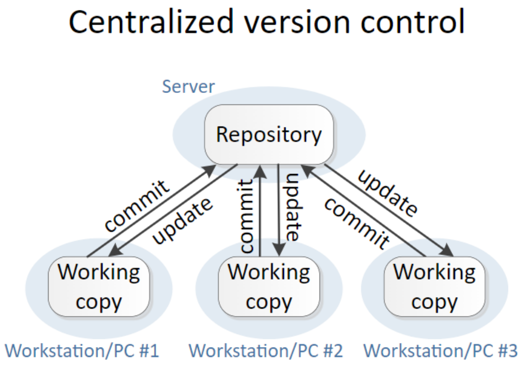


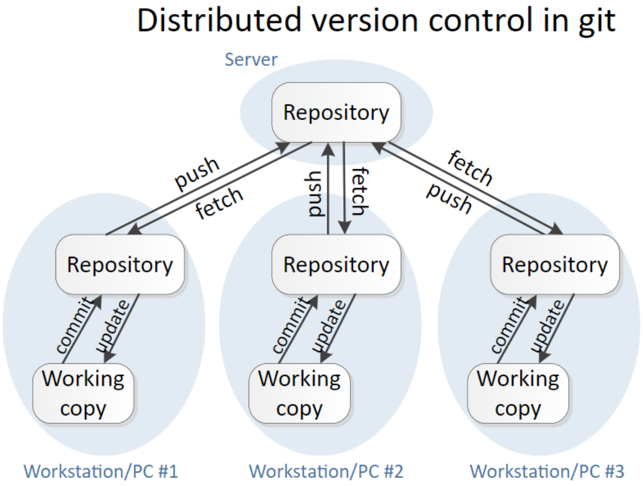
- When you are happy with your edits, you **commit** your changes to a repository.
- You can **update** your working copy to incorporate any new edits or versions that have been added to the repository since the last time you updated.
- If different users made edits simultaneously (this is sometimes called **branching**), the the version history splits and then merges again. In the example below, Version 4 is called a *merge*.



Distributed and centralized Version Control

- There are two types of version control that mainly differ in the number of repositories:
  - Centralized version control: there is one repository
  - Distributed version control: there are multiple repositories.
- Distributed version control is more modern while centralized version control is rarely used these days. Git, the most popular version control, is a distributed version control system.





Centralized version control	Distributed version control in git
Each user gets their own working copy, but there is just one central repository	Each user gets their own repository *and* working copy.
As soon as you commit, it is possible for your co-workers to update and to see your changes	- After you commit, others have no access to your changes until you push your changes to the central repository - When you update, you do not get others' changes unless you have first fetched those changes into your local repository
For others to see your changes, two things must happen - You commit - They update	For others to see your changes, four things must happen: - You commit - You push - They fetch - They update

Git and GitHub

- **GitHub** is a cloud-based platform where you can store, share, and work together with others to write code.
- GitHub is built upon Git. Git is what makes collaborative working possible while using GitHub.
- When you upload files to GitHub, you'll store them in a "Git repository."
- This means that when you make changes (or "commits") to your files in GitHub, Git will automatically start to track and manage your changes.
- Most people work on their files locally (on their own computer), then continually sync these local changes—and all the related Git data—with the central "remote" repository on GitHub.

There are alternatives to GitHub, like GitLab and Bitbucket, but we will be focusing only on GitHub.

Initial set-up

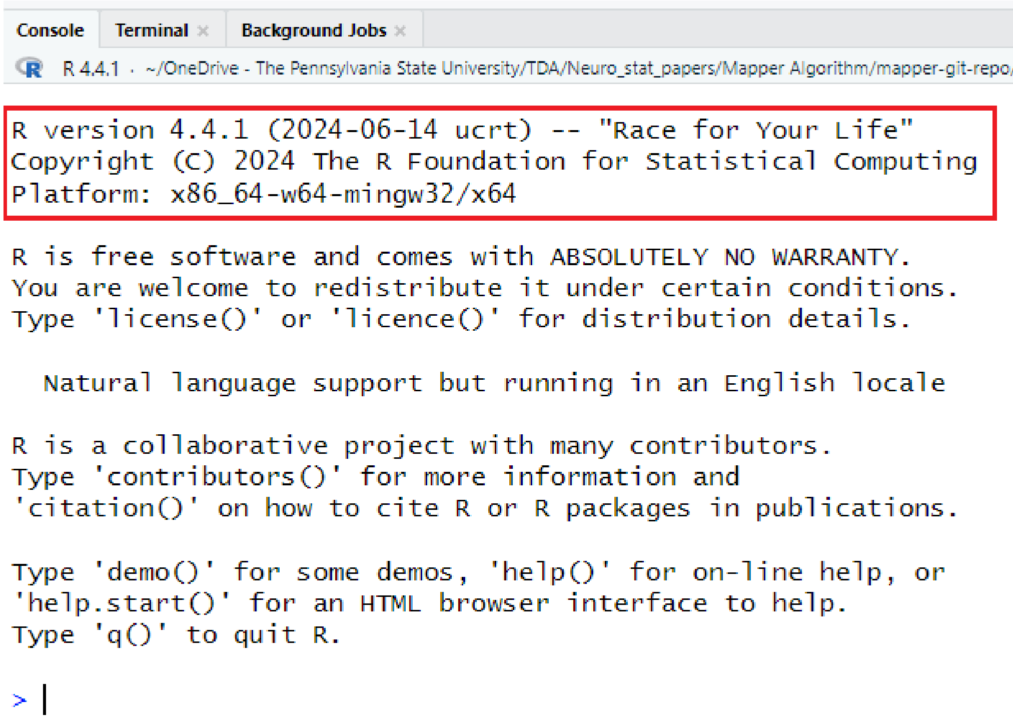
- Register a free GitHub account
- Install or upgrade R and RStudio
- Install Git

How to update the R version in your RStudio? Identify the current R version you are using with the following code:

```
# We can see R version in the console as soon as we start a new R session
# Other option is the command below
R.version.string
```

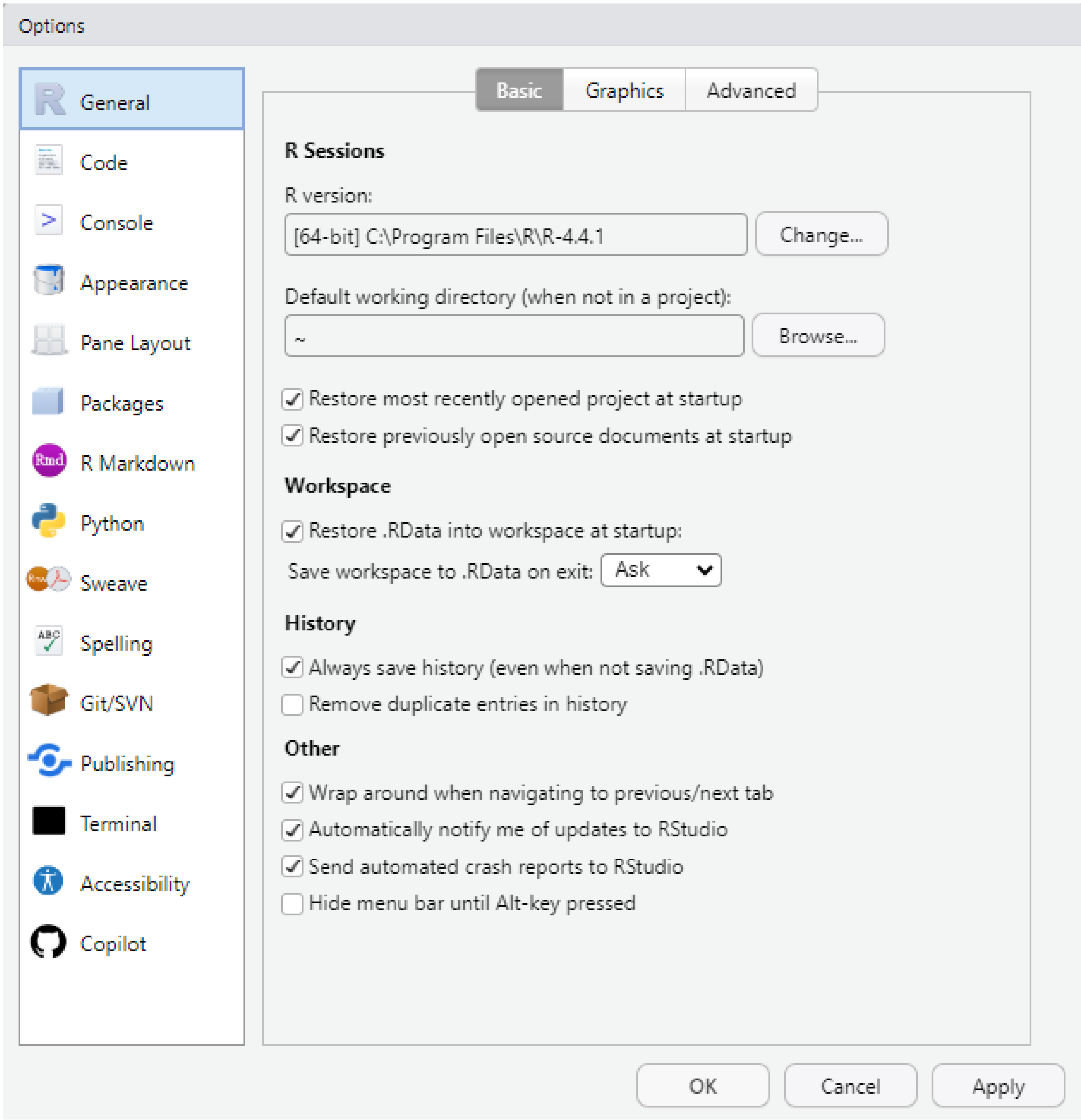
```
## [1] "R version 4.4.1 (2024-06-14 ucrt)"
```

- Go to the CRAN website.
  - For Windows Users: https://cran.r-project.org/bin/windows/base/
  - For Mac Users: https://cran.r-project.org/bin/macosx/
- Download and run the latest R installer. This will update R.
- Now, open/re-open RStudio.
- Check if RStudio has detected the updated R version. As soon a new R session starts, you can see R version information in the console.


Console
Terminal
Background Jobs
R 4.4.1 · ~/OneDrive - The Pennsylvania State University/TDA/Neuro_stat_papers/Mapper Algorithm/mapper-git-repo/

```
R version 4.4.1 (2024-06-14 ucrt) -- "Race for Your Life"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

- If RStudio doesn't automatically detect the latest R version, do the following: Tools » Global options.



R Sessions

R version: [64-bit] C:\Program Files\R\R-4.4.1

Default working directory (when not in a project): ~

☑ Restore most recently opened project at startup
☑ Restore previously open source documents at startup

Workspace

☑ Restore .RData into workspace at startup:
Save workspace to .RData on exit: Ask

History

☑ Always save history (even when not saving .RData)
□ Remove duplicate entries in history

Other

☑ Wrap around when navigating to previous/next tab
☑ Automatically notify me of updates to RStudio
☑ Send automated crash reports to RStudio
□ Hide menu bar until Alt-key pressed

- Select the appropriate R version in the “R version” section of the right panel using the `change` button.
- After choosing the appropriate R version, click on “Apply” and then “OK” to complete the process.

References:

1. Version control with Git by Software Carpentry.
2. Version control concepts and best practices, notes by M Ernst
3. Happy Git and GitHub for the useR
4. About GitHub and Git