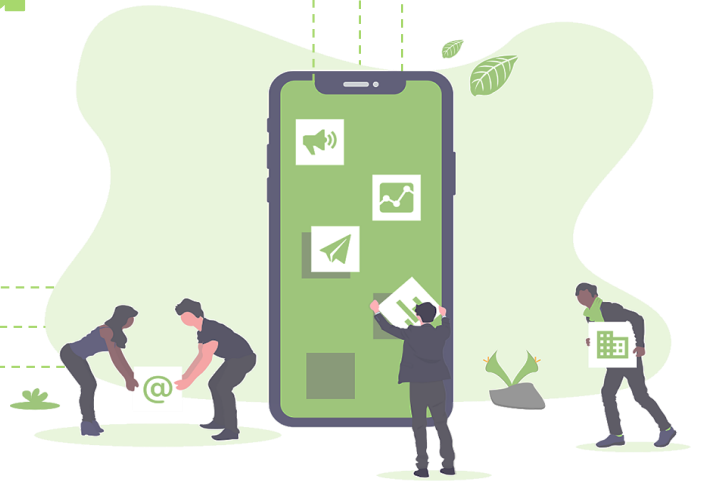


The Data Science Track



Prepared By: R. Daynalo

1

1

4. Creating a GitHub Repository



2

1



Recap: Git vs. GitHub




- You don't need GitHub to use Git
- Git = Local (on your computer); GitHub = Remote (on the web)
- GitHub allows you to:
 - Share your repositories with others
 - Access other users' repositories
 - Store remote copies of your repositories (on GitHub server) in case something happens to your local copies (on your computer)




Prepared By: R. Daynolo

3


3



Creating a GitHub Repository



- Two methods of creating a GitHub repository:
 1. Start a repository from scratch
 2. "Fork" another user's repository
- We'll start with the first method
- Note: A repository is often referred to as simply "repo"



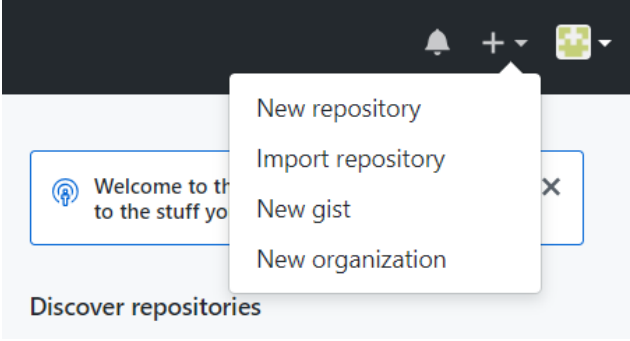
Prepared By: R. Daynolo

4

4

Start a Repository From Scratch

- Go to your profile page (<https://github.com/yourUserNameHere/>) and click on "New repository" in the upper righthand corner of the page



The screenshot shows the GitHub profile page with a dropdown menu open in the top right corner. The menu options are: New repository, Import repository, New gist, and New organization. The background shows a 'Welcome to GitHub' message and a 'Discover repositories' section.

Prepared By: R. Daynolo


5

Start a Repository From Scratch

- Create a name for your repo and type a brief description of it
- Select "Public" (Private repos require a paid [or education] account)
- Check the box next to "Initialize this repository with a README"
- Click the "Create repository" button

Prepared By: R. Daynolo

6



Start a Repository From Scratch

Create a new repository

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

Owner: User / **test-repo** ✓

Great repository names are short and memorable. Need inspiration? How about [jubilant-octo-giggle](#).

Description (optional)



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

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

☒ **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: **None** Add a license: **None** ⓘ

Create repository

7

7



Start a Repository From Scratch

User / **test-repo** Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

No description, website, or topics provided. [Edit](#)

[Manage topics](#)

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file **Clone or download**



RDayns Initial commit Latest commit 12d16fa 17 seconds ago

[README.md](#) Initial commit 17 seconds ago

README.md


test-repo

Prepared By: R. Daynola





8

8



Creating a Local Copy




- Now you need to create a copy of this repo on your computer so that you can make changes to it
- Open Git Bash
- Create a directory on your computer where you will store your copy of the repo:

```
$ mkdir ~/test-repo
```

- Navigate to this new directory using the following command:


```
$ cd ~/test-repo
```




Prepared By: R. Daynolo

9

9



Creating a Local Copy




- Initialize a local Git repository in this directory

```
$ git init
```

- Point your local repository at the remote repository you just created on the GitHub server

```
$ git remote add origin https://github.com/yourUsernameHere/test-repo.git
```



Prepared By: R. Daynolo

10

10



Creating a Local Copy



```

Welcome to Git (version 1.8.4-preview20130916)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Nick@NICK-PC ~
$ mkdir ~/test-repo

Nick@NICK-PC ~
$ cd ~/test-repo

Nick@NICK-PC ~/test-repo
$ git init
Initialized empty Git repository in c:/Users/Nick/test-repo/.git/

Nick@NICK-PC ~/test-repo (master)
$ git remote add origin https://github.com/ncarchedi/test-repo.git

Nick@NICK-PC ~/test-repo (master)
$
  
```

Prepared By: R. Daynolo

11

11



Fork Another User's Repository



- The second method of creating a repository is to make a copy of someone else's
- This process is called "forking" and is an important aspect of open-source software development
- Begin by navigating to the desired repository on the GitHub website and click the "Fork" button shown below



<https://help.github.com/articles/fork-a-repo>

Prepared By: R. Daynolo

12

12



Clone the Repo



- You now have a copy of the desired repository on your GitHub account
- Need to make a local copy of the repo on your computer
- This process is called “cloning” and can be done using the following command

```
$ git clone https://github.com/yourUsernameHere/repoNameHere.git
```

- Note: This will clone the repository into your current directory



Prepared By: R. Daynolo

13

13



What Else?



- If you make changes to your local copy of the repo, you’ll probably want to push your changes to the GitHub at some point
- You also may be interested in staying current with the changes made to the original repository from which you forked your copy
- Some great resources:
 - <https://help.github.com/articles/fork-a-repo/>
 - <https://git-scm.com/book/en/v2/Git-Basics-Getting-a-Git-Repository>



Prepared By: R. Daynolo

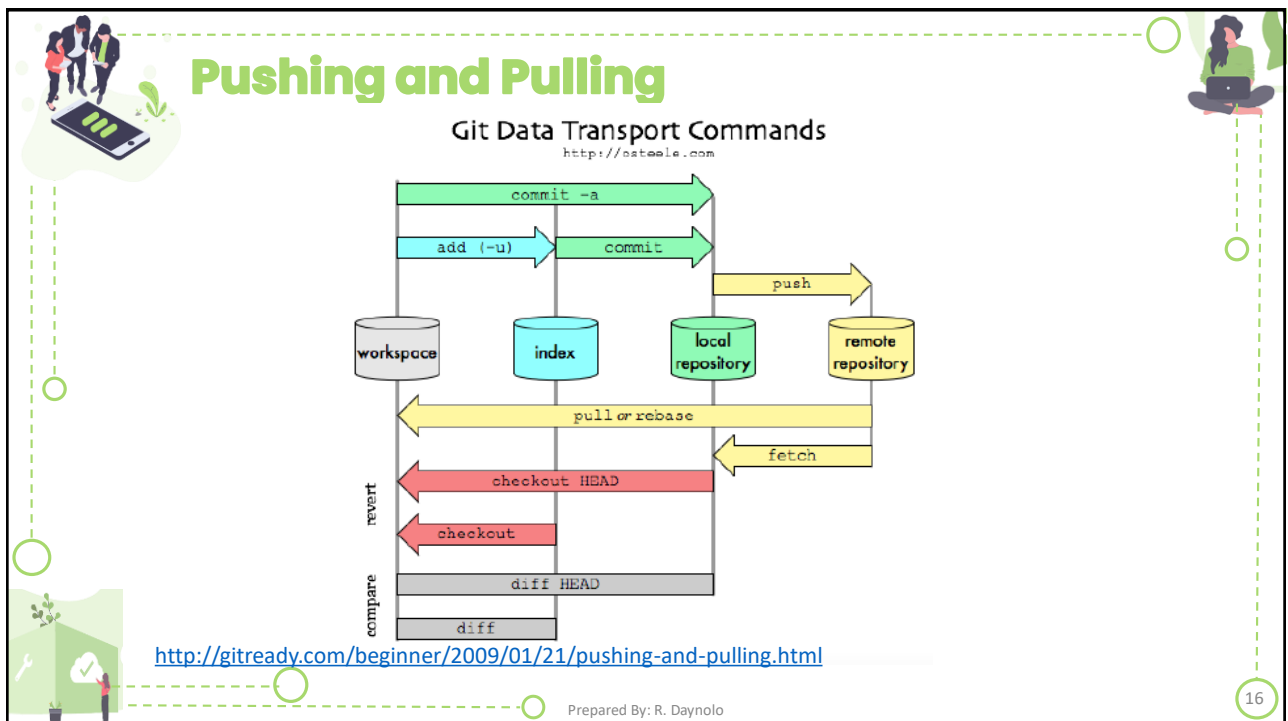
14

14


5. Basic Git Commands




15




16



Adding




- Suppose you add new files to a local repository under version control
- You need to let Git know that they need to be tracked
 - `git add .` adds all new files
 - `git add -u` updates tracking of files that changed names or were deleted
 - `git add -A` does both of the previous
- You should do this before committing




Prepared By: R. Daynolo

17


17



Committing



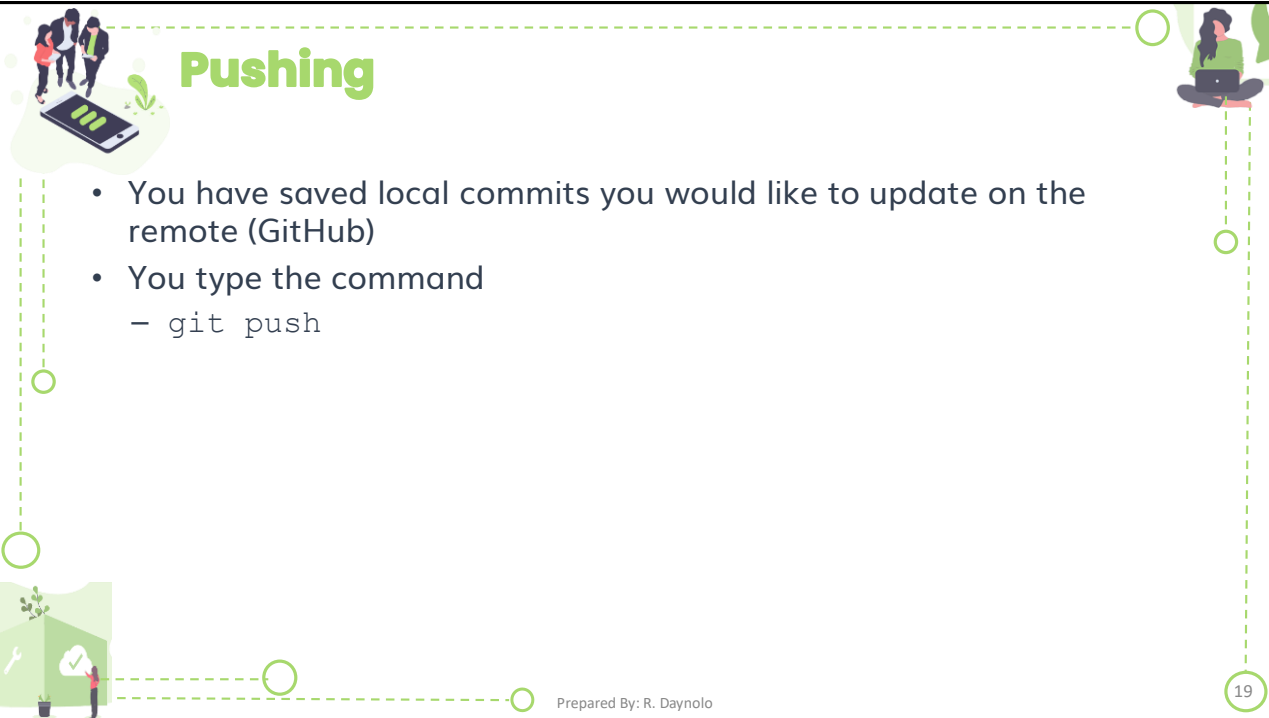
- You have changes you want to commit to be saved as an intermediate version
- You type the command
 - `git commit -m "message"` where message is a useful description of what you did
- This only updates you local repo, not the remote repo on GitHub



Prepared By: R. Daynolo

18

18



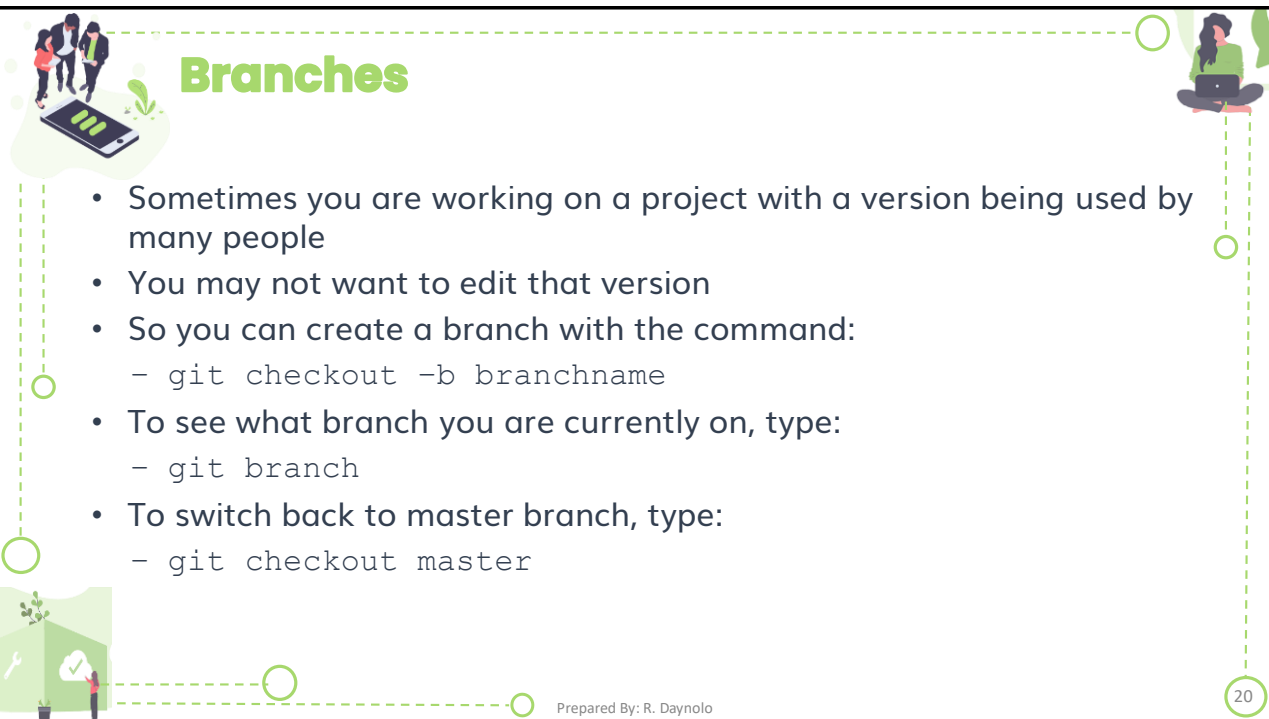
Pushing

- You have saved local commits you would like to update on the remote (GitHub)
- You type the command
 - `git push`

Prepared By: R. Daynolo

19

19




Branches

- Sometimes you are working on a project with a version being used by many people
- You may not want to edit that version
- So you can create a branch with the command:
 - `git checkout -b branchname`
- To see what branch you are currently on, type:
 - `git branch`
- To switch back to master branch, type:
 - `git checkout master`


Prepared By: R. Daynolo

20

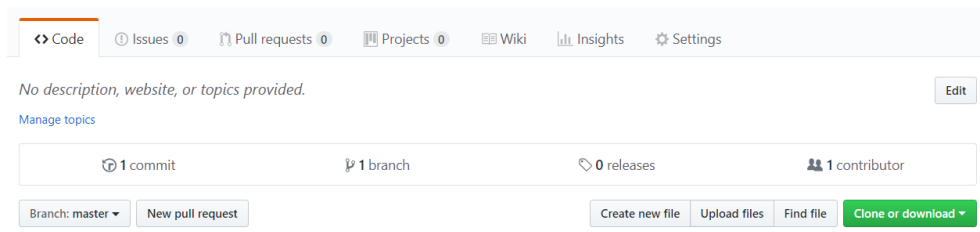
20



Pull Requests




- If you fork someone's repo or have multiple branches you will both be working separately
- Sometimes you want to merge in your changes into the other branch/repo
- To do so, you need to send a pull request
- This is a feature of GitHub




Prepared By: R. Daynola

21



Time to be a hacker!



- Git documentation: <http://git-scm.com/doc>
- GitHub help: <https://help.github.com/>
- Google/Stack Overflow


Prepared By: R. Daynola

22


6. Basic Markdown



23




Markdown



- text-to-HTML conversion tool for web writers. Markdown allows you to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML (or HTML)
- Created by John Gruber in 2004


<https://daringfireball.net/projects/markdown/>



Prepared By: R. Daynalo

24

24





Markdown Syntax

- Headings

```
## This is a secondary heading
### This is a tertiary heading
```

This is a secondary heading


This is a tertiary heading

Prepared By: R. Daynolo

25

25





Markdown Syntax

- Unordered List

```
* first item in list
* second item in list
* third item in list
```

- first item in list
- second item in list
- third item in list

Prepared By: R. Daynolo


26

26


7. Installing R Packages




27



R Packages



- When you download R from the Comprehensive R Archive Network (CRAN), you get that “base” R system
- The base R system comes with basic functionality; implements the R language
- One reason R is so useful is the large collection of packages that extend the basic functionality of R
- R packages are developed and published by the larger R community



Prepared By: R. Daynalo

28

28

Obtaining R Packages

- The primary location for obtaining R packages is **CRAN**.
- For biological applications, many packages are available from the **Bioconductor Project**.
- You can obtain information about the available packages on CRAN with the `available.packages()` function.

```
a <- available.packages()
head(rownames(a), 3) ## Show the names of the first few packages
```

```
## [1] "A3"      "abc"     "abcdeFBA"
```

- Currently, the CRAN package repository features **13,626** available packages, covering a wide range of topics.

Prepared By: R. Daynola

29

29

Installing R Packages

- Packages can be installed with the `install.packages()` function in R.
- To install a single package, pass the name of the package to the `install.packages()` function as its first argument.
- The following code installs the **ggplot2** package from CRAN

```
install.packages("ggplot2")
```

- The command downloads the **ggplot2** package from CRAN and installs it on your computer
- Any packages on which this package depends will also be downloaded and installed.

Prepared By: R. Daynola

30

30

Installing R Packages

- You can install multiple R packages at once with a single call to `install.packages()`
- Place the names of the R packages in a character vector

```
install.packages(c("ggplot2", "lattice", "devtools"))
```

Prepared By: R. Daynola

31

31

Installing R Packages in RStudio

The screenshot shows the RStudio interface with the 'User Library' pane open. The pane displays a list of installed and available packages. The 'Name' column lists the package names, the 'Description' column provides a brief description, and the 'Version' column shows the installed version. The 'Status' column indicates whether the package is installed or available for installation.

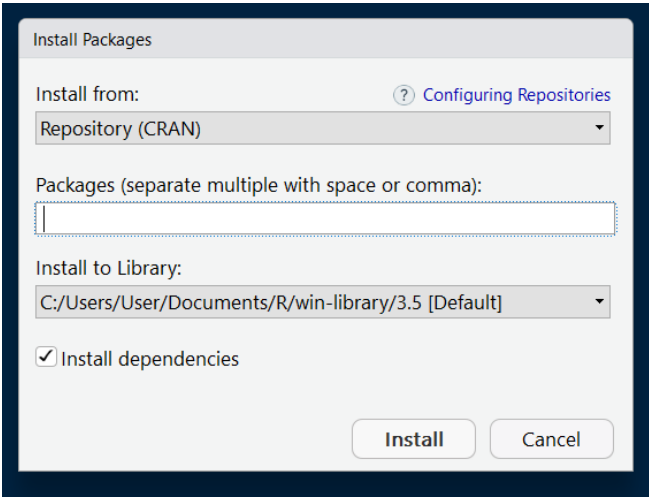
Name	Description	Version	Status
abind	Combine Multidimensional Arrays	1.4-5	●
acepack	ACE and AAS for Selecting Multiple Regression Transformations	1.4-1	●
AppliedPredictiveMo...	Functions and Data Sets for 'Applied Predictive Modeling'	1.1-7	●
assertthat	Easy Pre and Post Assertions	0.2.0	●
backports	Reimplementations of Functions Introduced Since R-3.0.0	1.1.2	●
base64enc	Tools for base64 encoding	0.1-3	●
BayesFactor	Computation of Bayes Factors for Common Designs	0.9.12-4.2	●
BH	Boost C++ Header Files	1.66.0-1	●
bindr	Parameterized Active Bindings	0.1.1	●
bindrcpp	An 'Rcpp' Interface to Active Bindings	0.2.2	●
bitops	Bitwise Operations	1.0-6	●
broom	Convert Statistical Analysis Objects into Tidy Tibbles	0.5.0	●
devtools	Tools for Managing R Packages	1.11.0	●

Prepared By: R. Daynola

32

32

Installing R Packages in RStudio



The image shows the 'Install Packages' dialog box in RStudio. It has a title bar 'Install Packages'. Below it, there's a section 'Install from:' with a dropdown menu set to 'Repository (CRAN)' and a link '? Configuring Repositories'. Below that is a text field for 'Packages (separate multiple with space or comma):'. Then, 'Install to Library:' with a dropdown menu set to 'C:/Users/User/Documents/R/win-library/3.5 [Default]'. At the bottom, there's a checked checkbox for 'Install dependencies' and two buttons: 'Install' and 'Cancel'.

Prepared By: R. Daynola

33

Installing an R Package from Bioconductor Project

- To install core packages, type the following in an R command window:




```
if (!requireNamespace("BiocManager"))
  install.packages("BiocManager")
BiocManager::install()
```
- To install specific packages, e.g., "GenomicFeatures" and "AnnotationDbi", with


```
BiocManager::install(c("GenomicFeatures", "AnnotationDbi"))
```

<http://bioconductor.org/install/>

Prepared By: R. Daynola

34




Installing an R Package from Bioconductor Project

- Installing a package does not make it immediately available to you in R; you must load the package
- The `library()` function is used to load packages into R
- The following code is used to load the `ggplot2` package into R

```
library(ggplot2)
```

Note: Do not put the package name in quotes!



Prepared By: R. Daynalo

35