These slides are intended to illustrate the steps for interacting with GitHub and Git to get (fork) the template file for the second programming assignment in the Coursera R Programming course (involving lexical scoping) from Roger Peng's GitHub repository, cloning a copy onto your computer, and then pushing your completed cachematrix.R file back onto your GitHub repository for this assignment.

Figuring out how to interact with GitHub was about the hardest thing for me when I took the R Programming course (I hadn't taken The Data Scientist's Toolbox course), and I'm no expert on GitHub or Git to say the least -- but perhaps the following might help. First, if you like using GUI interfaces, see this article "Configuring RStudio / Github Integration -- Windows Version" by Leonard Greski, or instructions on GitHub for GitHub Desktop.

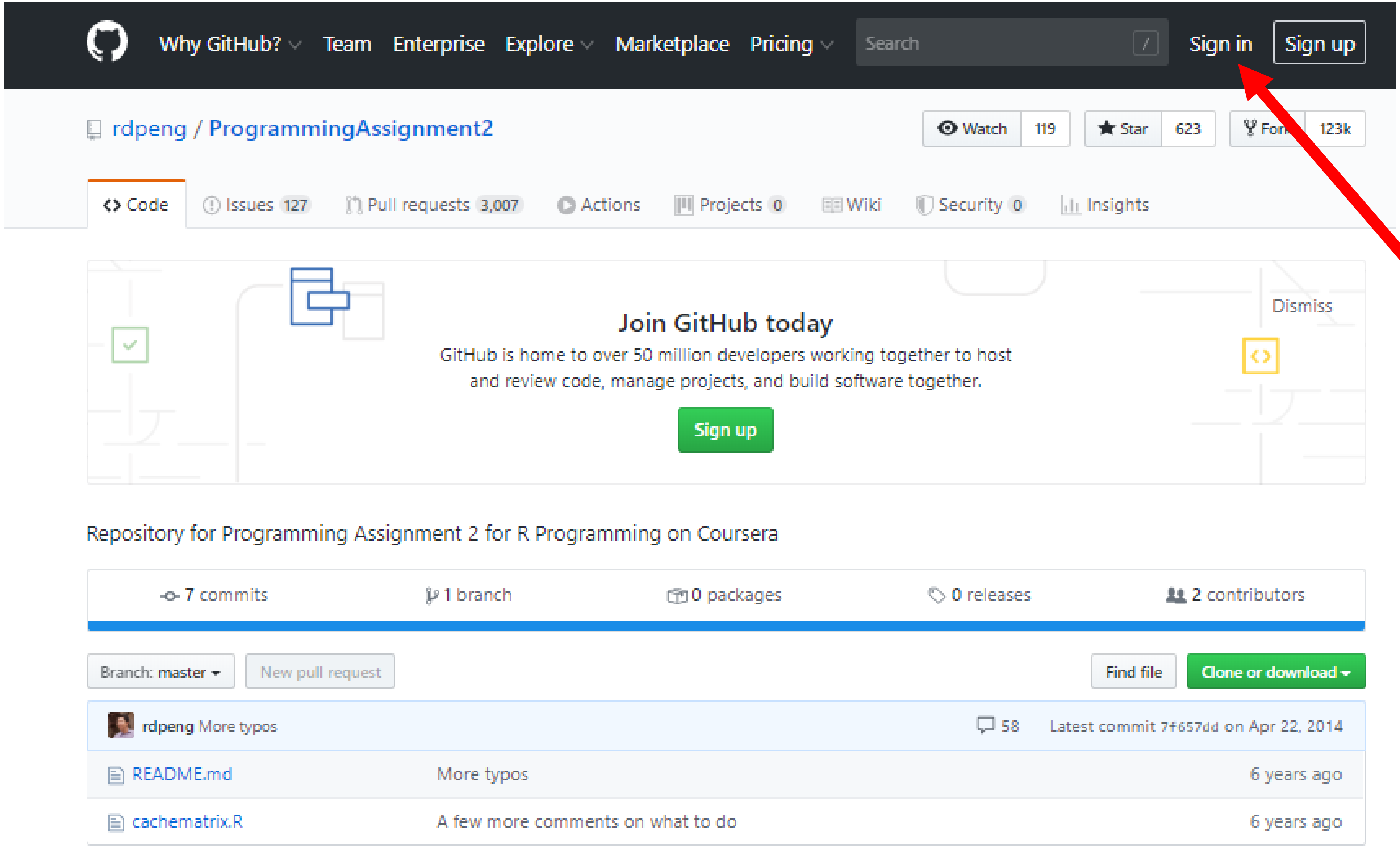
For using the Git command line interface, you might try the following.

Hope this is helpful, Alan Berger 19 May 2020

If you haven't created your own GitHub account, go here to create one. Then when you want to log in to GitHub go here: https://github.com/login To interact with GitHub via command line, you also need to download **Git** which gives a Unix style command line interface (Git Bash) for managing GitHub repositories: go to this web site: https://git-scm.com/downloads to obtain it.

For the lexical scoping assignment, you need to get a copy of the "template" cachematrix.R file where you will fill in your makeCacheMatrix and cacheSolve functions. To do this, following the instructions for the assignment, go to this GitHub repository belonging to Roger Peng

If you were not already logged into your GitHub account, the web page you will see is (next slide)



rdpeng / ProgrammingAssignment2

Join GitHub today

GitHub is home to over 50 million developers working together to host and review code, manage projects, and build software together.

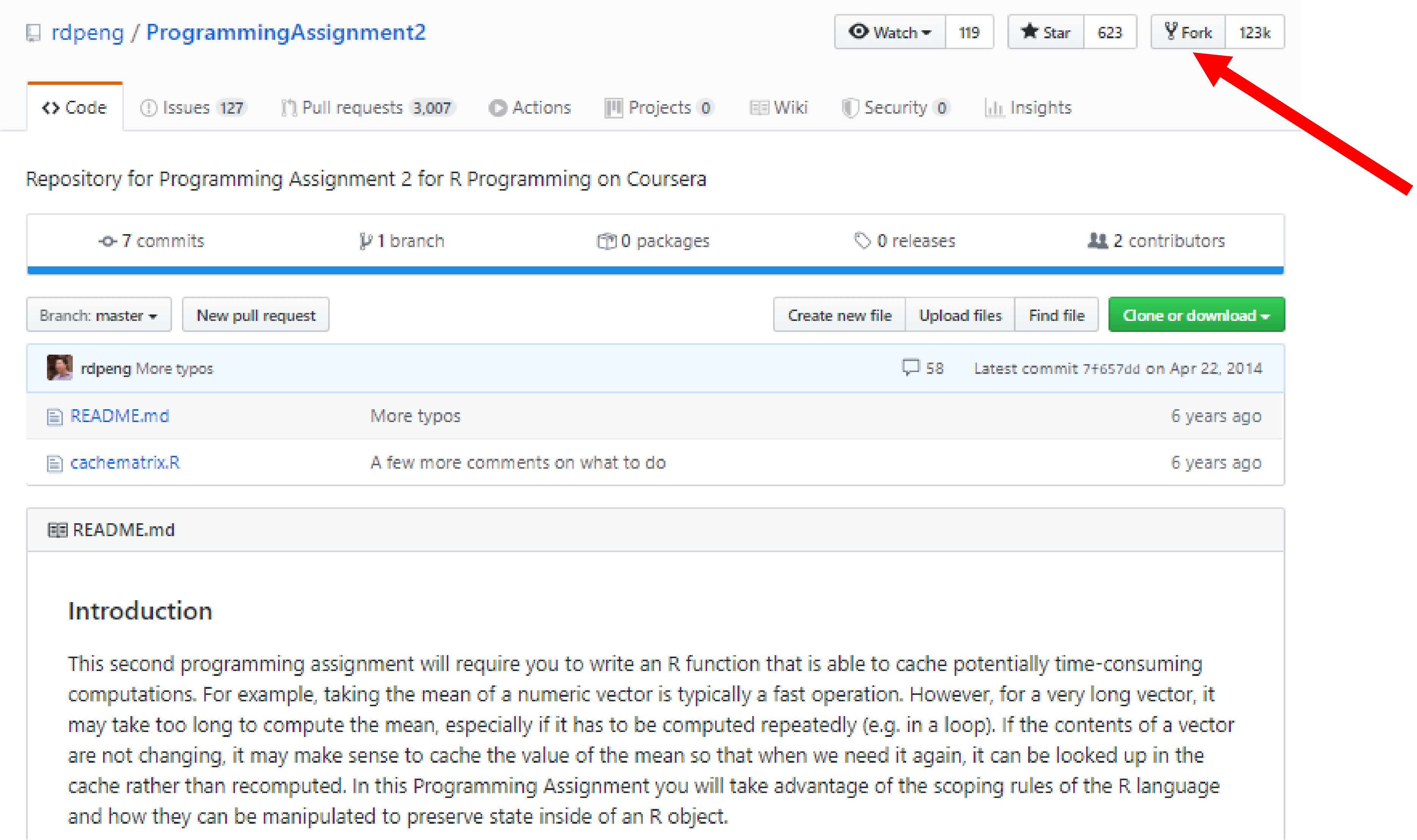
Repository for Programming Assignment 2 for R Programming on Coursera

7 commits	1 branch	0 packages	0 releases	2 contributors

rdpeng More typos	58	Latest commit 7f657dd on Apr 22, 2014
README.md	More typos	6 years ago
cachematrix.R	A few more comments on what to do	6 years ago

Now click on “Sign in” and log in to your GitHub account, then you will see (next page)

Then you will see this (you will see this directly if you were already signed into GitHub when you clicked on the link to this repository (I have scrolled down a little bit on this web page)



Repository for Programming Assignment 2 for R Programming on Coursera

rdpeng More typos — Latest commit 7f657dd on Apr 22, 2014

README.md — More typos

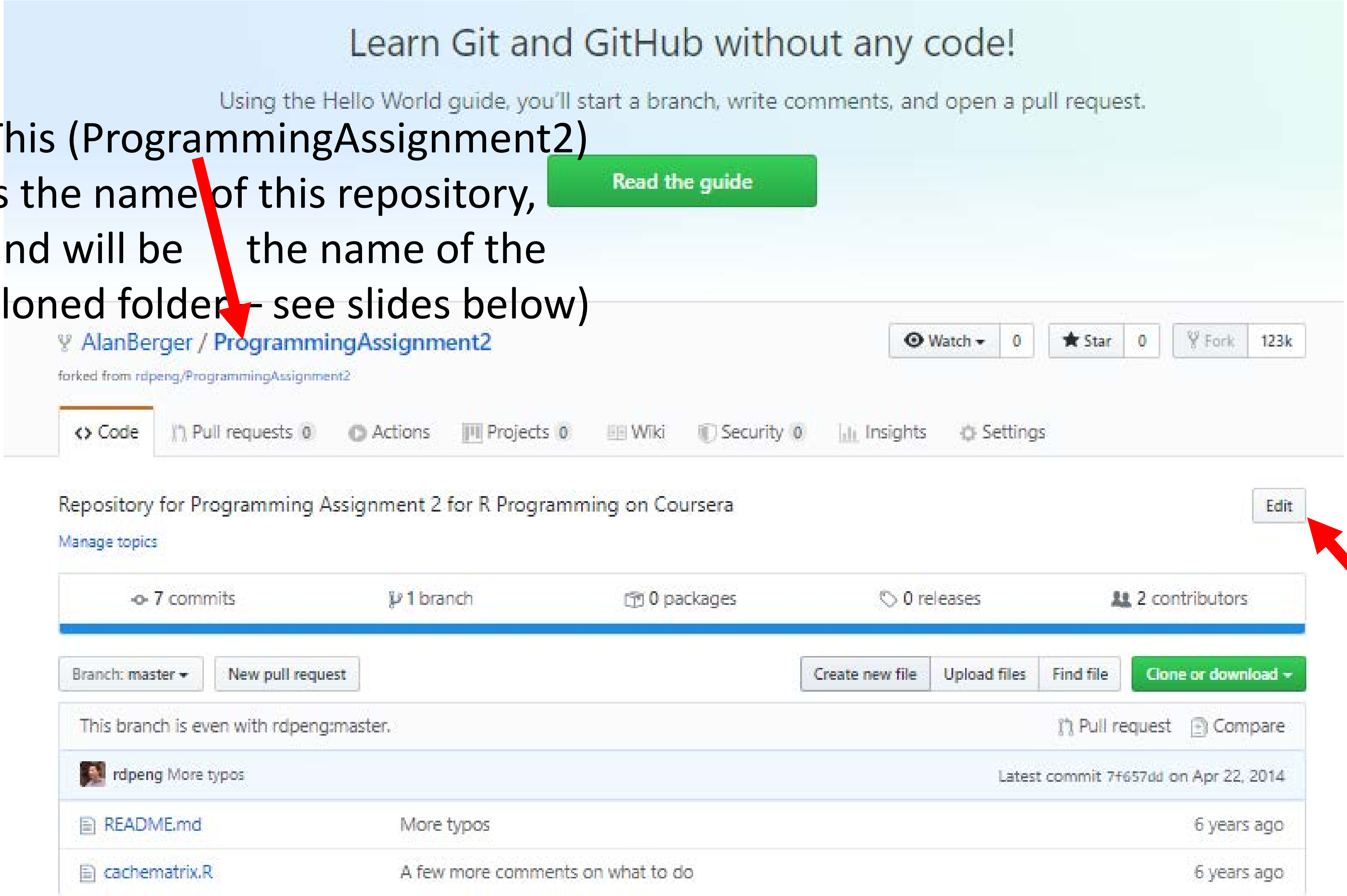
cachematrix.R — A few more comments on what to do

Introduction

This second programming assignment will require you to write an R function that is able to cache potentially time-consuming computations. For example, taking the mean of a numeric vector is typically a fast operation. However, for a very long vector, it may take too long to compute the mean, especially if it has to be computed repeatedly (e.g. in a loop). If the contents of a vector are not changing, it may make sense to cache the value of the mean so that when we need it again, it can be looked up in the cache rather than recomputed. In this Programming Assignment you will take advantage of the scoping rules of the R language and how they can be manipulated to preserve state inside of an R object.

Click on fork to get a copy of this repository into "your own space" on GitHub

After a couple seconds you should see (in your "GitHub "space") see next slide



Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

Read the guide

AlanBerger / ProgrammingAssignment2

forked from rdpeng/ProgrammingAssignment2

Watch ▾ 0 | ★ Star 0 | Fork 123k

<> Code | Pull requests 0 | Actions | Projects 0 | Wiki | Security 0 | Insights | Settings

Repository for Programming Assignment 2 for R Programming on Coursera

Edit

Manage topics

7 commits | 1 branch | 0 packages | 0 releases | 2 contributors

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

This branch is even with rdpeng:master.

Pull request | Compare

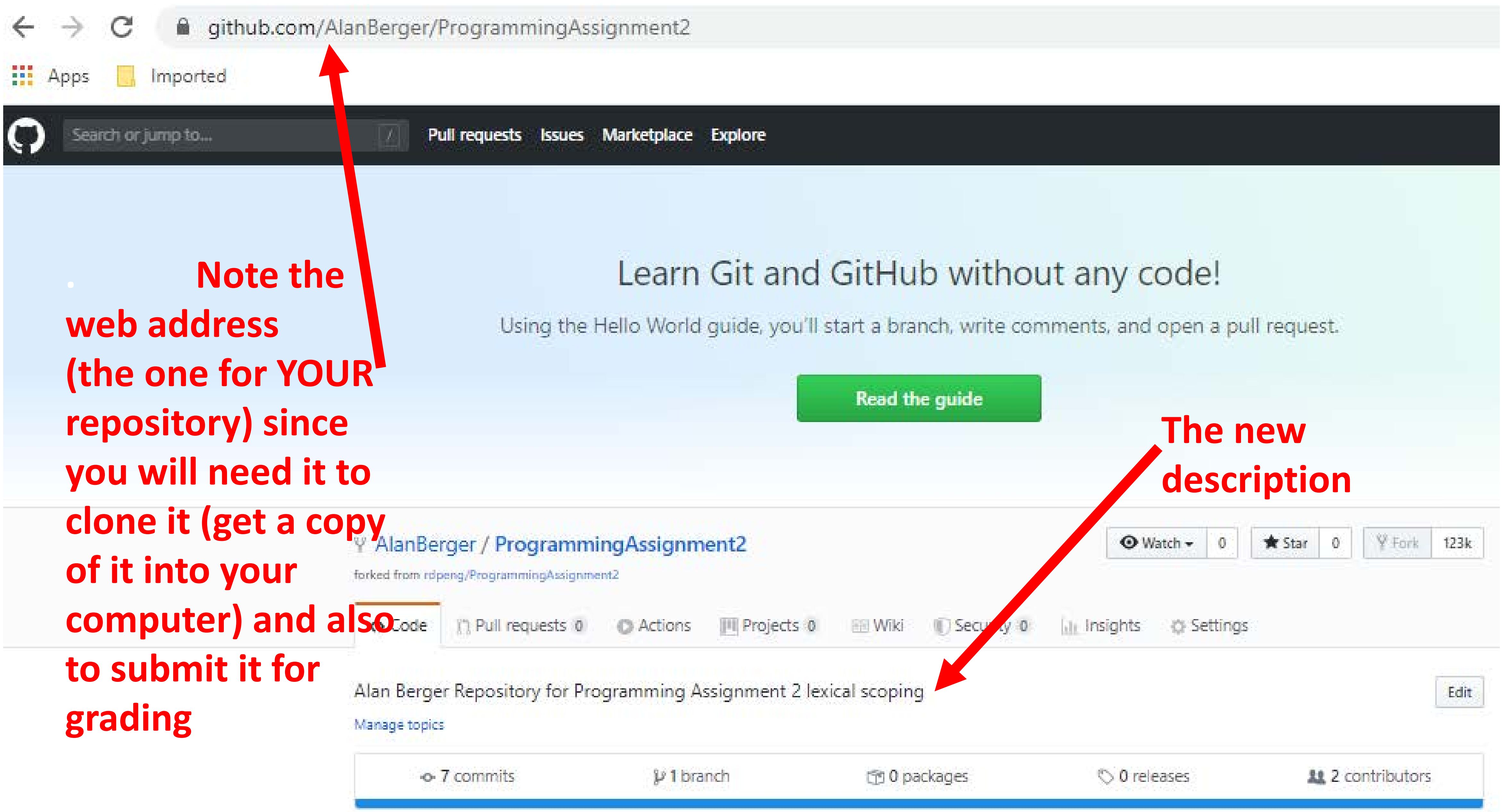
rdpeng More typos — Latest commit 7f657dd on Apr 22, 2014

README.md	More typos	6 years ago
cachematrix.R	A few more comments on what to do	6 years ago

This (ProgrammingAssignment2) is the name of this repository, and will be the name of the cloned folder – see slides below)

So this is now what you should see (but under your GitHub account). The next step is to change the description to have your name on it and an appropriate title

click on "edit" and edit the description for this repository, for example this is what I called mine (next slide)



github.com/AlanBerger/ProgrammingAssignment2

Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

AlanBerger / ProgrammingAssignment2

forked from rdpeng/ProgrammingAssignment2

Alan Berger Repository for Programming Assignment 2 lexical scoping

Note the web address (the one for YOUR repository) since you will need it to clone it (get a copy of it into your computer) and also to submit it for grading

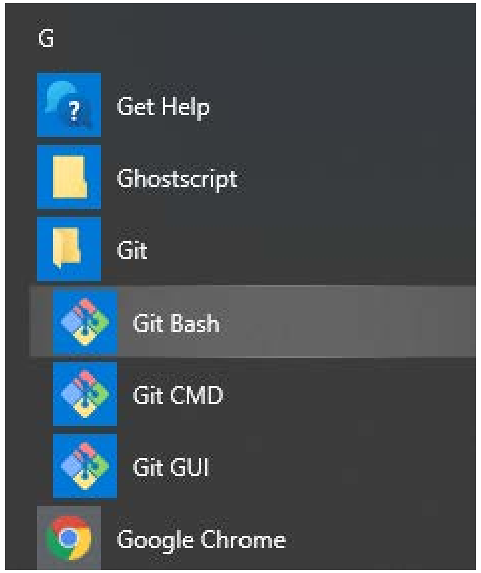
The new description

Now you want to **clone** a copy of this repository "down" into your computer where you can at your leisure edit and then check the cachematrix.R file has correct versions of the makeCacheMatrix and cacheSolve functions (along with sufficient comments to show your peer graders that you understand how these functions work – how they use lexical scoping and the super-assignment <<- operator). You do **not** need to edit the README.md file. After you edit cachematrix.R on your computer, you need to **push** the local (on your computer) version of cachematrix.R back out onto your GitHub repository for this assignment.

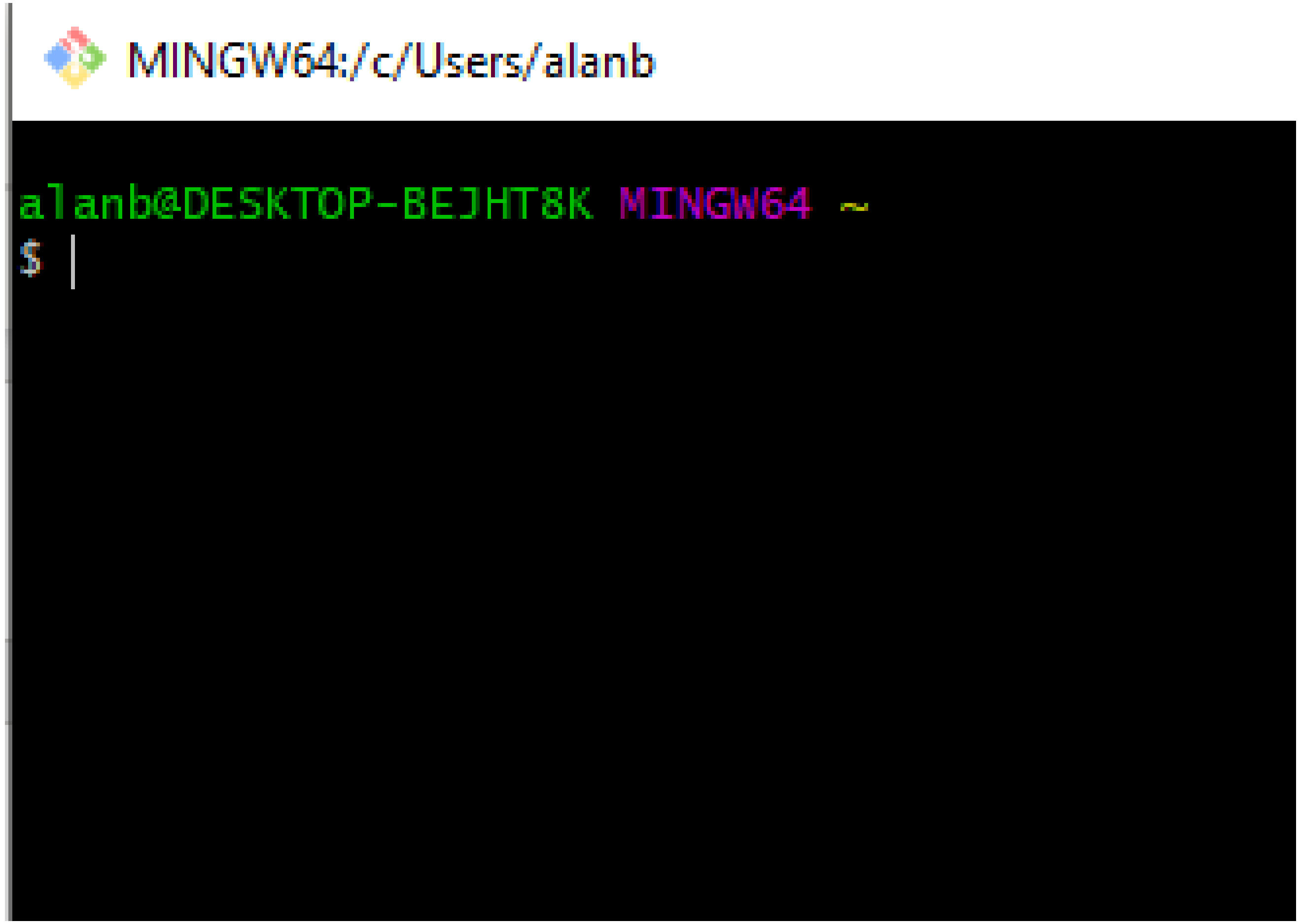
In the following slides I'll show how to do this using **Git**, but, sorry, the cachematrix.R file I'm putting out on my repository only shows how to check your makeCacheMatrix and cacheSolve functions using some simple test matrices (which is the material in this post "Simple test matrices for the lexical scoping programming assignment").

The next step is to use Git to **clone** a copy of your GitHub repository for the assignment down into your computer. This will be a folder on your computer that has the files from the repository, and also version control information that allows Git to interact with the GitHub repository.

Open the Git Bash interface (on my Windows computer, under Start, this is part of the display (I also have a shortcut to Git Bash on the Desktop)).



You will then see something that looks like this (**next slide**)



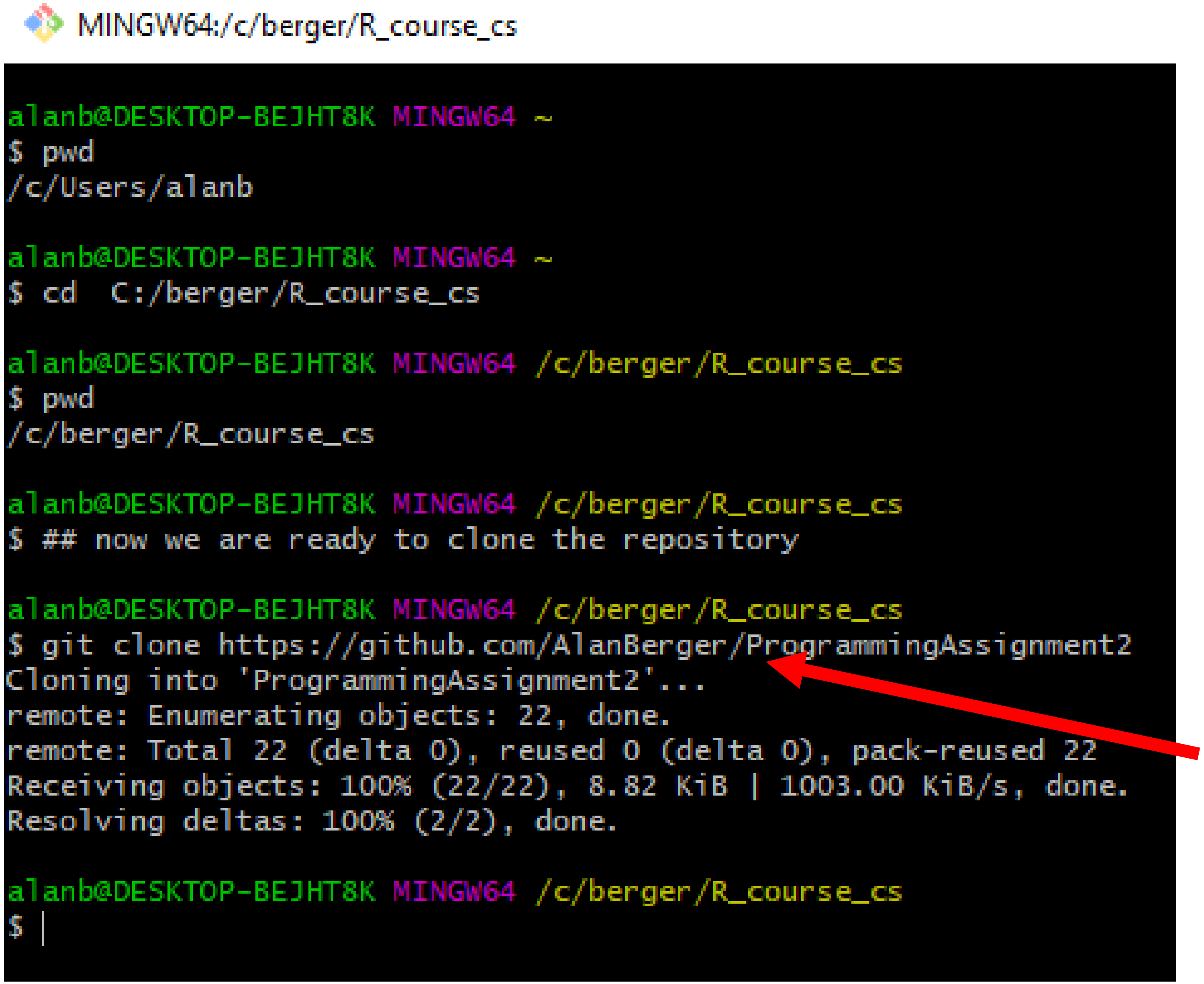
MINGW64:/c/Users/alanb

```
alanb@DESKTOP-BEJHT8K MINGW64 ~
$ 
```

In the Git Bash window, what we want to do is get ourselves to the folder / directory **in which** we want the cloned repository. In my computer I want it in **C:/berger/R_course_cs** Note the Unix style forward slashes in this directory name.

Note the **$** is the Git Bash command prompt

The Git Bash command **pwd** is **p**rint the current **w**orking **d**irectory and **cd** is **c**hange **d**irectory. To paste into the Git Bash command line text you have copied, do a right click on the mouse then choose paste (in Windows, control v does not work) (**next slide**)



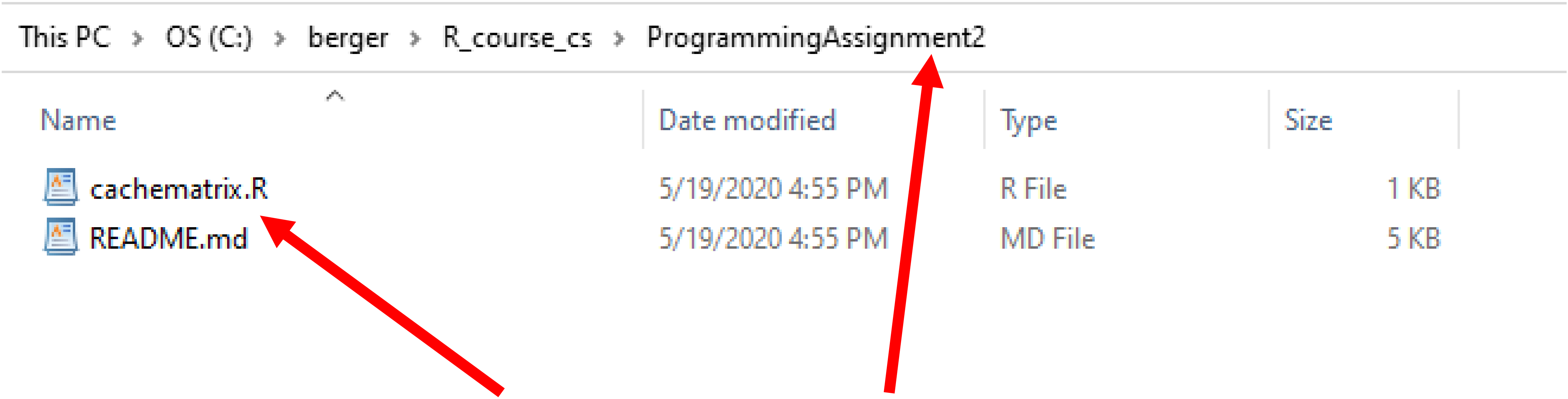
```
alanb@DESKTOP-BEJHT8K MINGW64 ~
$ pwd
/c/Users/alanb

alanb@DESKTOP-BEJHT8K MINGW64 ~
$ cd  C:/berger/R_course_cs

alanb@DESKTOP-BEJHT8K MINGW64 /c/berger/R_course_cs
$ pwd
/c/berger/R_course_cs

alanb@DESKTOP-BEJHT8K MINGW64 /c/berger/R_course_cs
$ ## now we are ready to clone the repository

alanb@DESKTOP-BEJHT8K MINGW64 /c/berger/R_course_cs
$ git clone https://github.com/AlanBerger/ProgrammingAssignment2
Cloning into 'ProgrammingAssignment2'...
remote: Enumerating objects: 22, done.
remote: Total 22 (delta 0), reused 0 (delta 0), pack-reused 22
Receiving objects: 100% (22/22), 8.82 KiB | 1003.00 KiB/s, done.
Resolving deltas: 100% (2/2), done.

alanb@DESKTOP-BEJHT8K MINGW64 /c/berger/R_course_cs
$ |
```

So I “looked to see where I was” with **pwd**, and then did **cd** to the directory where I wanted the cloned repository. These were Unix style commands which Git Bash provides. Then I did a git command (git followed by a command, which looks like $ git <a git command> in the displays here (**$** is the command prompt , not part of the command):

git clone <web address of repository>

to **clone** the specified repository to a folder in the current working directory in Git. Git “says” it did it (**next slide**)

So, is it there (on my computer)?



This PC › OS (C:) › berger › R_course_cs › ProgrammingAssignment2

Name	Date modified	Type	Size
cachematrix.R	5/19/2020 4:55 PM	R File	1 KB
README.md	5/19/2020 4:55 PM	MD File	5 KB

Indeed it is! with the two files, and the folder name is the GitHub repository name

So we can edit cachematrix.R and then **push** it back onto our GitHub repository (details below).

Some technical notes: The Git Bash command window allows some Unix style commands such as pwd and cd, which look like

```
$ pwd
$ cd desired.directory.name
```

in the Git Bash display (the $ is the command prompt)

Git Bash “contains” **git commands** for interacting with GitHub (doing file management and version control). They look like, for example

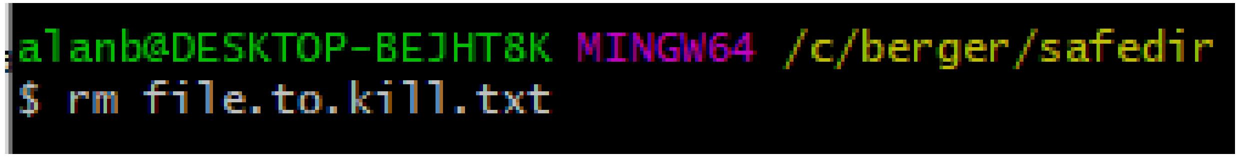
```
$ git <followed by a git command>
```

For example

```
$ git clone <web address>
```

We’ll see some more shortly.

Warning: Git Bash has a Unix **rm** command. Unlike rm in R which only removes objects in your current R session, rm in Git Bash will **permanently remove** files from your computer’s disk. If in Git Bash you do



```
alanb@DESKTOP-BEJHT8K MINGW64 /c/berger/safedir
$ rm file.to.kill.txt
```

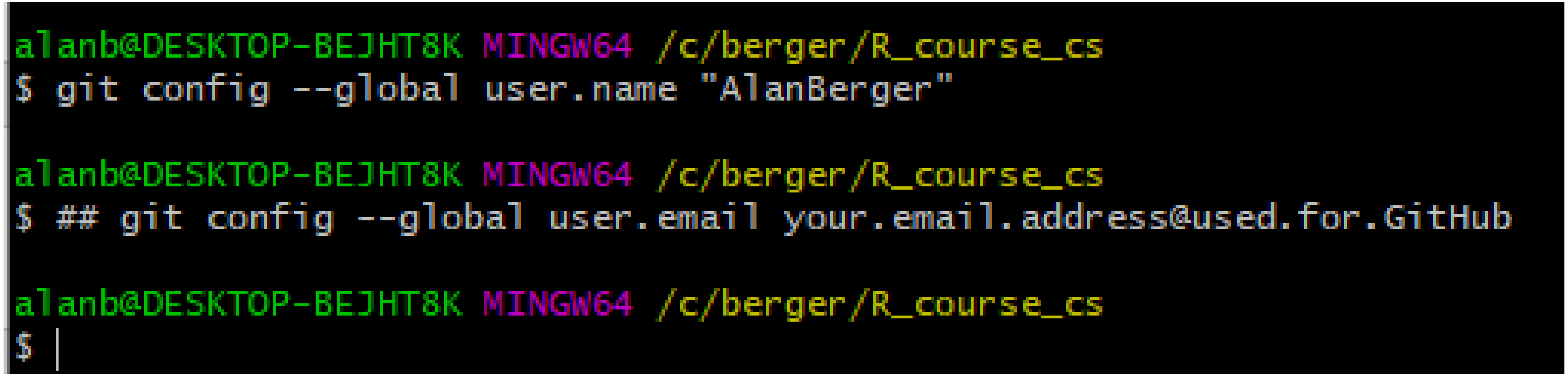
Then file.to.kill.txt is

G O N E

(there is no trash bin in Unix from which to recover mistakenly deleted files)

Before doing any more with git, you should “tell” Git your name (use the name for your GitHub account: my main GitHub page is: https://github.com/AlanBerger
So I’ll “give” Git my name as AlanBerger

And you should “tell” Git your email address using the one you used for GitHub, as in these lines in Git Bash below



```
alanb@DESKTOP-BEJHT8K MINGW64 /c/berger/R_course_cs
$ git config --global user.name "AlanBerger"

alanb@DESKTOP-BEJHT8K MINGW64 /c/berger/R_course_cs
$ ## git config --global user.email your.email.address@used.for.GitHub

alanb@DESKTOP-BEJHT8K MINGW64 /c/berger/R_course_cs
$ |
```

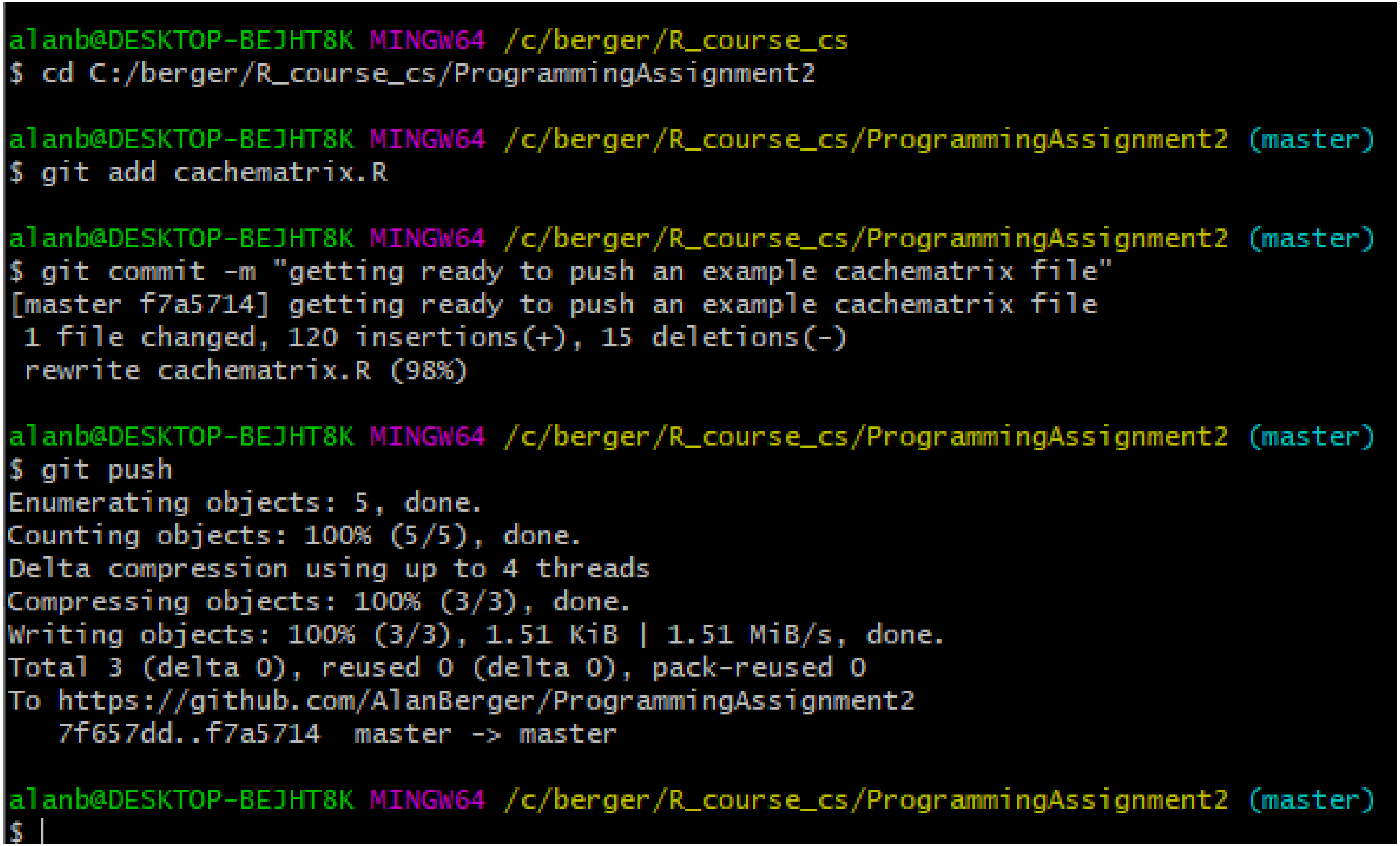
To check you typed in what you wanted, you can do: **git config --list**
If you want to change your name or email for git just do the corresponding command again.

I’ve now edited cachematrix.R so the top part is now

```
## some instructions for testing your makeCacheMatrix and cacheSolve functions

This is from this post: "Simple test matrices for the lexical scoping programming assignment"

https://www.coursera.org/learn/r-programming/discussions/weeks/3/threads/ePlO1eMdEeahzg7_4P4Vvg

R session:

=============================================
```

and so I’m ready to get it back out into GitHub. So how do we use git to **push** the modified cachematrix.R back out onto the GitHub repository from which we cloned the copy on our computer.

Open the Git Bash command window; then in the Git Bash window, cd (change directory) to the directory of your local Repository where your cachematrix.R file is; then issue the commands on **the next slide** (with commentary on the following slide – note when you do the “git push”, GitHub will ask for your user name and password (unless you have set up this information in git (which I’m not covering here)).



```
alanb@DESKTOP-BEJHT8K MINGW64 /c/berger/R_course_cs
$ cd C:/berger/R_course_cs/ProgrammingAssignment2

alanb@DESKTOP-BEJHT8K MINGW64 /c/berger/R_course_cs/ProgrammingAssignment2 (master)
$ git add cachematrix.R

alanb@DESKTOP-BEJHT8K MINGW64 /c/berger/R_course_cs/ProgrammingAssignment2 (master)
$ git commit -m "getting ready to push an example cachematrix file"
[master f7a5714] getting ready to push an example cachematrix file
 1 file changed, 120 insertions(+), 15 deletions(-)
 rewrite cachematrix.R (98%)

alanb@DESKTOP-BEJHT8K MINGW64 /c/berger/R_course_cs/ProgrammingAssignment2 (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.51 KiB | 1.51 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/AlanBerger/ProgrammingAssignment2
   7f657dd..f7a5714  master -> master

alanb@DESKTOP-BEJHT8K MINGW64 /c/berger/R_course_cs/ProgrammingAssignment2 (master)
$ 
```

To get your completed assignment version of cachematrix.R back out onto GitHub, open the Git Bash command window; then in the Git Bash window, cd (change directory) to the directory of your local Repository where your cachematrix.R file is; then issue the command: ($ is the Git Bash command prompt)

```
$ git add filename
```

where filename is the name of the file you want to put out on GitHub (here cachematrix.R) (in general, if more than one file, give their names separated by a space, do NOT put quotes around the filename(s)).
There was no "response" from Git Bash after this command other than to issue a new command prompt
Then

```
$ git commit -m "getting ready to push an example cachematrix file"
```

does the commit (here do enclose your title in quotes), and Git Bash returned (when I did this)

```
[master f7a5714] getting ready to push an example cachematrix file
 1 file changed, 120 insertions(+), 15 deletions(-)
 rewrite cachematrix.R (98%)
```

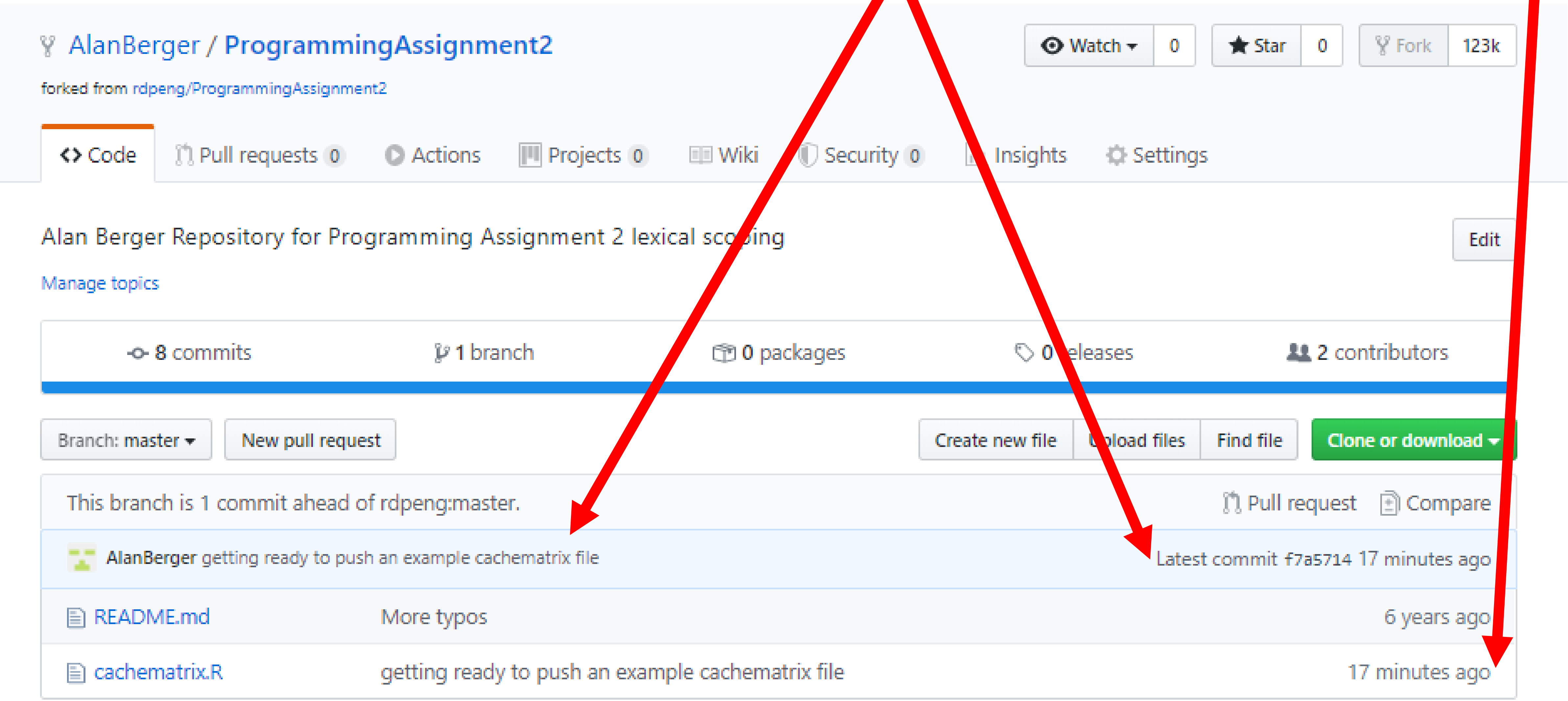
Then

```
$ git push
```

"magically" did the push into my corresponding GitHub Repository of my cachematrix.R file along with returning a technical description of its action (In a separate window GitHub asked for my GitHub user name and password, before the push was completed)

The modified version of my GitHub repository now looked like (**next slide**)

The updated repository with new commit and updated cachematrix.R file



AlanBerger / ProgrammingAssignment2

forked from rdpeng/ProgrammingAssignment2

Alan Berger Repository for Programming Assignment 2 lexical scoping

This branch is 1 commit ahead of rdpeng:master.

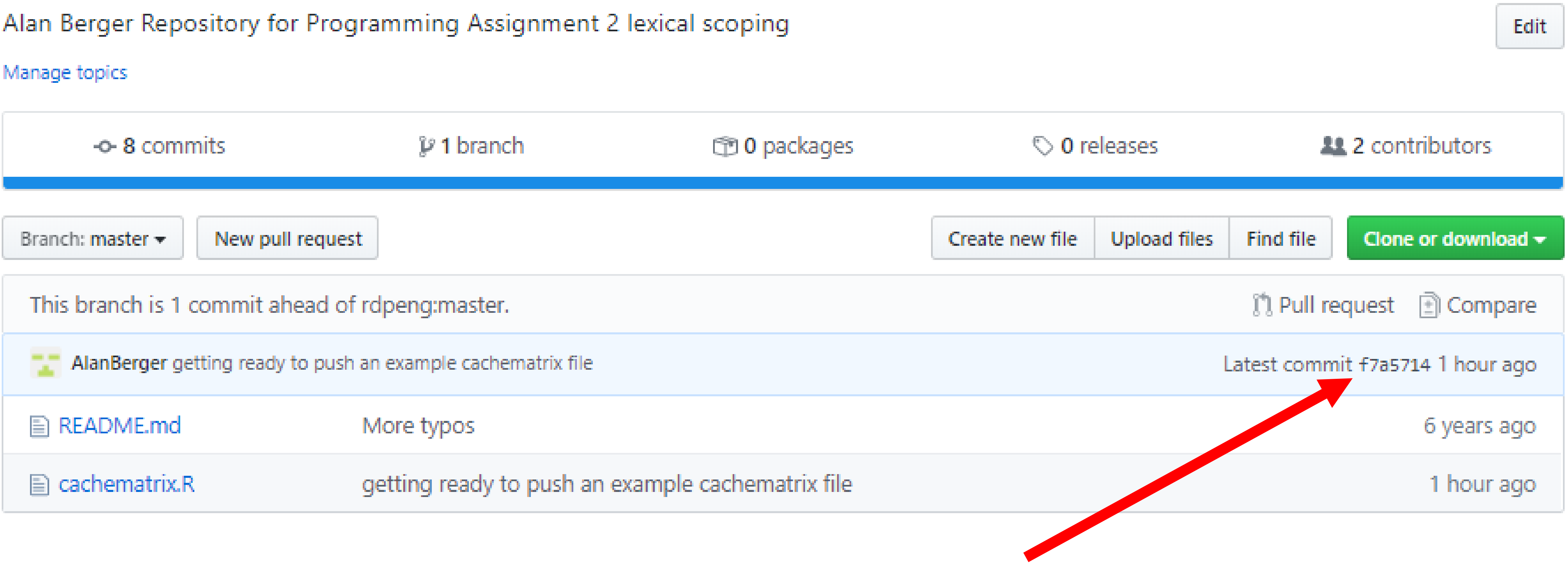
AlanBerger getting ready to push an example cachematrix file — Latest commit f7a5714 17 minutes ago

README.md	More typos	6 years ago
cachematrix.R	getting ready to push an example cachematrix file	17 minutes ago

Note you only need to submit your cachematrix.R file, not a new README.md file; your grade is on the correctness of your functions and your added comments showing you understand how the functions and Lexical Scoping and the super-assignment operator <<- work (and also on submitting the GitHub repository web address (URL) of your submission, for example mine is:
https://github.com/AlanBerger/ProgrammingAssignment2
and also the text string of its 40 character SHA-1 hash code – see the next slide for how to get the full SHA-1 code). If you need more help on lexical scoping in this context be sure to read Len Greski's Demystifying makeVector article if you haven't already done so.

How to get the 40 character SHA-1 hash code of a commit on GitHub:

This is the relevant part of my GitHUB repository for the lexical scoping assignment



Alan Berger Repository for Programming Assignment 2 lexical scoping

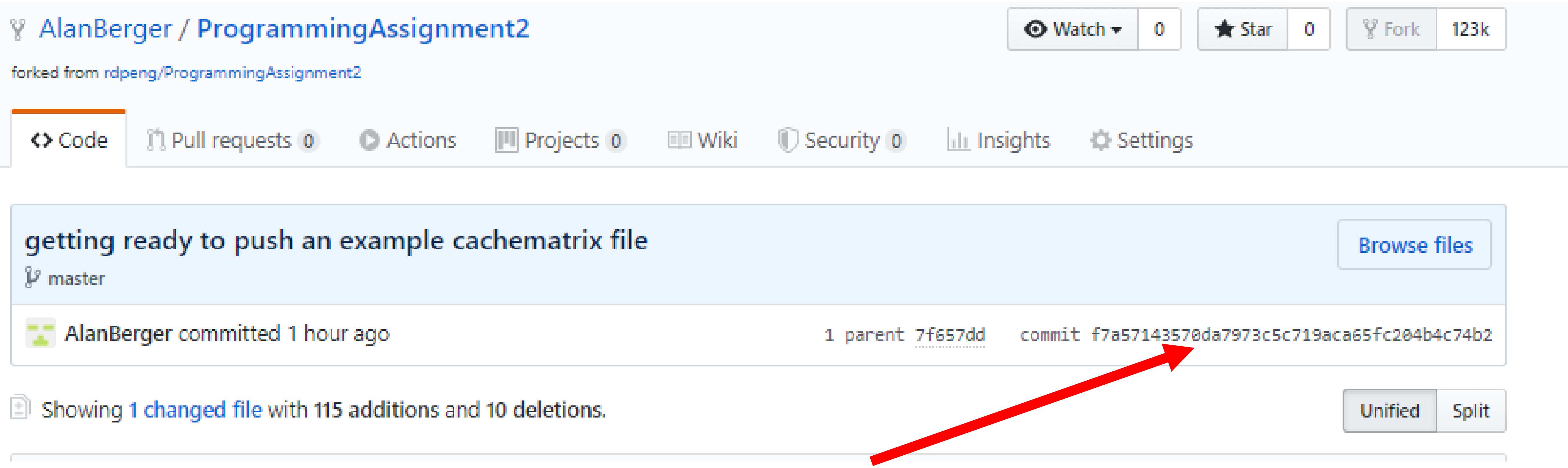
8 commits | 1 branch | 0 packages | 0 releases | 2 contributors

This branch is 1 commit ahead of rdpeng:master.

AlanBerger getting ready to push an example cachematrix file Latest commit f7a5714 1 hour ago

README.md	More typos	6 years ago
cachematrix.R	getting ready to push an example cachematrix file	1 hour ago

This is the first few characters of the 40 character SHA-1 hash code – click on it to display and copy all of it to a text file so you can submit it for the assignment



AlanBerger / ProgrammingAssignment2

forked from rdpeng/ProgrammingAssignment2

getting ready to push an example cachematrix file

master

AlanBerger committed 1 hour ago 1 parent 7f657dd commit f7a57143570da7973c5c719aca65fc204b4c74b2

Showing 1 changed file with 115 additions and 10 deletions.

After clicking on the first few characters of the SHA-1 hash code this window appeared with the full hash code – copy it and paste as text someplace where you can use it when you submit this assignment

I hope this “tutorial”, though lengthy, will help you complete the assignment as well as be a start for learning GitHub and Git,

Alan Berger 19 May 2020