1.6.1

Using GitHub

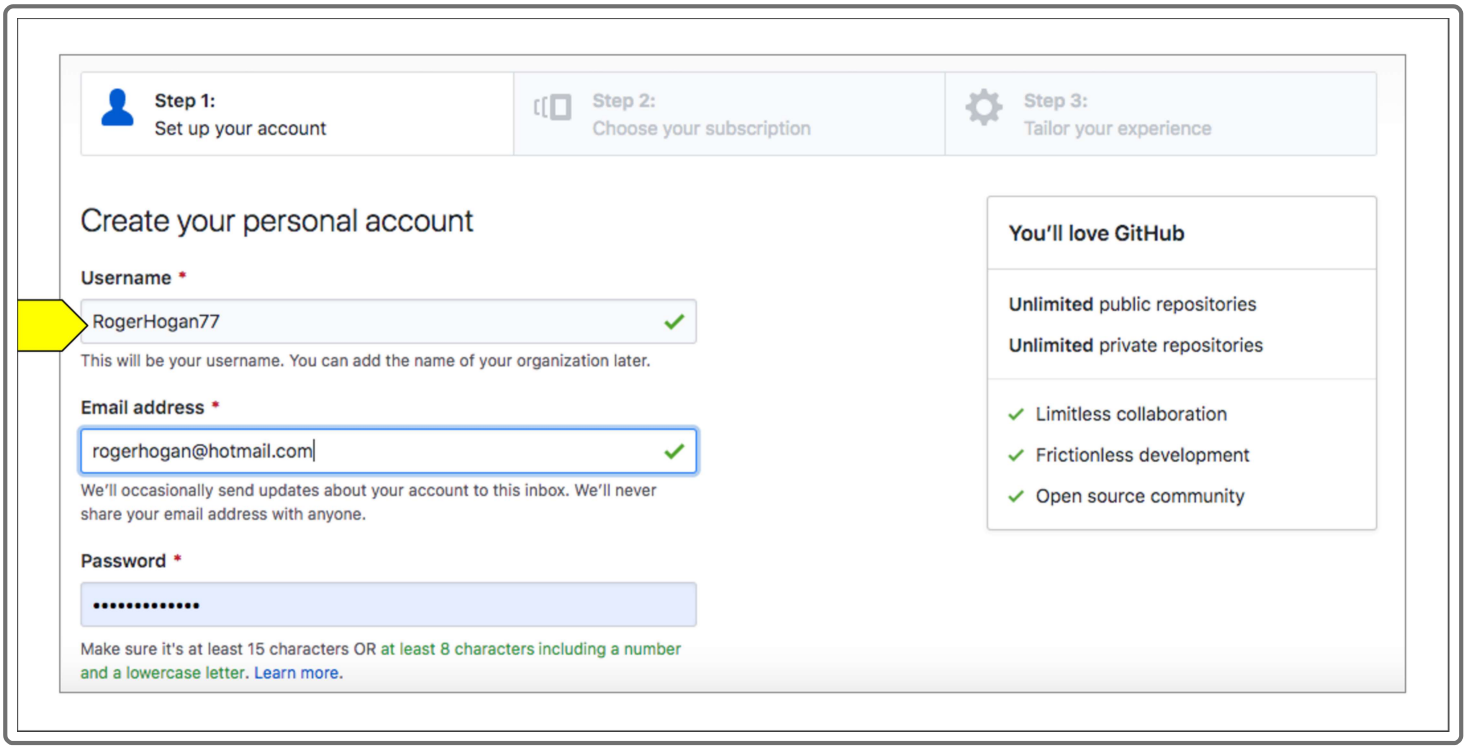
Now that we've completed our analysis, we need to create a report to showcase our findings. We also need a place to host our report and images, so we'll create an account on GitHub and set it to share our report.

We've gleaned a lot of helpful information and insights from this data. We've organized, sorted, filtered to narrow down our scope, and then we created visualizations and performed statistical computations. Now it's time to put it all together and share our findings with Louise. For this step we'll be using GitHub.

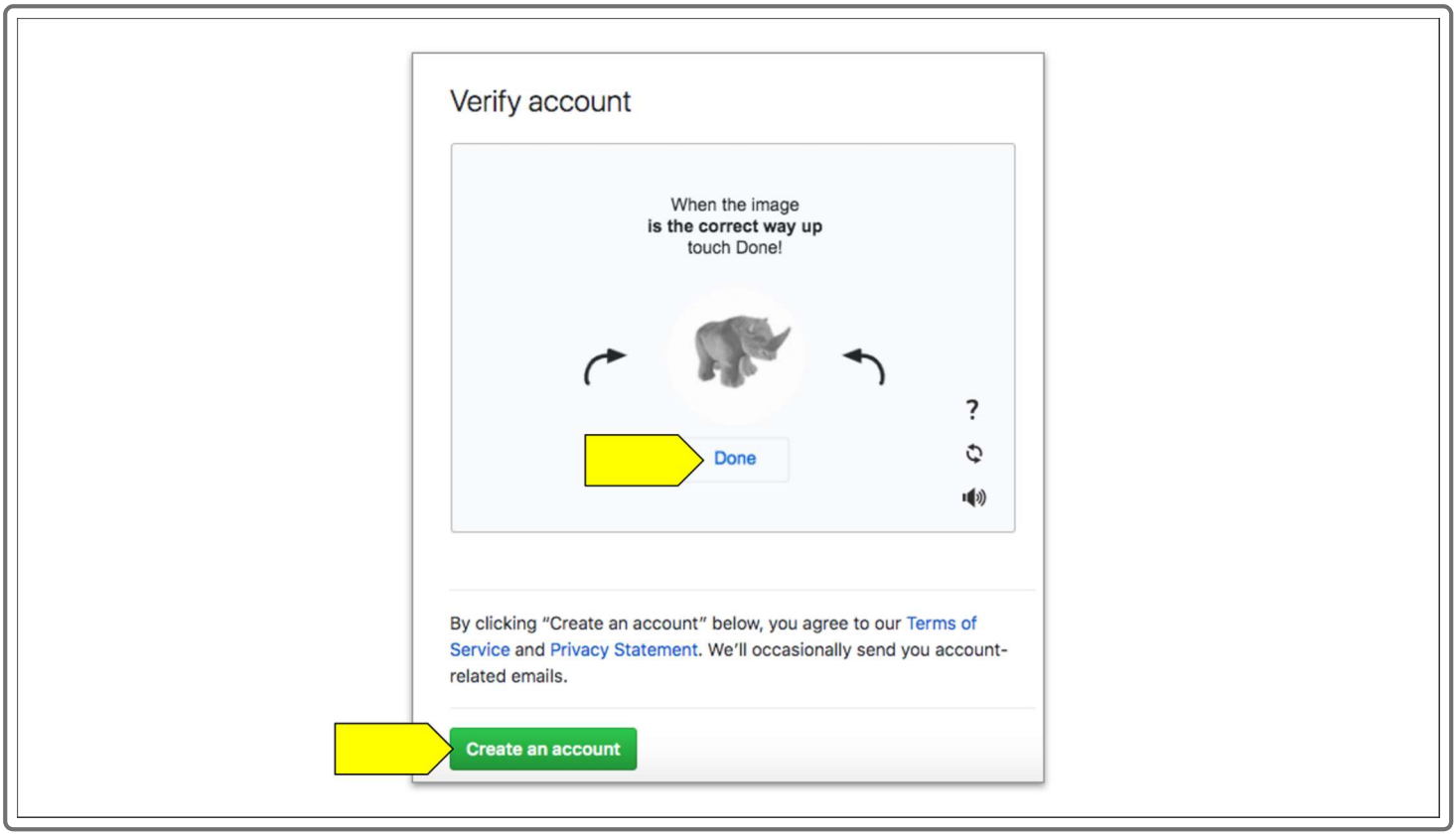
Create a GitHub Account

GitHub is an online hosting service for storing files, images, and other content online. GitHub is quick, convenient, and widely used by developers of all types.

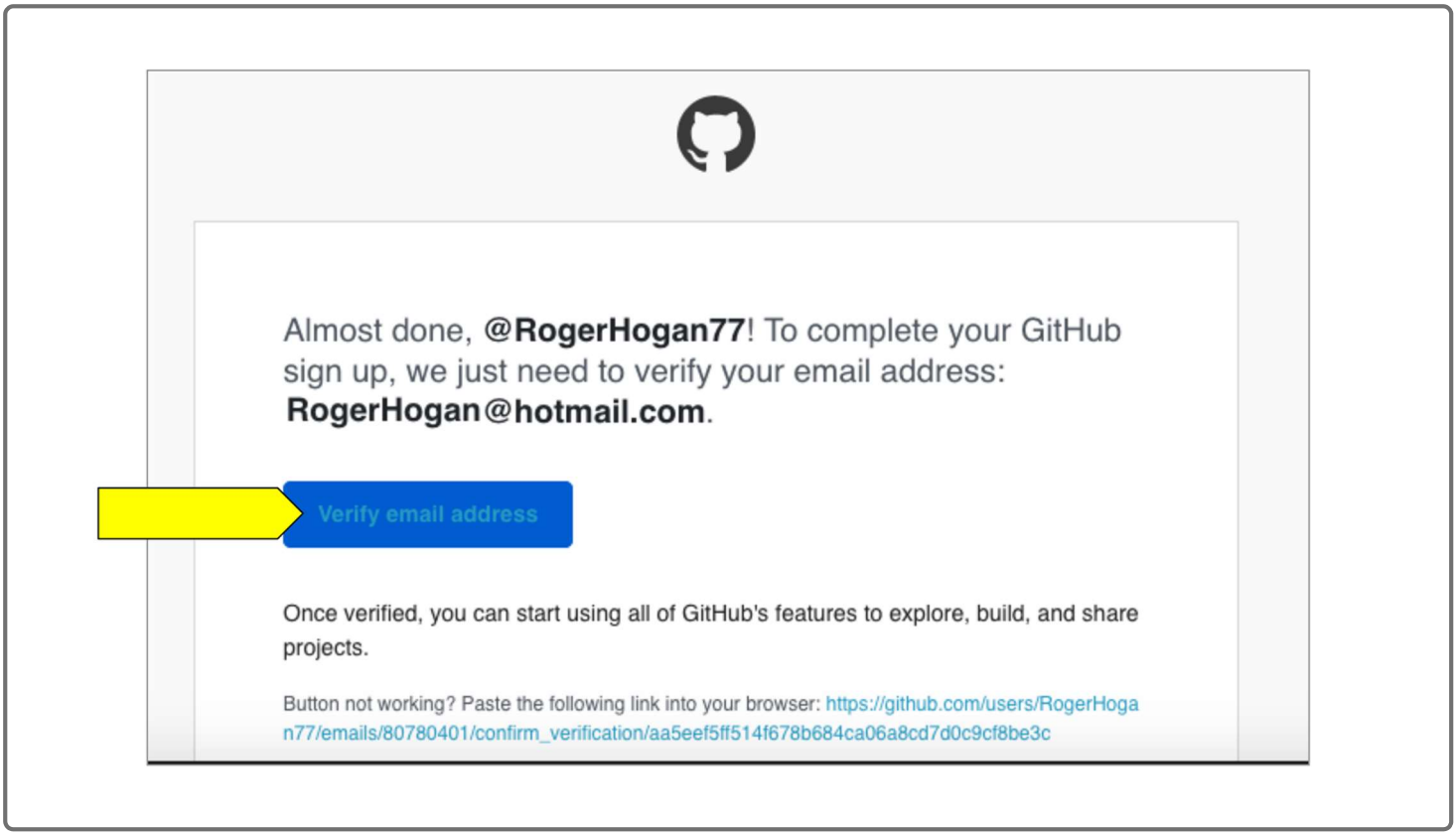
To get started, go to the **GitHub website (https://github.com/)** and create an account, or log in if you already have one. Note that your username is public and will be tied to each repository you create.



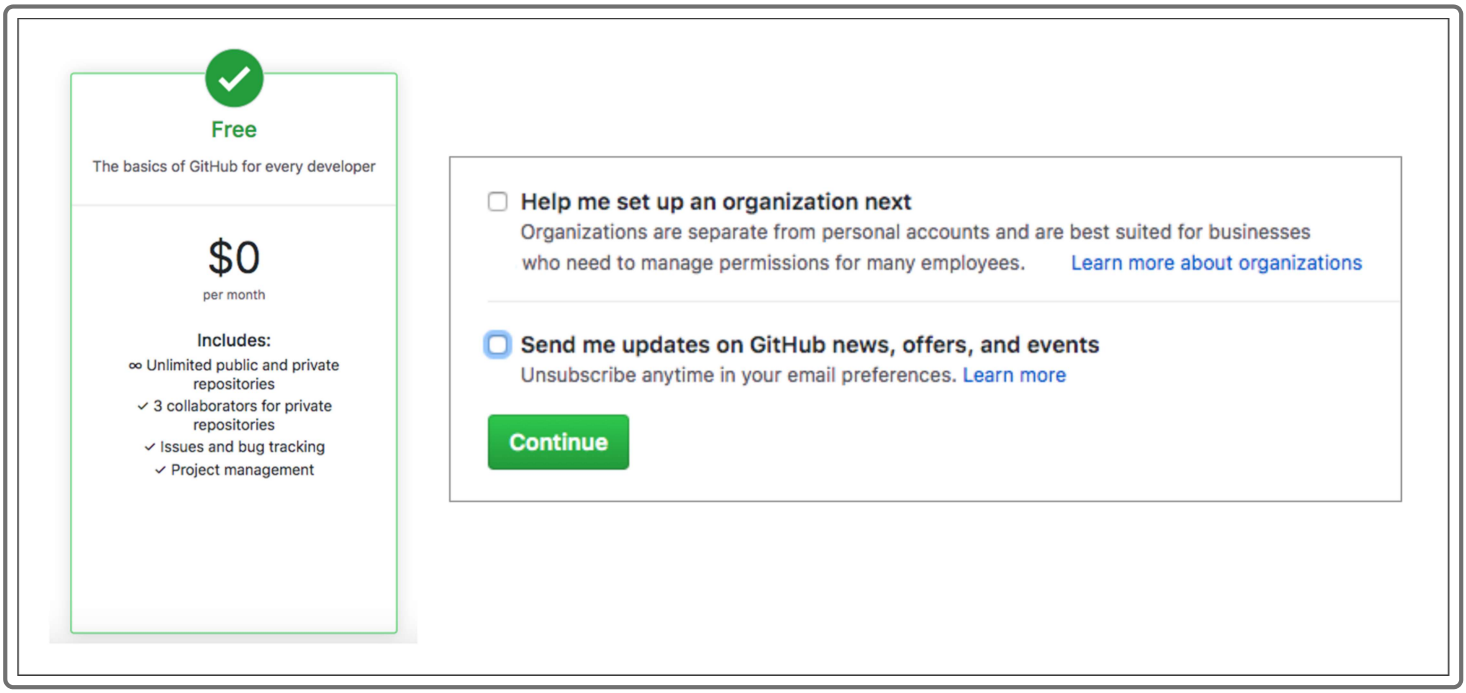
After you enter your username, email, and password, you will be sent to a page to verify your account. This is where GitHub will confirm that you aren't a bot (an automated program). This verification step will likely entail some kind of short puzzle, such as using arrows to rotate an image.



You'll also receive an email from GitHub to verify your email address. Don't skip this step!

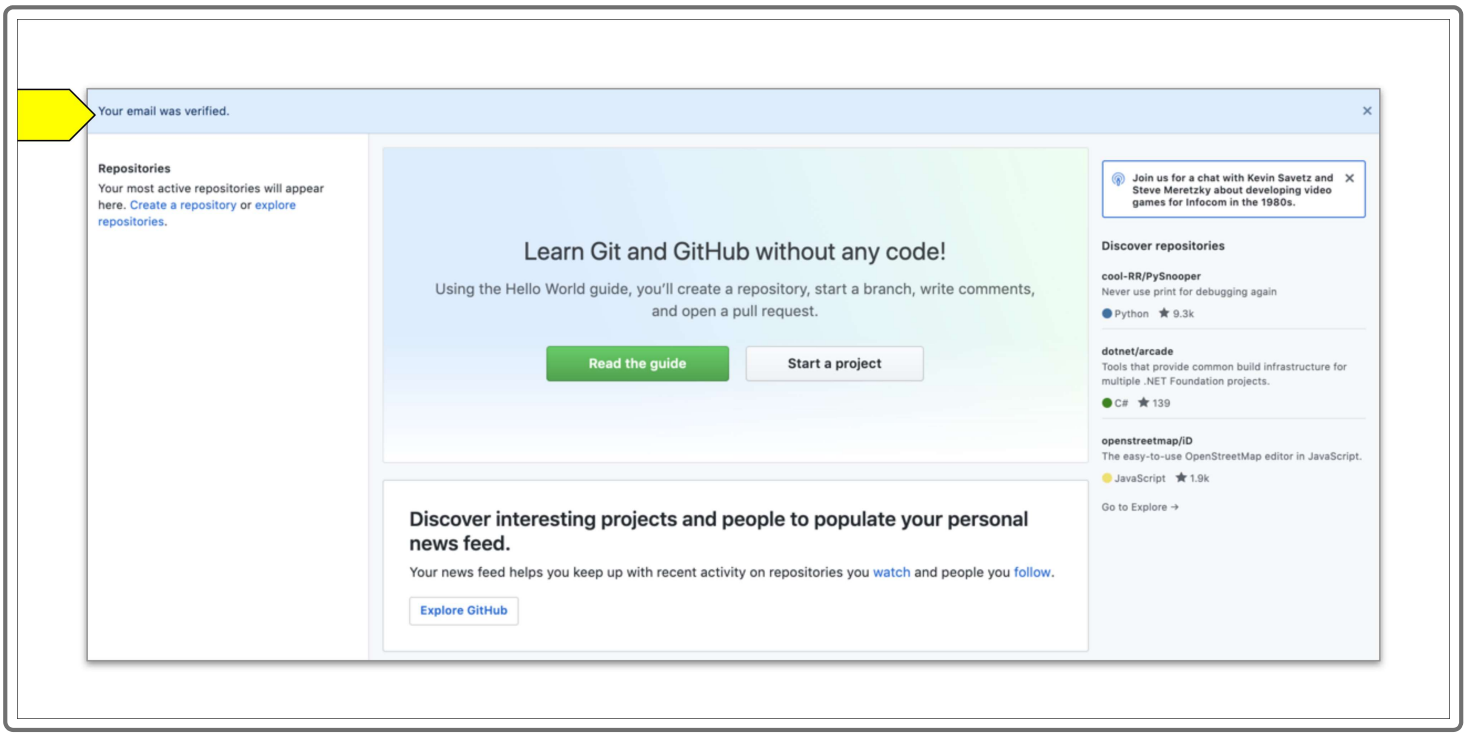


After completing the verification, you'll select a subscription type. Be sure to select the free subscription option that is outlined in green. Scroll down to uncheck the box allowing your email address to be added to GitHub's mailing list (unless you want to receive emails from GitHub). Then click Continue.



The next page you're directed to is a survey to help tailor your GitHub experience. It isn't necessary to complete this form, and you can click "Skip This Step" to finish creating your account.

If you haven't already verified your email address, take the time to complete this step now. Navigate to your email inbox, open the message from GitHub, and click the provided link to verify your account. This will open a new window that will automatically open in your web browser:



Now you're ready to create your first repository!

IMPORTANT

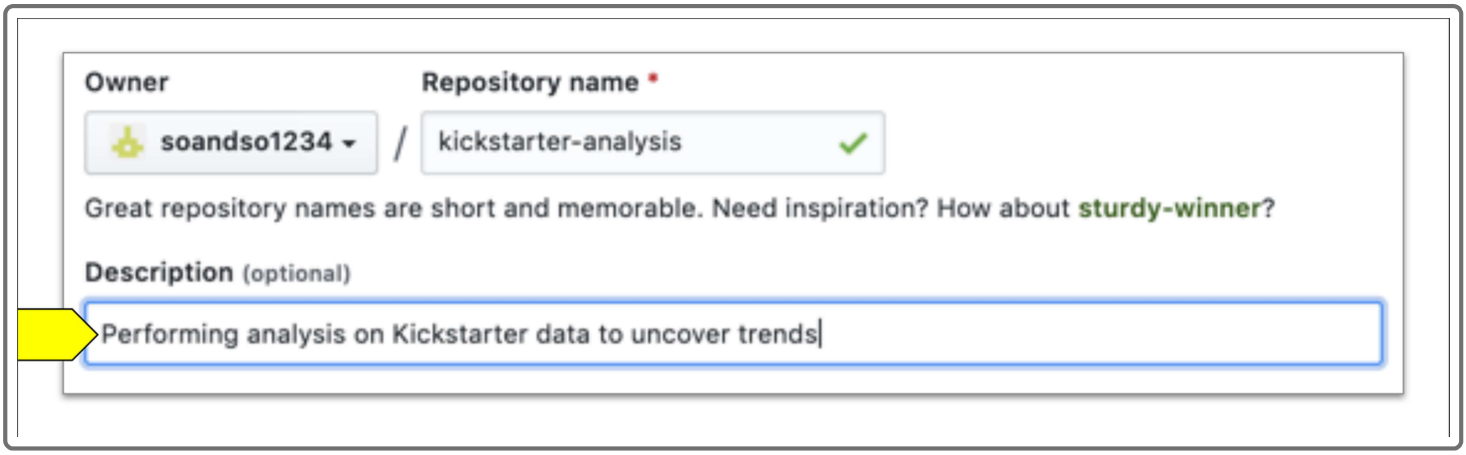
A **repository**, or repo for short, is the project-specific holding space for the files used in a project. Another great thing about GitHub is that there are *unlimited* repositories per user! Many companies use GitHub because it is a perfect collaboration hub for development teams.

Create a Repository

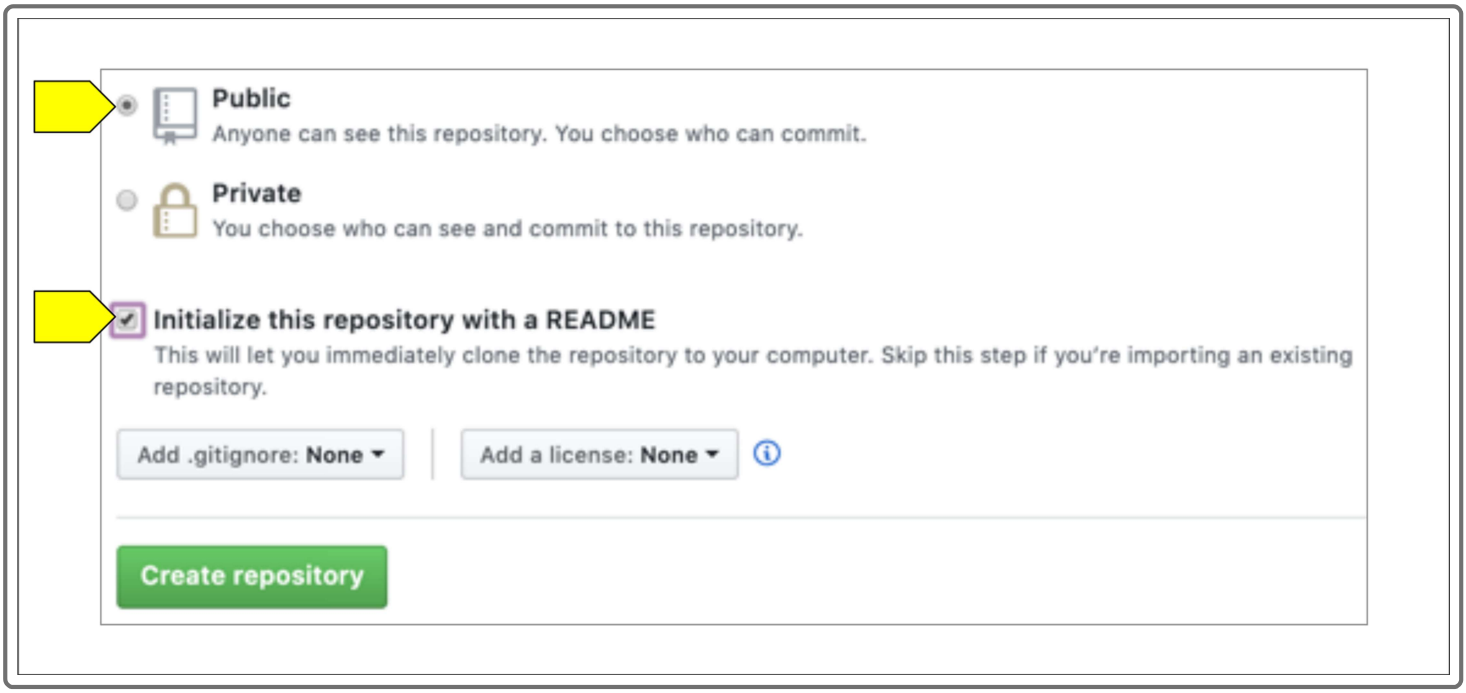
Now that we're logged in, we'll create a new repository. Click the "Start a Project" button to go to the "Create a Repository" page, which is where we'll begin to set up our project.

Each repo should have a unique name. You can use GitHub's random repo name generator, but it's better to use a name that describes the project. Let's name our repo kickstarter-analysis to get started.

Enter a brief description of the project. Keep it short and succinct; the point of the description is to help others understand the purpose of the repo. A project description will also help jog your memory when you revisit old repositories weeks or months down the line. See the following example:



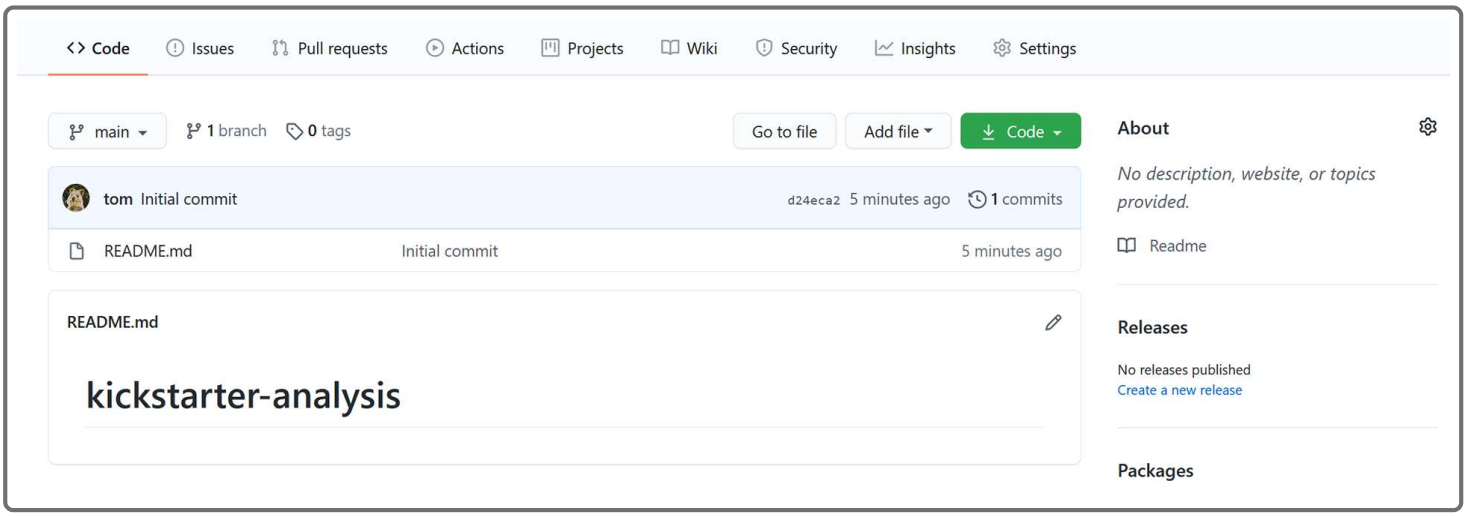
Once you've given your repo a name and added a description, check the box that says "Initialize this repository with a README." Keep the default setting to make the repo public.



NOTE

A README file contains an explanation or description of the project so that others know how to use the information provided. These files are typically written in markdown format and will be automatically displayed when someone visits your repository. We'll cover markdown in greater detail later.

Click the green Create Repository button to make the new repo official. You will then be automatically redirected you to a new (and somewhat empty) page.



Upload Files

Now we'll upload our Excel spreadsheet with the completed analysis. Click the Add File button by the green Code button. On the dropdown list that will appear, click Upload Files. You can either drag and drop the appropriate .xls file or click "Choose Your Files " to manually navigate through your directory to locate the correct file.

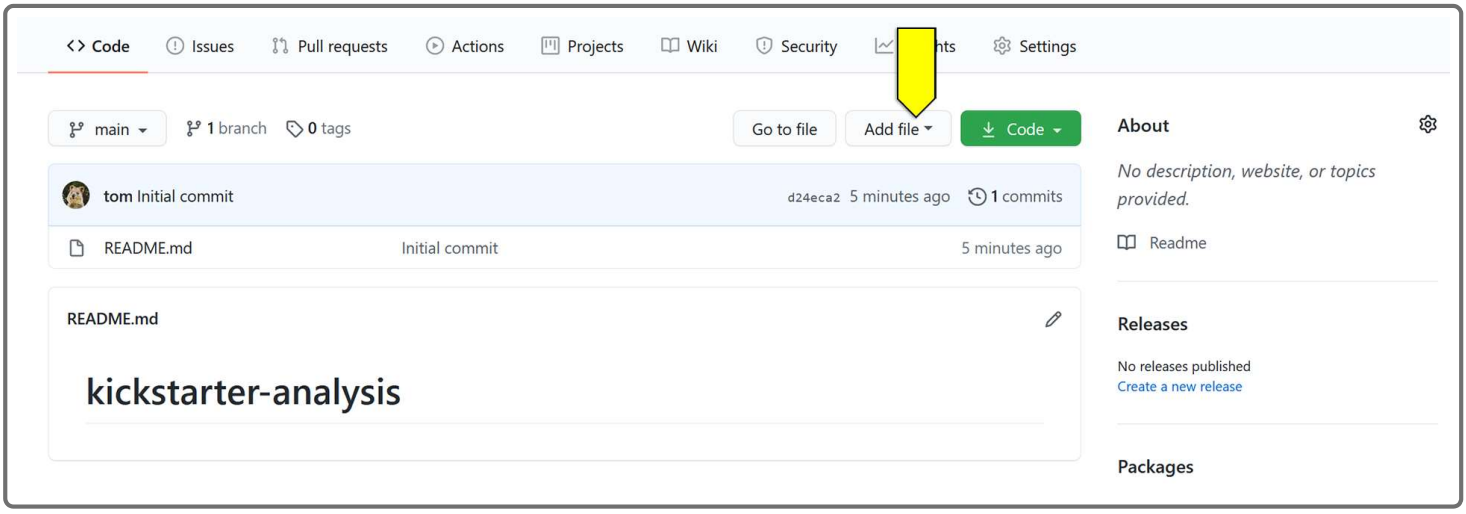
If your file size is more than 25 MB, then you'll run into an error when you try to upload your work. You'll need to compress your file first (also known as zipping it), and then upload the zipped file.

For instructions on how to compress your Excel workbook file, visit one of the following links:

macOS (https://www.imore.com/how-compress-file-your-mac)

Windows (https://support.microsoft.com/en-us/help/14200/windows-compress-uncompress-zip-files)

Later, you'll learn more about how to upload large files without zipping them.



Note that multiple files can be uploaded at once, which will save some time as we continue to prepare our analysis report.

Next, we'll "commit" our changes. A **commit** is part of the GitHub process that captures why a change is made. Any action---from adding a file, updating it, or deleting a part of it---is reflected through commit messages. This feature is similar to saving our work in Excel, and it's easy to revert back to or reference a previous version if needed.

A commit message should be descriptive but to the point. For example, when uploading our data, the message could read, "Uploading completed Kickstarter analysis." This way, we aren't inundated with text if we check our commits; a quick glance will give us the pertinent information.

Make sure the "Commit directly to main branch" button is selected, and then click "Commit changes."

IMPORTANT

Branching is a common component of working in GitHub. The **main** branch is the main branch that showcases your completed project. For a large-scale project, different aspects of the project will each have its own branch. This is especially useful when working in teams.

For example, Branch A was created to add another component to a project. This branch contains a copy of the main branch. When a developer adds items to this branch, the new changes are isolated within Branch A. This allows developers to iron out any kinks that may come up when making file changes while keeping the main branch intact.

Now that there are two branches, main and Branch A, how do new additions get integrated? After the changes in Branch A have been checked for errors and quality, they are merged into the main branch.

This process, called **merging,** pulls in only the newly added or adjusted files.

NOTE

Historically, the most common name for the main body of a codebase has been **master**. Recently, however, **main** has been gaining in popularity. In fact, GitHub now uses **main** as the default name for its repositories - as do the projects in this course. Be aware that you might see instances of both throughout your development career, or hear experienced coders use the term "master branch" out of habit.

We can view our files and commit messages on the main page of the repo. The README, which is automatically displayed, doesn't have any content---yet! Let's fix that.

Edit the README

A **README** file is the face of a repository: it provides an explanation of the project. It can be customized using **markdown** to provide a pleasing aesthetic.

Markdown is a plain text editing language used to make plain text prettier. It can insert images, create hyperlinks, create bulleted lists, and many other customizations. Through GitHub, markdown is even rendered as a webpage, though it doesn't possess as much customization as a page powered with HTML. Commonly found in forums and README files, markdown is a language that is fairly simple to pick up and put into practice.

We can edit the README markdown by clicking the button with a pencil icon on the main repository page. The very first line in our new editor should read `# kickstarter-analysis`. The pound sign, or hashtag, signifies a first-level header, or the title of the document. "Kickstarter-analysis" is the repository name, which is integrated into the document automatically by GitHub.

This file is where we'll compose our analysis report to be shared with others---or, in this case, Louise. Here's a breakdown of basic markdown syntax:

- **Headers** are signified with a #. The number of hashtags indicates the level of the header. For example, ### is a third-level header.
- **Bullets** are added in three ways: a typical bullet with an asterisk, a numbered bullet with numbers, or a hyphenated bullet with a hyphen.
- **Images** can be added with the following syntax: `![image_name](path/to/image_name.png)`.
- **Hyperlinks** to relevant files are added in a similar manner: `[filename](path/to/filename.xlxs)`.
- **Line breaks** are added by using three consecutive hyphens: `---`.

Refer to the **GitHub Help page (https://help.github.com/en/articles/basic-writing-and-formatting-syntax)** for additional information about creating documentation using markdown format.

Now we're ready to flesh out our analysis!

Begin by updating the header to a more appropriate title like "An Analysis of Kickstarter Campaigns." Below the header, add a brief description of the project.

In addition to your findings, you should also include charts and graphs (visualizations) in your report. Visualizations help communicate the story the data is telling by making it more understandable and coherent. At the end of the report, add any recommendations you have for Louise based on your findings. Then add a descriptive commit message and click "Commit changes." This descriptive README file adds depth to the repository as well as your analysis. It's ready to be shared!

Return to the main page of your GitHub repository, highlight the address, and copy it. This link, which should read `https://github.com/yourusername/kickstarter-analysis` can now be emailed to our hopeful playwright, Louise, so she can begin her Kickstarter preparations.