15.S60: Computing for Optimization and Statistics
Homework 1
Due: January 13, 2026 11:59pm EST

For this assignment, you will explore one use of Git, Github, and terminal in more depth.

1 Writing a LaTeX document and citing in Zotero

We will create a LaTeX document that will be used for the homework submission. For this step, you must include the following:

1. Add a header (you can use the`/maketitle` command) that has the title "Homework 1", the author as your name, and optionally the date.
2. Add the first section and name is "LaTeX and Zotero"
3. In the content of the section, add a citation of some article, book, etc using Zotero. For example, you can just write, "I am citing [insert citation here]".

2 Connect Git and Github with SSH

We will now use git/terminal to push changes to Github; we need to tell git/terminal who you are and authenticate your Github credentials. We do this by connecting over HTTPS or with a SSH (Secure Shell) key. Follow instructions here in the "Authenticating with GitHub from Git" section to set up the connection.

3 Creating a personal website via Github Pages

GitHub Pages is a static site hosting service that takes HTML, CSS, and JavaScript files straight from a repository on GitHub and publishes a website, automatically updating when there are new commits. Each Github user can host one personal website. In this assignment, you will build and deploy a personal website from an academic website template called Academic Pages (example site), all while practicing skills from Session 7. Once you've completed the assignment, you can complete your personal website by customizing the Academic Pages template with your info, finding a different template for your site, or building your own from scratch.

1. Fork the Academic Pages Github. Name your forked repo `yourgithubusername.github.io` (for example, `lisaxeverest.github.io`). This will become the URL of your website. Following this exact naming scheme is crucial. If your repo is named anything other than this, it will not be published by Github. If you already have a site hosted by Github pages at that domain, then you can either create a new Github account to complete the homework or you can follow the instructions in the Creating a Project Website section below.
2. Your site will go live once you make a Github commit, triggering the Github Pages deployment process. In steps 3-6, we will make changes to your website by cloning to a local repo, editing locally, and pushing to Github. But if you want a sneak preview of your site before you edit anything, click the `README.md` file in the Github repo, click the pencil icon to edit, make a small change like adding a space, then scroll to the bottom and click "Commit changes". In a few minutes, your site template should be live at `https://yourgithubusername.github.io`. If your site is not live, you can check on the site deployment process by navigating to the Actions tab of your repository and clicking on the "Pages build and deployment" workflow. Click "Re-run all jobs" to try deployment again.

3. Open Terminal or Git Bash. Clone your forked repo to your local machine into the directory of your choice.

4. Make at least *two* commits to your cloned repo to make the website your own. For example, you could:

   - Change your name, location, and sidebar links in `config.yml`
   - Upload a profile picture to images folder and change avatar to the image file name in `config.yml`
   - Add some text to your home page in `pages/aboutme.md`
   - Change the pages displayed in the navigation bar by editing `data/navigation.yaml`
   - Upload your resume or CV as a PDF to the files folder. Add a link to the PDF in the navigation bar by editing `data/navigation.yaml` and changing the URL for the CV page to `/files/filename.pdf`

   You don't need to finish editing your website as part of this homework. The goal is just to get some practice with Git and Github and get you started with a personal website if you would like one.

5. **Take a screenshot of your commit history** that shows at least two commits.

6. Push your changes to Github. The remote connection should already be set up since you cloned the repo. To complete this step, you must have completed the SSH key set up from the beginning of this assignment.

7. Go to `https://yourgithubusername.github.io`. It may take 5-10 minutes for changes you pushed to be published. **Take a screenshot of your site** with your name at the top. It's fine if most of your site is still generic text and the default format.

8. Create a new section in your LaTeX document, called "Personal website". Add the two screenshots, commit history and live website, to the LaTeX document (hint: you can use the `plot.tex` file from class to see an example of this). **Compile the LaTeX document as a PDF and submit the PDF to Canvas, Assignment 1.**

9. At this time, you can either leave your site live or unpublish it. To unpublish, navigate to the Settings tab of your repo and click Pages in the left sidebar. At the top where it says "Your site is live at", click the three dots icon on the right and select "Unpublish site". If you finish editing your site later and want to re-publish it, navigate to the Actions tab of your repository, click on the "Pages build and deployment" workflow, then click the "Re-run all jobs" button in the upper right corner of the page. After the workflow run has completed, your site will be published.

4 Creating a Project Website

If you are already using your Github Pages personal domain (https://⟨githubusername⟩.github.io) for another purpose, then you can alternatively fork the Academic Pages template and create a **project** website for it.

1. Fork the Academic Pages Github repo and give it any name you like

2. Go to the setting of this repository in Github

3. Click "Pages" in the left sidebar and be sure that the "Source" dropdown says "Deploy from a branch". Use the "Branch"drop down to select `master` or `main` branch

4. In a few minutes, your new project site will be available at ⟨githubusername⟩.github.io/reponame. If your site is not live, you can check on or re-run the site deployment workflow by navigating to the Actions tab of your repository and clicking on the "Pages build and deployment" workflow.