Tech Setup Challenge

Getting Started with Your Analytics Toolkit

🚀 Tech Setup Challenge - Quick Start Guide

Challenge Overview

This challenge will get you set up with many of the essential tools for our Decision Analytics course. You'll learn to use modern development tools that are standard in the analytics industry.

Part 1: Core Tools Setup ⚙️

1. 🔧 Git Installation & Setup

What is Git? Git is a version control system that tracks changes to your files. It's essential for collaborating and backing up your work.

Install Git:

- **Windows:** Download from git-scm.com → Run installer with default settings
- **Mac:** Download from git-scm.com or install via Xcode Command Line Tools: `xcode-select --install`

Configure Git (Required First-Time Setup):

You need to tell Git who you are. You'll run the commands below in a **terminal** (a text-based way to interact with your computer—don't worry, you just copy/paste the commands!).

- **How to open a terminal (easiest first):**
  - **In Cursor (easiest):** Press ``Ctrl+` `` (backtick key, usually under Esc) or click the terminal icon in the activity bar. Run the commands below there—no need to leave the editor.
  - **External terminal (alternative): Windows:** Press `Win + R`, type `cmd`, press Enter (or search "Command Prompt" in Start menu). **Mac:** Press `Cmd + Space`, type "Terminal", press Enter.

- **Run these commands** (replace with your actual name and email):

  ```
  git config --global user.name "Your Name"
  git config --global user.email "your_email@example.com"
  ```

  Example: `git config --global user.name "John Doe"` and `git config --global user.email "johndoe@example.com"`. Use the same email you use for GitHub.

- **Verify setup:** Run `git config --list` to see your configuration. To confirm Git is installed, run `git --version` (you should see something like `git version 2.x.x`).

- **Windows users (optional):** When installing Git, at "Line ending conversion" choose **"Checkout as-is, commit Unix-style line endings"** (recommended)—helps avoid line-ending issues later.

2. 📝 Cursor AI Setup

Two Parts: Online Dashboard + Local App

- **Online Dashboard:** Go to cursor.com/students
    - Sign up with your **udel.edu email** (free for students for 1-year! 🎓)
    - This is just for billing/account setup - you won't do work here
- **Local App:** Download and install Cursor AI from the same page
    - This is where you'll actually write code and do your work
    - Launch the app and sign in with your udel.edu email
    - **Troubleshooting:** If free access doesn't work immediately, start the 7-day free trial and verify student status later

2.5. 🎛️ Cursor AI Interface & Git Integration

Getting Familiar with Cursor AI: Challenge yourself to explore the menus and use documentation to find the below items. (https://docs.cursor.com/en/welcome)

- **Activity Bar** (left side): Contains icons for different features
    - Explorer (files)
    - Search
    - Source Control (Git)
    - Agents
    - 📦 Extensions
- **Sidebar** (right of activity bar): Shows content for the selected activity

💾 Saving Files in Cursor AI:

- **Auto-save:** Cursor AI automatically saves files to memory as you type (you'll see a small dot on the file tab when there are unsaved changes)
    - **Note:** Auto-save is convenient but doesn't guarantee your work is written to disk
- **Manual save:** Press `Ctrl+S` to save the current file to disk
    - **Important:** This ensures your work is permanently stored and visible to Git
- **Save all:** Press `Ctrl+Shift+S` to save all open files to disk
- **File indicators:**
    - White dot on tab = unsaved changes
    - No dot = file is saved
    - Always manually save before closing files, switching tasks, or committing to Git!

AI Interaction in Cursor:

- **Quick Chat (Ask Mode):** Press `Ctrl+L` to open a quick chat with AI
    - Perfect for asking questions about your code
    - Gets context from your current file
    - Great for debugging, explaining code, or getting suggestions
- **Agent Mode:** Click the **Agents icon** in the activity bar
    - More powerful AI assistant that can work across your entire project
    - Can analyze multiple files, suggest refactoring, and help with complex tasks
    - **Difference:** Ask mode is for quick questions, Agent mode is for deeper project-wide assistance

Connect Git to Cursor AI:

- Click the **Source Control icon** (three-branch icon) in the activity bar
- Since you already installed Git, Cursor AI should automatically detect it
- This sets up Git tracking (version control) for your local project folder
  - **Why?** Version control lets you save your work progress and collaborate with others
- Later in this tutorial, you will see a "Changes" section in the sidebar showing modified files
  - **Why?** This helps you see exactly what files you've changed before submitting
- **Alternative:** You can also click "Clone Repository" in the Source Control panel to get your GitHub repository

3. 🔗 GitHub Account

- Create account at github.com (if you don't have one)
- Not necessarily needed, but use your **udel.edu email** for consistency/simplicity

4. 📥 Get Course Materials

Fork & Clone:

- Go to [course repo: https://github.com/flyaflya/decAnalytics](https://github.com/flyaflya/decAnalytics) on GitHub (make sure you are logged in to GitHub, otherwise you won't see the Fork button)
- Click the **"Fork"** button (top right) to create your own copy
  - **Why?** Forking creates your own copy on GitHub where you can make changes without messing up Adam's GitHub repository
  - **Why?** You can't directly edit the original course repository
- Now, get a copy of your GitHub repository locally so you can edit with Cursor AI. In Cursor AI, press `Ctrl+Shift+P` → "Git: Clone"
- Paste your forked repo URL: `https://github.com/YOUR_USERNAME/decAnalytics`
  - **Why?** This downloads your personal copy to your computer
- Choose where to save locally (I recommend somewhere easy to find like C:). A folder named `decAnalytics` will be placed in this location for you upon cloning.

5. 👤 Create Your Student Folder

Using the Activity Bar & Sidebar:

- Click the **Explorer icon** (looks like stacked files) in the activity bar to open the file explorer
- In the sidebar, you'll see your project files
- Navigate to `student-portfolios/AdamF`
  - **Tip:** Click the small arrows (>) next to folder names to expand them
- **Right-click** in the `student-portfolios` folder → "New Folder"
- Name your folder: `YourNickNameLastInitial` (e.g., `JohnD`, `SivaX`, etc.)
  - **FERPA Privacy Option:** If you prefer not to use your real name, you can use a pseudonym like `Student001`, `Analyst2025`, or `DataLearner` instead. Just email Adam (ajf@udel.edu) and he'll note the connection.
- **Right-click** on `AdamF/README.md` → "Copy" (this will be your template to modify for your submission)
- **Right-click** in your new nickname folder → "Paste" (this is the file you will modify for your submission)
- **Double-click** the copied `README.md` (i.e. the one in your newly created nickname folder) to open and edit it
- Edit `README.md` with your info and add two cool pictures of you and/or your interests
  - **Important:** Save the image files in your student folder (e.g., `photo1.jpg`, `photo2.png`)
  - **Why?** The images need to be in your folder to be included in your GitHub submission
  - **Note:** These will be public images, so choose appropriate photos

- 💾 **Save your work:** Press `Ctrl+S` to save the file after making your edits
- 🔄 **Commit your progress:** After saving, commit your changes to protect your work:
  - Press `Ctrl+Shift+G` to open Source Control
  - Click the "+" next to your modified files to stage them
  - Write a commit message like "Add personal info to my README.md"
  - Click "Commit" (✓ icon)
  - **Why commit now?** This creates a backup of your work in case something goes wrong!

5.5. 🔄 Sync to GitHub & Verify Content

Before submitting your pull request, let's make sure everything is properly uploaded and visible on GitHub:

- 💾 **Save all files first:** Press `Ctrl+Shift+S` to save all your work
- **Push to GitHub:** In Cursor AI, click "Sync Changes" or "Push" to upload your changes
  - **Why?** This uploads your local changes to your GitHub fork
- **Verify on GitHub:** Go to your forked repository: `https://github.com/YOUR_USERNAME/decAnalytics`
- **Check your student folder:** Navigate to `student-portfolios/[YourNickNameLastInitial]/` on GitHub
- **Verify your content:**
  - □ Your `README.md` file is visible and contains your personal information
  - □ Your photos are properly displayed in the README.md file
  - □ All image files are uploaded to your student folder
  - □ No broken image links or missing content
- **If anything is missing or incorrect:**
  - Go back to Cursor AI and make the necessary changes
  - Save, commit, and push again
  - Refresh your GitHub page to verify the updates
- **Only proceed to the next step when everything looks correct on GitHub!**

6. 🔄 Submit via Pull Request

What is a Pull Request? A pull request is like asking "Hey, can you add my changes to your project?" It's how you submit your work for review and inclusion in the main course repository. Think of it as a digital way to turn in your homework!

Since you already synced your changes to GitHub in step 5.5, you can now create your pull request directly:

- Go to your forked repo on GitHub to create the pull request

Create Pull Request:

- Go to your forked repo: `https://github.com/YOUR_USERNAME/decAnalytics`
- You should see a banner saying "This branch is X commits ahead of flyaflya:main"
  - **Why?** This shows your fork has changes that aren't in the original course repo
- Click **"Compare & pull request"**
  - **Why?** This requests to merge your changes into the original course repository
- Add description: "Student submission for [YourName]"
  - **Why?** This helps identify your submission and explains what you did
- Click **"Create pull request"**

    - **Why?** This submits your work for review and inclusion in the course materials

Congratulations! You’re Done! ✅

You’ve completed the tech setup challenge! Check the checklist below to make sure you’ve completed all the steps.

📋 Submission Checklist

□ Git installed and configured with your name/email
□ Cursor AI installed and configured
□ Git connected to Cursor AI (Source Control)
□ GitHub account created
□ Course materials forked and cloned
□ Student folder created with README.md
□ Personal info and photos added to README.md
□ Initial commit made for README.md
□ Pull request created for initial submission

Quick Git Basics

- **Terminal** = `` Ctrl+` `` - Open the integrated terminal (backtick key, usually under Esc)
- **Save** = `Ctrl+S` - Save current file to disk (do this frequently!)
- **Commit** = Save your changes to Git history so you can go back to these files at this instant in time - do often as AI will gladly mess up your files and give you an unrecoverable mess on your hands
- **Push** = Upload to GitHub
- **Pull Request** = Request to add your changes to original repo
- **Fork** = Your copy of the original repo

💡 **Pro Tip:** Save files frequently (`Ctrl+S`) and commit your work (`Ctrl+Shift+G` → stage → commit) after completing each major step. This way, if something goes wrong, you can always recover your work!

🆘 Troubleshooting

- **“Git is not recognized”** – Install Git, then fully restart Cursor so PATH is picked up. Use the integrated terminal (`` Ctrl+` ``) rather than an external terminal.
- **PowerShell execution policy (Windows):** If you get an execution policy error when running scripts (e.g. later in the course), in PowerShell run:
  `Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser`
  Type `Y` when prompted. This allows locally created scripts to run (safe default).

🆘 Need Help?

- GitHub tutorials: guides.github.com

- Cursor AI has built-in help and AI assistance. Note cursor is built on VS Code, so google searches using VS Code in place of Cursor will often lead to helpful solutions. Search "VS Code [your question]"
- Ask questions in class or office hours!

Total time to complete: ~70 minutes ⏱️

Why These Tools Matter 🎯

In the Real World:

- **GitHub** = Industry standard for collaboration
- **Cursor AI** = Modern AI-assisted development
- **Git** = Version control (essential for any coding project)

For This Course:

- **Challenges** will require Quarto reports (not required for this challenge)
- **Portfolio** will be built on GitHub
- **Collaboration** will happen through GitHub

You're learning leading tools that are used by data scientists, analysts, and developers worldwide! 🌎

🔄 Keeping Your Fork Updated (Getting Course Updates)

Why you need this: When I add new course materials, challenges, or updates, your fork won't automatically get them. You'll need to sync manually.

Easy Method (GitHub Web Interface):

1. Go to your forked repo: `https://github.com/YOUR_USERNAME/decAnalytics`
2. Look for a **"Sync fork"** banner at the top (appears when there are updates)
3. Click **"Update branch"** to pull in the latest changes
4. In Cursor AI: `Ctrl+Shift+P` → "Git: Pull" to download changes to your computer

What this does:

- **Syncs your GitHub fork** with my original repository
- **Downloads updates** to your local computer
- **Keeps you current** with new course materials and challenges

💡 **Pro Tip:** Check for the "Sync fork" banner whenever you start working on new course materials. This ensures you're working with the latest version!