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 BUAD 442. You'll learn to use modern development tools that are standard in the analytics industry.

Part 1: Core Tools Setup

1. Cursor AI Setup

- Go to cursor.com/students
- Sign up with your **udel.edu email** (free for students for 1-year! , but may need credit card)
  - **Backup plan:** If student account approval is delayed or has issues, you can sign up for a free 2-week trial at cursor.com while waiting for your student account to be approved
- Download and install Cursor AI (default choices for installation are fine)
- Launch Cursor AI

1.5. Cursor AI Interface & Git Setup

Getting Familiar with Cursor AI:

- **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 as you type (you'll see a small dot on the file tab when there are unsaved changes)
- **Manual save:** Press `Ctrl+S` to save the current file
- **Save all:** Press `Ctrl+Shift+S` to save all open files
- **File indicators:**
  - White dot on tab = unsaved changes
  - No dot = file is saved
  - Always save before closing files or switching tasks!

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

Set up Git for Local Source Control:

- Click the **Source Control icon** (three-branch icon) in the activity bar
- Follow prompts to Download Git (use default settings when prompted)
  - **Why?** Git tracks all changes to your files, so you can see what you've modified
- **Return to Cursor** after Git installation completes
- 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

2. 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

3. Get Course Materials

Option A - Fork & Clone (Recommended):

- Go to course repo: https://github.com/flyaflya/buad442Fall2025 on GitHub (ake 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/buad442Fall2025`
  - **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 `buad442Fall2025` will be placed in this location for you upon cloning.

4. 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 `challenges/01-Tech-Setup-Challenge/student-submissions/`
  - **Tip:** Click the small arrows (>) next to folder names to expand them
- **Right-click** in the `student-submissions` 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/submission.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 `submission.md` (i.e. the one in your student folder) to open and edit it
- Edit `submission.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 submission.md"
  - Click "Commit" (icon)
  - **Why commit now?** This creates a backup of your work in case something goes wrong!

5. 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!

In Cursor AI:

- **Save all files first:** Press `Ctrl+Shift+S` to save all your work
- Stage your changes: `Ctrl+Shift+G` → click "+" next to your new folder
  - **Why?** This tells Git which files you want to include in your submission
- Commit: Write message like "Add student submission for [YourName]"
  - **Why?** This creates a permanent record of your changes with a description
- Push: Click "Sync Changes" or "Publish Branch"

  - **Why?** This uploads your changes to your GitHub fork

- 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/buad442Fall2025`
- 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

Part 2: Quarto Setup

Why Quarto?

Quarto is essential for creating **professional reports** that combine:

- **Text** (like this document)
- **Code** (Python/R analysis)
- **Visualizations** (charts, graphs)
- **Interactive elements** (dashboards)

It's not just for code - it's for creating beautiful, reproducible documents that are perfect for business presentations and reports.

6. Install Quarto

- Go to quarto.org/docs/get-started/
- Download and install Quarto for your operating system
- Verify installation: Open terminal/command prompt and type `quarto --version`
- **Recommended:** Install the Quarto extension in Cursor AI for better editing experience:

  - Press `Ctrl+Shift+X` to open Extensions
  - Search for "Quarto"
  - Install the "Quarto" extension by Quarto
  - **Why?** This gives you syntax highlighting, live preview, and better editing support for `.qmd` files

7. Test Quarto

Create a simple test document:

1. Create a new file in your student submission folder called `test.qmd`
2. Add this content (replace bracketed text with your nickname):

```
---
title: "My First Quarto Document"
format: html
---

# Hello World!

This is my first Quarto document. My nick name is [YourNickNameLastInitial].
```

3. **Save the file:** Press `Ctrl+S` to save your new `test.qmd` file
4. **Quick commit:** After saving, commit this new file:

   - Press `Ctrl+Shift+G` → click "+" next to `test.qmd`
   - Write commit message: "Add test Quarto document"
   - Click "Commit" (icon)

5. Render it using the terminal/command prompt: **Opening Terminal in Cursor AI:**

   - Press 'Ctrl + " (backtick key, usually under the Esc key) to open the integrated terminal
   - OR go to the top menu: **Terminal** → **New Terminal**
   - OR press `Ctrl+Shift+P` → type "Terminal: Create New Terminal" → press Enter

   Navigating to Your File:

   - The terminal opens in your project root folder, but you need to be in your student folder
   - Navigate to your student folder: `cd "C:\buad442Fall2025\challenges\01-Tech-Setup-Challenge`
   - Replace `YourNickNameLastInitial` with your actual folder name
   - **Pro Tip:** Press the `Tab` key while typing to auto-complete folder names! This saves time and prevents typos.

Alternative (if you installed the Quarto extension):

- Open your `test.qmd` file in Cursor AI
- Look for the **“Preview”** button in the top-right corner of the editor
- Click it to render and view your document without using the terminal!

Running the Render Command:

- Type: `quarto render test.qmd` and press Enter
- You should see output indicating the file was rendered successfully
- This will create a `test.html` file in the same folder
- **Commit the generated HTML:** After successful rendering, commit the new `test.html` file:
  - Press `Ctrl+Shift+G` → click “+” next to `test.html`
  - Write commit message: “Add rendered Quarto HTML output”
  - Click “Commit” (icon)

Alternative Method (if terminal seems complicated):

- Right-click on `test.qmd` in the file explorer
- Look for “Open in Terminal” or “Open in Command Prompt” option
- Then type: `quarto render test.qmd`

6. Open the generated `test.html` file
7. **Final commit before submission:**
   - Press `Ctrl+Shift+S` to save all files
   - Press `Ctrl+Shift+G` → stage any remaining changes
   - Write commit message: “Complete Quarto setup and testing”
   - Click “Commit” (icon)
   - **Why?** This ensures all your work is safely backed up before the final submission!

8. Update Your Pull Request

Your pull request will update with new commits! Here’s how:

- **Push your new commits:** In Cursor AI, click “Sync Changes” or “Push” to upload your Quarto work
- **Check your pull request:** Go back to your GitHub pull request page
- **You’ll see:** The pull request now includes both your initial submission AND your Quarto testing
- **The pull request updates automatically** because it’s tracking your branch, not specific commits

What you should see:

- Your original pull request now shows all your commits in the "Commits" tab
- The "Files changed" tab will show both your `submission.md` and `test.qmd`/`test.html` files
- Everything is in one place for easy review

Pro Tip: This is how real development works - you keep adding commits to your branch, and the pull request shows all changes on that branch!

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

☐ Cursor AI installed and configured
☐ GitHub account created
☐ Course materials forked and cloned
☐ Student folder created with submission.md
☐ Personal info added to submission.md
☐ Initial commit made for submission.md
☐ Pull request created for initial submission
☐ Quarto installed and verified
☐ Test Quarto document (test.qmd) created
☐ Quarto document rendered to HTML
☐ All Quarto files committed
☐ Pull request updated with Quarto work
☐ All work pushed to GitHub

Quick Git Basics

- **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!

Need Help?

- GitHub tutorials: guides.github.com
- Quarto docs: quarto.org/docs
- 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
- **Quarto** = Professional reporting and documentation
- **Git** = Version control (essential for any coding project)

For This Course:

- **Challenges** will require Quarto reports
- **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/buad442Fall2025`
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!