

# EC1410-Fall 2026

## Problem Set 1.AI

(Updated 31 August 2026)

Matt Turner

(100 points possible)

This problem set asks you to install and set up a Claude coding agent and to use this agent to plot a land price gradient from data describing 19th century Chicago.

To do this, you need to first install some software related to Python, the language the coding agent will use, and some software related to JavaScript, the language that runs the agent. To do these things, you need to be able to issue commands from a shell window, so that is the first step. Once this is all done, you will be able to ask the agent to do your data analysis for you.

If you get errors while following these instructions, copy them into the chat window of the LLM of your choice (I use Claude) and ask for help.

1. **Shell Windows.** Many of the steps below require you to type commands into a *shell window* — a text-based interface where you give your computer instructions by typing rather than clicking.

This setup uses two different shell windows, and it matters which one you use for which step. Each step below states which one to open.

**Anaconda Prompt** is a shell window installed alongside Anaconda or Miniconda. It is configured to know about conda and Python, which ordinary shell windows are not. This will only work after you install Anaconda (below).

- **Windows:** Click the Start menu (the Windows logo, bottom-left of your screen), type **Anaconda Prompt**, and press Enter. A dark window opens with a prompt like: (base) C:\Users\yourname>
- **Mac:** Press **Cmd+Space**, type **Terminal**, and press Enter.

**PowerShell** is a shell window built into Windows.

- **Windows:** Right-click the Start menu (Windows logo) and click **Windows PowerShell** or **Terminal**. The prompt looks like PS C:\Users\yourname>.
- **Mac:** Press **Cmd+Space**, type **Terminal**, and press Enter.

2. **Install Python via Anaconda or Miniconda.** (*Open Anaconda Prompt (Windows) or Terminal (Mac) for this step.*)

Anaconda and Miniconda are both standard ways to install Python. Miniconda is a lighter-weight version of Anaconda; either works fine for this course. If you already have one of them installed, skip this step.

- (a) Go to <https://www.anaconda.com/download> (full Anaconda) or <https://docs.conda.io/en/latest/miniconda.html> (Miniconda) and download the installer for your operating system. Either is fine.
- (b) Run the installer, accepting all defaults.
- (c) Open an Anaconda Prompt (Windows) or Terminal (Mac).
- (d) Accept the Terms of Service. Recent versions of Anaconda require this before you can create environments. Run each of the following three commands, pressing Enter after each one:

```
conda tos accept --override-channels --channel https://  
repo.anaconda.com/pkg/main
```

```
conda tos accept --override-channels --channel https://  
repo.anaconda.com/pkg/r
```

```
conda tos accept --override-channels --channel https://  
repo.anaconda.com/pkg/msys2
```

If you do not see a Terms of Service error, you can skip this step.

- (e) Create a dedicated environment for this course. A Python *environment* is a collection of packages — tools and functions that Python programs can borrow. The command below creates an environment named `urbanecon` and installs the packages the coding agent will need:

```
conda create -n urbanecon -c conda-forge python=3.11  
pandas matplotlib scipy statsmodels requests jupyterlab  
pyreadstat -y
```

This takes a few minutes. When it finishes, activate the environment. This tells your computer to use this particular toolbox:

```
conda activate urbanecon
```

You will need to run `conda activate urbanecon` every time you open a new Anaconda Prompt session before using the coding agent. If you forget, the agent will launch fine but any Python code it runs will fail to find the required packages.

3. Install Node.js (*Open PowerShell (Windows) or Terminal (Mac) for this step.*)

The coding agent is a Node.js application — a program written in JavaScript that runs outside the browser. You do not need to know JavaScript; Node.js is simply the engine the agent runs on, the same way Python is the engine that runs `.py` scripts. The two coexist independently on your computer.

- (a) Go to <https://nodejs.org> and download the **LTS** version for your operating system.
- (b) Run the installer with default settings.
- (c) Open a PowerShell window (Windows) or Terminal (Mac). Type the following and press Enter:

```
node -v
```

You should see a version number such as v22.x.x. If you get an error, ask an LLM: *"I installed Node.js LTS on [your OS] but the node command is not found. How do I fix this?"*

- 4. Install Claude Code. (*Open PowerShell (Windows) or Terminal (Mac) for this step.*)

Type the following commands one at a time, pressing Enter after each. You may need to type them rather than cut-and-paste (copy-paste from a PDF can introduce invisible spaces around hyphens and other characters that cause errors.) The characters @, /, and - in the package name are required and must be typed exactly.

First, install the package:

```
npm install -g @anthropic-ai/claude-code
```

You will likely see a warning about *install scripts not yet covered by allowScripts*. This is expected. Fix it by running:

```
npm config set allow-scripts=@anthropic-ai/claude-code --  
location=user
```

Then install once more so the post-install script runs:

```
npm install -g @anthropic-ai/claude-code
```

This time the warning should not appear. You may see a notice about a new version of npm being available — this is informational and can be ignored.

- 5. Get an Anthropic API Key. *The API key setup uses a web browser. The verification command at the end of this step is run in Anaconda Prompt (Windows) or Terminal (Mac).*

Claude Code uses Anthropic's API. You need an account with a small amount of credit loaded. Each problem set task will cost a few cents.

- (a) Go to <https://console.anthropic.com> and create an account.
- (b) Add a payment method. You will not be charged until you use the API.
- (c) Go to **Settings** → **API Keys** and click **Create Key**.

(d) Give it a name (e.g., “urban econ”) and copy the key — it begins with sk-ant-  
...

(e) **Save it somewhere safe.** You cannot view it again after closing that page.

Now store the key as a permanent environment variable on your computer. Ask an LLM: “How do I permanently set an environment variable called `ANTHROPIC_API_KEY` to [your key] on [Mac / Windows]?” Follow the instructions. Then close and reopen your terminal.

Open an Anaconda Prompt and confirm the key is visible:

|                    |                                       |
|--------------------|---------------------------------------|
| <b>Mac / Linux</b> | <code>echo \$ANTHROPIC_API_KEY</code> |
| <b>Windows</b>     | <code>echo %ANTHROPIC_API_KEY%</code> |

You should see your key printed back. If you see nothing, the variable is not set correctly — revisit this step.

## 6. Create a Homework Folder and Start the Agent.

**Important: always navigate into your homework folder before running the agent.**

The coding agent (claude) has no hard restriction on which parts of your filesystem it can access. It uses judgment-based guardrails and will generally stay in your working directory, but these are not enforced at the operating-system level. Before every session, open an Anaconda Prompt and navigate into your homework folder. For example:

```
cd Desktop/echw/pset1
```

This ensures output files land in the right place and reduces the chance of the agent touching files elsewhere on your computer. **Never run** claude **from your home directory or the root of your filesystem.**

(Open Anaconda Prompt (Windows) or Terminal (Mac) for this step.)

Type the following commands one at a time, pressing Enter after each.

(a) Create a folder for this course’s homework and navigate into it:

```
mkdir echw
cd echw
mkdir pset1
cd pset1
```

(b) Activate your conda environment:

```
conda activate urbanecon
```

The prompt should change to show (urbanecon) at the start, confirming the environment is active.

(c) Launch the agent:

```
claude
```

You will see a welcome prompt. You are now talking to a coding agent that can see your working folder, write files to it, and execute Python.

Your homework is to make a plot of the land price gradient in 19th century Chicago using the data in the replication package for *The Value of Piped Water and Sewers: Evidence from 19th Century Chicago* by Michael Coury et al., published in the *Review of Economics and Statistics*. Ask the Claude agent to download the replication package and make the picture. A suggested prompt is below.

*The dataset DOI is doi:10.7910/DVN/9TWG7D on Harvard Dataverse. Download the replication data using the Dataverse API. Inspect the files to identify a land or housing price variable and a distance-to-city-center variable. Produce a scatter plot of price on the y-axis and distance on the x-axis with a LOWESS smoother overlaid. Save the plot as pricegradient.png. Print the variable names you used and describe in a comment what units each variable is in.*

The agent will download, inspect, code, and iterate on its own. When it finishes, pricegradient.png will be in your pset1 folder. If a result looks wrong, describe the problem and it will try again.

1. (50) Hand in the plot you have just generated and answer the following questions:
2. (10) What are the exact units on the x axis?
3. (10) What are the exact units on the y-axis?
4. (30) Does the gradient behave the way the theory presented in class predicts?