AI, coding, and judgment

Coding Lesson 1 · Math Camp 2026

Rony Rodriguez-Ramirez

Harvard Kennedy School · API 209

Thursday, August 13, 2026

Lesson map

Welcome and the question

Where we begin

Why coding still matters

From a model to an agent

Learning with an agent

Delegating a bounded task

From Lesson 1 to Lab 1

Welcome

Rony Rodriguez-Ramirez

- I work in education policy, with a focus on economics of education.
- I have worked with research teams at the World Bank and Innovations for Poverty Action.
- I like coding when it helps us ask a clearer question.
- I will teach these coding lessons and work with you during the labs.

We have different starting points

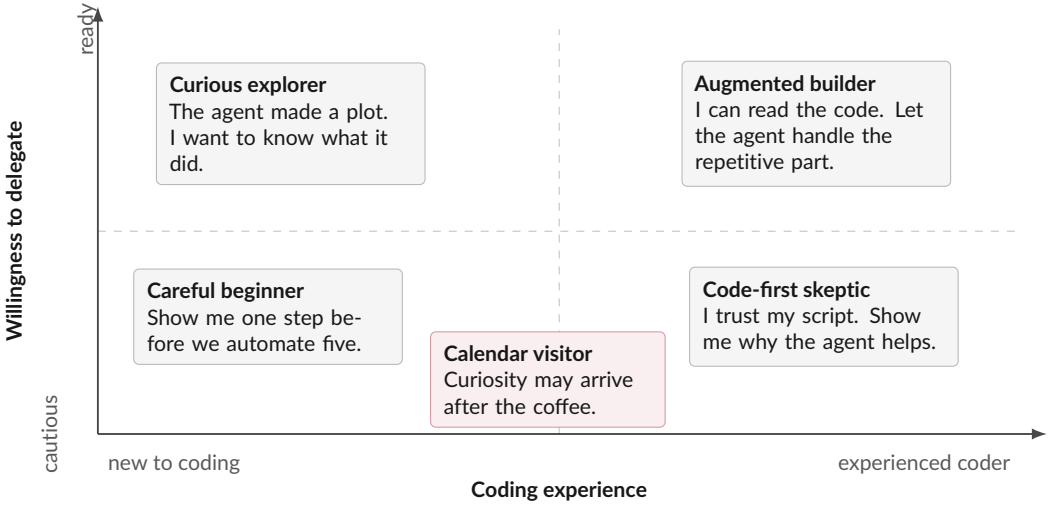
Some people in this room are writing code for the first time. Others already use R for research. Many people know ChatGPT. A coding agent that works inside a project will be a newer experience for most of us.

This is fine

We will use one common project. Each student can take steps that fit their experience with coding and their current comfort with AI.

Your starting point can also change from one task to another.

Some AI + coding archetypes



These positions are approximate. Your position can change with the task.

The question for today

What should we delegate, to which system, and with what evidence?

The task and the evidence determine which tool belongs in the workflow.

What I want you to do by the end

1. Distinguish a model, a chat interface, a coding agent, and a harness.
2. Explain the different jobs of R, RStudio, Quarto, and Codex.
3. Decide whether a task should stay with you or can be delegated.
4. Write one bounded request with a visible completion check.
5. Inspect an agent result before you keep it.

Today we need enough shared understanding to use this workflow together. Expertise will grow through practice during Math Camp and the semester.

Two hours before Lab 1

Minutes	Part	What we will accomplish
00–50	Block 1	Concepts plus checkpoints at 16–20, 30–34, and 46–50.
50–60	Break	Ten minutes away from the screen.
60–110	Block 2	Learning and delegation, two checkpoints, then a live policy briefing.
110–120	Handoff	Open the project, locate the readiness check, and move into Lab 1.
120–180	Lab 1	Installation clinic: every laptop reaches green or leaves with an owned repair plan.

Is it still worth learning to code?

Yes, because code helps us

- make an instruction precise;
- repeat the same operation;
- inspect what a tool produced;
- change one part and test it.

Learning code is less about remembering every function. It is more about reading, predicting, changing, testing, and explaining.

Andrew Ng argues that a lower barrier to coding increases what people can build. Even a small amount of coding knowledge can therefore support more ambitious work (Ng, 2023, 2025).

The target has changed

A useful coding learner can:

- read a proposed script and say what it should do;
- change the relevant line without changing the question;
- run a small test and interpret the result;
- find where a claim enters the code;
- notice when the code and the policy question separate.

Syntax still matters alongside supervision, testing, and interpretation.

Why the workflow changed

Earlier loop

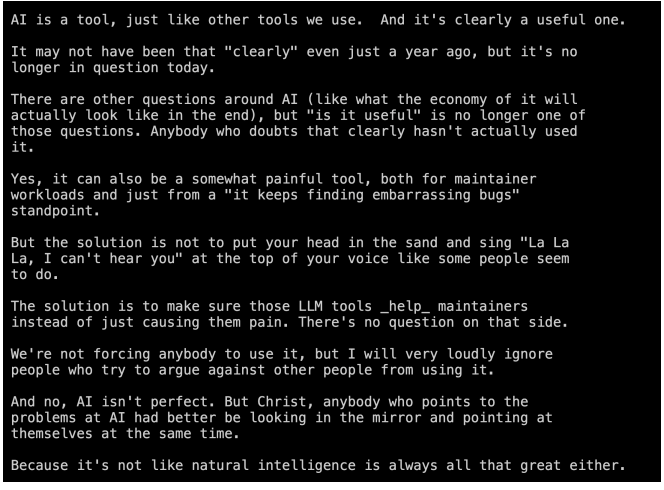
1. Write code.
2. Run code.
3. Search the error.
4. Try again.

Agent-assisted loop

1. Define the result and evidence.
2. Let the agent inspect or propose.
3. Watch files, commands, output, and the diff.
4. Test, then accept, revise, or reject.

The agent makes production faster. It also makes supervision part of computational literacy.

Torvalds: AI is a tool



```
AI is a tool, just like other tools we use.  And it's clearly a useful one.

It may not have been that "clearly" even just a year ago, but it's no
longer in question today.

There are other questions around AI (like what the economy of it will
actually look like in the end), but "is it useful" is no longer one of
those questions. Anybody who doubts that clearly hasn't actually used
it.

Yes, it can also be a somewhat painful tool, both for maintainer
workloads and just from a "it keeps finding embarrassing bugs"
standpoint.

But the solution is not to put your head in the sand and sing "La La
La, I can't hear you" at the top of your voice like some people seem
to do.

The solution is to make sure those LLM tools _help_ maintainers
instead of just causing them pain. There's no question on that side.

We're not forcing anybody to use it, but I will very loudly ignore
people who try to argue against other people from using it.

And no, AI isn't perfect. But Christ, anybody who points to the
problems at AI had better be looking in the mirror and pointing at
themselves at the same time.

Because it's not like natural intelligence is always all that great either.
```

Screenshot supplied from Linus Torvalds's July 2026 reply in the linux-media thread "Linking Patchwork with Sashiko?" (Torvalds, 2026).

A useful tool can still move work

> *"AI is a tool, just like other tools we use. And it's clearly a useful one."*
>
> *"The solution is to make sure those LLM tools help maintainers instead of just causing them pain."*

The maintainer test

1. Does the tool reduce total work or move work to reviewers?
2. Does it surface errors early enough to correct them?
3. Can the responsible person inspect, accept, or reject the result?

For policy analysis, replace *maintainer* with *analyst*, *manager*, *or decision maker*. The coordination question is the same.

Selected excerpts from Linus Torvalds's July 2026 reply in the linux-media thread "Linking Patchwork with Sashiko?" (Torvalds, 2026).

Checkpoint 1: Why learn code?

Block 1 · minutes 16–20

A classmate says: “If an agent can write the code, I do not need to learn R.”

Decide together:

1. Which part of today’s coding target still requires the student?
2. What is one part of the work you would be comfortable delegating?

2 minutes discuss $\longrightarrow$ **2 minutes share**

Lesson map

Welcome and the question

From a model to an agent

- *The parts of the system*
- *Choose the task first*

Learning with an agent

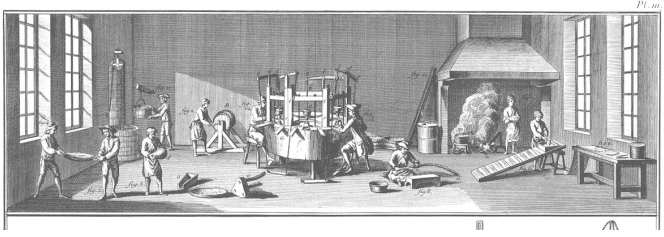
Delegating a bounded task

From Lesson 1 to Lab 1

The pin factory coordinates the division of labor

"The greatest improvements in the productive powers of labour" grew from the division of labour.

Smith (1776, Book I, Chapter I).



Diderot and d'Alembert, *Encyclopédie*, "Épinglier," plate III (1765). Digital image: ARTFL Encyclopédie Project.

Specialization raises output; coordination makes divided tasks one product.

Pedagogical framing inspired by a teaching slide shared by Luis Garicano.

Delegation has a full cost

$$\text{Net value of delegation} = \underbrace{\text{time saved} + \text{quality gain}}_{\text{benefit}} - \underbrace{\text{specification} + \text{verification} + \text{expected error cost}}_{\text{full cost}}.$$

A narrow calculation

The agent produces a WDI comparison in 20 minutes instead of 40.

The complete workflow

Writing the request and checking definitions, years, revisions, and arithmetic require another 30 minutes.

"The agent can do it" does not establish that delegation improves the workflow. Count the handoff and review, not only production (Coase, 1937).

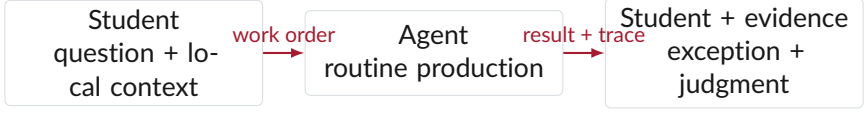
Knowledge is divided too

Hayek: knowledge is dispersed

Relevant facts do not arrive in one complete bundle. People hold different knowledge of context, time, place, and purpose (Hayek, 1945).

Garicano: route problems to knowledge

Routine problems can stay close to production; exceptional problems move to specialized problem solvers (Garicano, 2000).



The student does not disappear from the workflow. The student supplies context, recognizes exceptions, and decides whether the answer can be kept.

Checkpoint 2: Who knows what?

Block 1 · minutes 30–34

The WDI file and a ministry report give different under-five mortality values for the same country-year.

Decide together:

1. What knowledge might each source hold that the other does not?
2. What should the agent inspect, and what judgment remains with us?

2 minutes discuss $\longrightarrow$ **2 minutes share**

AI includes several kinds of systems

Term	Working definition for this course
Artificial intelligence	A broad family of systems for prediction, perception, language, planning, or action.
Large language model	A model that produces language, code, or a tool request from an instruction and context.
Chat interface	A message window around a model.
Coding agent	A model working in a loop with tools that can act inside a project.

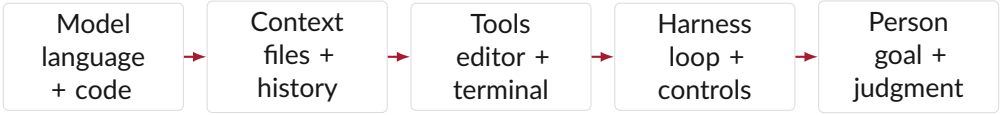
We use these terms separately because they imply different actions and different things to verify.

From conversation to action

Experience	Who executes	What we can observe
Chat	A person copies, pastes, and runs each step.	A response.
Code completion	A person works; the model suggests nearby code.	A code fragment.
Agent	The system plans, uses tools, observes, and continues.	A sequence of actions.
Coding agent	The agent operates on project files and the terminal.	Diffs, commands, logs, and artifacts.

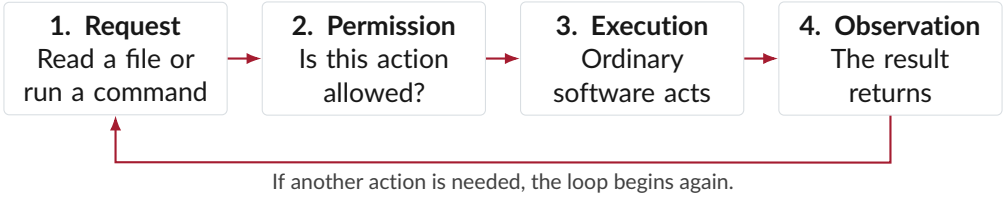
Codex and Claude Code are coding agents. Their project access allows them to read files, use tools, and return observable results (OpenAI, 2026; Anthropic, 2025).

A coding agent is a system of parts



The harness connects the model to the environment and applies permissions. We still define the objective, the standard, and the final decision.

Watch one tool-call loop



This loop helps us ask what actually happened. Evidence of an action comes from the exposed tool call, result, or log.

Four course tools, four jobs

Tool	What it does	Decision left to the analyst
R	Runs data and statistical operations.	Whether an operation represents the intended concept.
RStudio	Keeps scripts, console, files, objects, and plots visible.	Whether the code is substantively correct.
Quarto	Combines code, prose, and output in an optional notebook.	Whether the narrative follows from the evidence.
Codex	Reads, proposes, edits, runs, and checks inside a project.	Whether its output deserves our trust.

Choose the tool after the task

Need	Smallest useful system	Acceptance evidence
Calculate a summary	R script	A hand check and a reproducible result
Explain an error	Chat or agent with the error and code	Explanation matches a small test
Modify a project	Coding agent with limited access	Diff, clean run, and relevant checks
Decide a policy interpretation	Person with evidence	Definitions, assumptions, and argument

More autonomy creates more places where something can go wrong. Start with the simplest system that lets you verify the result.

Checkpoint 3: Choose the smallest system

Block 1 · minutes 46–50

Before comparing mortality across years, you need to know whether the indicator's definition or unit changed.

Decide together:

1. Which is the smallest useful system: R, chat, a coding agent, or a person?
2. What evidence would let you accept its answer?

2 minutes discuss $\longrightarrow$ **2 minutes share**

Ten-minute break

Return at minute 60.

When we return: understanding, cognitive debt, risk, and a bounded work order.

Lesson map

Welcome and the question

From a model to an agent

Learning with an agent

- *Understanding and cognitive debt*
- *A learning routine*

Delegating a bounded task

From Lesson 1 to Lab 1

Abundant output consumes scarce attention

Faster production does not remove scarcity; it moves scarcity to the attention required to choose, check, explain, and connect results to the question.

Simon argued that information abundance creates a corresponding scarcity of attention (Simon, 1971).

Understanding is the new bottleneck

Understand to verify

- Does the code run?
- Does it match the request?
- Do the checks pass?
- Should I keep it?

Understand to participate

- Why was this approach chosen?
- Which assumptions are inside it?
- What question should come next?
- How should the project change?

Adapted and paraphrased from Litt (2026). Litt argues that understanding supports approval of the current work and active participation in the next project loop.

Cognitive debt

The project can continue while the people responsible for it lose the ability to explain:

- what the system is supposed to do;
- why an important decision was made;
- which assumptions the output carries;
- how to change the work safely.

This is cognitive debt: the work advances faster than shared understanding. The cost appears later, when we need to modify, teach, or repair the project.

Concept adapted from the discussion in Litt (2026), which connects the term to Margaret-Anne Storey and Simon Willison.

External assistance is not practical judgment

Plato: reminder is not understanding

In the *Phaedrus*, writing is presented as an external aid that can preserve words while giving a learner only the appearance of wisdom.

Student test: Can I question, explain, and reconstruct what the agent produced?

Aristotle: technique is not judgment

Technē concerns making something well. *Phronēsis*, or practical wisdom, concerns deliberating well about situations that could be otherwise.

Policy test: What should be done, for whom, and under which uncertainty?

AI can extend memory and technique. Education still has to cultivate understanding and practical judgment.

Paraphrased from Plato, *Phaedrus* 274c–275b, and Aristotle, *Nicomachean Ethics*, Book VI (Plato, 1914; Aristotle, 1925).

Checkpoint 1: Is this understanding?

Block 2 · minutes 70–74

An agent produces a clean plot. The student can describe its appearance but cannot explain which rows were excluded.

Decide together:

1. What evidence of cognitive debt do you see?
2. What must the student do before keeping the plot?

2 minutes discuss $\longrightarrow$ **2 minutes share**

Education produces an artifact and a future capability

$$q_t = q(h_t, a_t),$$
$$h_{t+1} = h_t + \text{practice} + \text{AI scaffolding} - \text{displaced practice}.$$

- q_t: the quality of today's script, table, figure, or explanation.
- h_t: the student's current ability to understand and change it.
- a_t: the form and intensity of AI assistance.

Assistance can improve today's artifact while either building or weakening tomorrow's capability. The learning design determines which effect dominates.

What one coding experiment suggests

Shen and Tamkin (2026) randomly assigned AI assistance to 52 developers learning a new Python library.

Immediate product

The AI group finished about two minutes earlier. The difference was not statistically significant.

Later mastery

The AI group scored 50% on the quiz, compared with 67% without AI. The largest gap appeared in debugging.

This small study examines one skill. We use its result as a warning and avoid generalizing it to every learning setting.

AI can transmit patterns without transmitting judgment

Polanyi: knowledge can be tacit

Practice contains patterns, exceptions, and judgment that experts may use without being able to state as a complete rule (Polanyi, 1966).

Workplace evidence

AI assistance raised customer-support productivity by 14% on average and by 34% among novice and lower-skilled workers (Brynjolfsson et al., 2023).

One interpretation is that the system helped distribute patterns previously concentrated among experienced workers. The open question is whether users also learn when those patterns do not apply.

Coding-agent traces still show persistent returns to task expertise; that evidence is observational rather than causal (Hitzig et al., 2026).

Before, during, and after agent help

Moment	Student action	Useful agent support
Before	Predict the result and state the criterion.	Ask for a plan or a small example.
During	Compare the proposal with your prediction.	Ask why and test a small case.
After	Reconstruct and explain the central decision.	Ask for a quiz or an independent check.

The goal is to keep our understanding close to the speed of production (Shen and Tamkin, 2026; Litt, 2026).

The student standard

Before keeping a substantial block of agent-written code, be ready to state:

1. its purpose;
2. its input and output;
3. one important assumption;
4. one consequence of changing it;
5. one check that could reveal a mistake.

If this is difficult, ask the agent for a smaller example, an explanation with background, or a short quiz. Then try again in your own words.

Use the agent to help you understand

Explainer
Ask for the goal, background, important choices, and a literate walkthrough.

Quiz
Ask five questions that test whether you can reconstruct the central change.

Micro-example
Ask for a small object you can change and observe. Another long explanation may be harder to test.

These suggestions adapt Litt's techniques to a classroom: explanation, retrieval practice, and small environments that the learner can manipulate (Litt, 2026).

Checkpoint 2: Design a learning loop

Block 2 · minutes 84–88

An agent proposes code that joins the indicator dictionary to the WDI data.

Choose one action for each moment:

1. What will you predict *before* running it?
2. What will you inspect *during* the run?
3. What will you explain or test *afterward*?

2 minutes discuss $\longrightarrow$ **2 minutes share**

Lesson map

Welcome and the question

From a model to an agent

Learning with an agent

Delegating a bounded task

- *Risk and verifiability*
- *A work order*

From Lesson 1 to Lab 1

Capability, adoption, and impact are different claims

Capability
Can the system complete a task under specified conditions?

Adoption
Is it actually used inside a real workflow, by whom, and how often?

Impact
Does its use change productivity, quality, learning, or decisions?

90-second checkpoint
A coding agent improves on a benchmark. What has been established? Capability under those benchmark conditions—not adoption or impact.

A benchmark can guide a hypothesis about our workflow. It cannot replace evidence from our workflow (METR, 2025).

The frontier is irregular

Two tasks that look similar to us may fall on different sides of the system's capability.

Often easier to verify

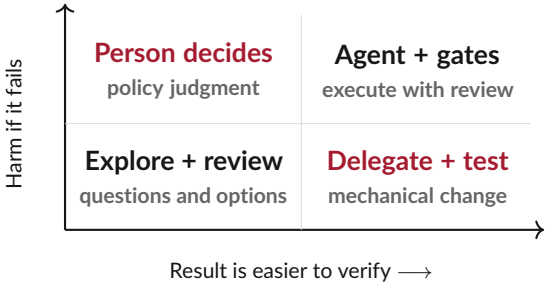
Rename variables, change a format, add a stated assertion, or reproduce a documented transformation.

Often harder to verify

Resolve an ambiguous policy concept, infer missing context, or decide whether an association justifies action.

Confidence gives us no information about where the task was on this "jagged frontier" (Dell'Acqua et al., 2026). Longer tasks also chain more decisions, so one local error can influence every later step.

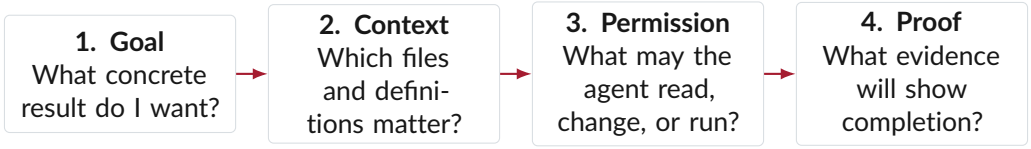
Match autonomy to risk and verifiability



Verifiability is high when a source, test, or independent comparison can expose the error.

Harm includes cost, reversibility, and consequences for people, data, and claims.

Write four things before execution



Describe the proof before delegating the task.

A weak request

Request

Use the data to show that higher income reduces child mortality.

What is missing?

- The conclusion and causal language are chosen in advance.
- The population, unit of observation, and period are undefined.
- The measures and data source are unspecified.
- The permission boundary is missing.
- Acceptable evidence and completion are undefined.

A bounded policy-analysis work order

```
Goal: Prepare an evidence inventory for this question: How is GDP per
capita associated with under-five mortality across countries in 2022?
Context: Use the shared frozen WDI data and indicator dictionary.
Focus on country, year, gdp_per_capita_ppp, and under5_mortality.
Permission: Read and summarize only. Do not edit or update the data,
retrieve new data, or fit a model.
Constraints: Use association language, flag missing values, and do not
invent definitions.
Proof: Report the unit, variable definitions, 2022 coverage, and three
checks the class should complete before analysis.
```

This work order asks for an evidence inventory, not a conclusion. It gives the class something concrete to verify before fitting a model or making a claim.

Useful context can change a decision

- Give the agent the source file directly.
- Identify the current version and the source of truth.
- Separate facts, decisions, examples, and open questions.
- Ask for references to a file, line, table, or URL.
- Leave out material that cannot change the decision.

Good context is information that can change a decision.

Give permissions according to the task

Reading

Web pages, documentation, code, public data, and metadata.

Reading is a good first step when we are still learning the project.

Action

Writing, running, deleting, publishing, sending, or buying.

Begin with less access. Increase it only when the task needs it.

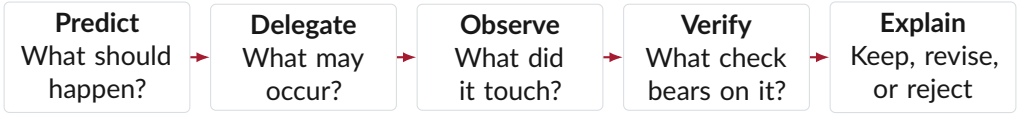
Never provide credentials, API secrets, identifiable student records, or confidential research material to an AI tool.

Define evidence for the product

Product	Minimum evidence
Code	It runs from a clean start and relevant checks pass.
Table or figure	Source, sample, units, and transformations agree.
Explanation	It matches the files and distinguishes fact from inference.
Text	Sources are attributed and no result is invented.

A good check is cheap enough to repeat and strong enough to expose a plausible failure.

Inspect the work behind the answer



"Reject" and "still unresolved" are valid outcomes. Keep a result when the available evidence supports it.

The loop echoes Wiener's account of control through communication and feedback: a system can correct only what it can observe (Wiener, 1948).

Live demonstration: the assignment

Paste both prompt slides as one request.

```
I meet the director of an East Africa energy-access program in
20 minutes. Prepare a concise briefing comparing Kenya, Rwanda,
Tanzania, and Uganda.

Go online and use official World Bank sources only. Retrieve access
to electricity (% of population), indicator EG.ELC.ACCS.ZS, through
the WDI Indicators API. Compare 2010 with each country's latest
available year; do not silently force the latest year to be the same.
```

First watch the plan and permission request. Then let the agent retrieve the data and build the analysis while the class follows its trace.

Live demonstration: deliverables and safeguards

This is the continuation of the same prompt.

```
Give me: (1) a 300-word meeting brief with one takeaway and a table
of 2010 value, latest year and value, and percentage-point change;
(2) three questions I should ask in the meeting; (3) reproducible R
code; and (4) a source note with links, definition, unit, retrieval
date, missing-data issues, and possible data revisions.

Do not impute values, make causal claims, or invent policy context.
Separate sourced facts from interpretation and cite each external
claim. Before acting, state your plan and any permission you need.
Afterward, show the checks you ran and list remaining uncertainties.
```

We will watch the retrieval, sources, code, calculations, and claim language. The answer is not prepared in advance; the verification happens live.

Lesson map

Welcome and the question

From a model to an agent

Learning with an agent

Delegating a bounded task

From Lesson 1 to Lab 1

What we keep

What we learned in Lesson 1

- Choose the task before choosing the tool.
- Keep understanding close to the speed of production.
- Match agent autonomy to risk and verifiability.
- Define completion evidence before execution.
- Keep the question, interpretation, and responsibility with the person.

We will reuse this supervision structure during the next lessons as the code changes.

Lab 1 is an installation clinic

The data exercise begins after every laptop passes the readiness gate.

A laptop is ready when the setup check is green. A red result leaves with the exact error, one repair step, and a named person for follow-up.

The readiness gate

Open the Terminal tab in RStudio and run:

```
Rscript code/check_setup.R --codex-confirmed
```

The checker verifies:

- the course project and the R version;
- required R packages;
- Quarto;
- the frozen teaching data;
- Codex access to the same project.

If your readiness gate is green

Spend 8–10 minutes using Codex in read-only mode while the room continues repairing setup:

```
Read this project without changing files or running commands. Locate
(1) the course start instructions, (2) the frozen WDI data, and
(3) the indicator dictionary. Return each exact file path and one
sentence about what the file contains. Separate statements you read
from the files from inferences you made.
```

Verify every path manually in RStudio's Files pane. Then write one sentence: What did the agent know from the files, and what did it infer?

Lab 1 route

Minutes	Focus	What we need to see
00–05	**Triage**	Green, yellow, or red output.
05–15	**Project + R**	Extracted folder, open project, current R.
15–25	**Packages**	Install only what the checker reports.
25–35	**Quarto**	Install, restart RStudio, and rerun.
35–50	**Data + Codex**	Frozen file and project access.
50–60	**Final gate**	Green result or an owned repair plan.
After green	**8–10 min proof**	Locate three files and verify every path.

References I

Anthropic, "Claude 3.7 Sonnet and Claude Code," Product announcement February 2025.

Aristotle, "Nicomachean Ethics, Book VI," The Internet Classics Archive 1925.

Brynjolfsson, Erik, Danielle Li, and Lindsey R. Raymond, "Generative AI at Work," Working Paper 31161, National Bureau of Economic Research 2023.

Coase, R. H., "The Nature of the Firm," *Economica*, 1937, *4* (16), 386–405.

Dell'Acqua, Fabrizio, Edward McFowland, Ethan R. Mollick, Hila Lifshitz-Assaf, Katherine C. Kellogg, Saran Rajendran, Lisa Krayer, François Candelon, and Karim R. Lakhani, "Navigating the Jagged Technological Frontier: Field Experimental Evidence of the Effects of AI on Knowledge Worker Productivity and Quality," *Organization Science*, 2026, *37* (2), 403–423.

References II

Garicano, Luis, "Hierarchies and the Organization of Knowledge in Production," *Journal of Political Economy*, 2000, *108* (5), 874–904.

Hayek, Friedrich A., "The Use of Knowledge in Society," *American Economic Review*, 1945, *35* (4), 519–530.

Hitzig, Zoe, Maxim Massenkoff, Eva Lyubich, Shaoyi Zhang, Ryan Heller, and Peter McCrory, "Agentic Coding and Persistent Returns to Expertise," Anthropic Economic Research June 2026.

Litt, Geoffrey, "Understanding Is the New Bottleneck," Essay and conference talk July 2026.

METR, "Measuring AI Ability to Complete Long Tasks," Research report March 2025.

Ng, Andrew, "Skills for Students with AI Goals," Stanford eCorner interview transcript October 2023.

References III

_ , “Building AI Startups: Important Technical AI Trends,” Presentation 2025.

OpenAI, “Codex Use Cases,” OpenAI Developers documentation 2026.

Plato, *Phaedrus*, Cambridge, MA: Harvard University Press, 1914. Sections 274c–275b.

Polanyi, Michael, *The Tacit Dimension*, Garden City, NY: Doubleday, 1966.

Shen, Judy Hanwen and Alex Tamkin, “How AI Impacts Skill Formation,” arXiv:2601.20245 2026.

Simon, Herbert A., “Designing Organizations for an Information-Rich World,” in Martin Greenberger, ed., *Computers, Communications, and the Public Interest*, Baltimore: Johns Hopkins Press, 1971, pp. 37–72.

Smith, Adam, *An Inquiry into the Nature and Causes of the Wealth of Nations*, London: W. Strahan and T. Cadell, 1776. Book I, Chapter I.

References IV

Torvalds, Linus, “Re: Linking Patchwork with Sashiko?,” linux-media mailing list reply July 2026. Reply posted July 14, 2026; archived by the Linux kernel mailing-list archive.

Wiener, Norbert, *Cybernetics: Or Control and Communication in the Animal and the Machine*, New York: John Wiley & Sons, 1948.