# Getting Research Done with ChatGPT Plus and Modern AI

## A one hour training for quantitative social sciences

Juan Pablo Posada Aparicio

Institute for Replication, University of Ottawa

October 2025

# Session goals

- Understand what large language models are, at a practical level.
- Learn ChatGPT Plus capabilities that matter for research.
- See how to plug AI into literature, coding, data, writing, and review.
- Get copy paste demo prompts you can run today.
- Leave with a short checklist to use AI responsibly and reproducibly.

# Roadmap, about 60 minutes

- 5 minutes. LLMs in plain language and prompting basics.
- 10 minutes. What is in ChatGPT Plus today.
- 30 minutes. Research workflow playbook, literature to code to writing.
- 10 minutes. Deep Research and Agent Mode.
- 5 minutes. Guardrails, reproducibility, and Q and A.
- 5 minutes. Codex, quick version, at the very end.

# What is a large language model

- A model that predicts the next token in context. It turns that simple mechanism into useful tasks by following instructions.
- Strengths. Language understanding, code generation, pattern completion, transformation.
- Weaknesses. Can be wrong or overconfident. May miss recent facts without browsing or tools. Needs clear instructions and constraints.
- Your edge. Provide context, constraints, and examples. Iterate and verify.

# Prompting that actually works

- Set role and objective. Then give the smallest possible working example.
- Ask for structured outputs you can reuse. For example markdown, csv, json, or a full script.
- Use iterate and verify. Ask for one improvement at a time.

# ChatGPT Plus, the useful bits

- **Advanced Data Analysis**. Write and run Python inside the chat. Upload files. Produce figures and tables.
- **Browsing**. Read the web for current facts when you ask for it.
- **Deep Research**. Multi step web research that collects and organizes findings before answering.
- **Custom GPTs**. Create specialized assistants with your instructions and files.
- **Agent Mode**. Let ChatGPT plan and execute multi step tasks with tools, while you supervise.

# Literature discovery and synthesis

- Ask for a scaffold first. Key theories, main debates, measurement, common datasets.
- Then ask for concrete items. Seminal papers, recent systematic reviews, and data portals.
- Use browsing or Deep Research when you need current sources.

# Demo prompts, literature

**Demo.** Copy and run one prompt.

```
You are my research assistant. Topic: the effect of community radios on
forced displacement in Colombia. Output a literature scaffold with
sections: theory, identification strategies seen in the field,
measurement choices, typical datasets, open questions. Limit to 10 bullets
```

**Demo.** With browsing or Deep Research on.

```
Find 8 recent and credible sources on community radios and violence.
Return a table with columns: citation in APA, year, identification
strategy, sample, main finding, link. Put uncertainties in a final note.
```

# Data access and documentation

- Ask for a data dictionary plan. Variable names, units, refresh cadence, and owners.
- Have the model draft a README template for each dataset.
- Convert messy documentation into a clean table you can ship.

# Demo prompts, data documentation

**Demo.** Dictionary and README.

```
Create a minimal data dictionary schema for a panel of municipalities by
month. Include: name, label, type, units, allowed values, missing policy,
owner, refresh cadence, source URL.
```

**Demo.** Turn messy text into a table.

```
Extract a clean csv with columns variable, definition, unit, period
from the following pasted notes. If needed, ask one clarifying question.
```

# Cleaning and analysis with Advanced Data Analysis

- Upload a small sample first. Ask for a quick EDA, missingness, and obvious anomalies.
- Ask for clean, reproducible code with comments and a final function.
- Iterate to reach publication grade figures.

# Demo prompts, cleaning and EDA

**Demo.** Quick EDA on upload.

```
Run a 10 minute EDA on the uploaded csv. Show schema, missingness,
basic summaries, and 3 plots that reveal structure. Save code to a .py
file I can download.
```

**Demo.** Robust cleaning function.

```
Write a Python function clean_df(df) that trims whitespace, parses dates,
standardizes municipality names using a provided mapping, and validates
unique keys (muni_id, month). Return cleaned df and a log of fixes.
```

# Interpreting results and writing

- Paste tables or model summaries and ask for a plain English interpretation.
- Ask for a short abstract or a paragraph that highlights identification and magnitude.
- Use a style checklist. Clarity, claims tied to evidence, and limits.

# Demo prompts, interpretation and writing

**Demo.** Interpret a regression.

```
Given this regression table, explain the sign, magnitude, and precision
of key coefficients. Avoid causal language unless identification is explic
```

**Demo.** Draft a concise abstract.

```
Draft a 150 word abstract given these bullet points. Emphasize design,
data, and main effect with uncertainty. Return a single paragraph.
```

# Peer review and reproducibility

- Ask the model to act as a critical but fair reviewer.
- Have it produce a replication log template, aligned with your PAP.
- Use it to standardize robustness checks and pre analysis notes.

# Demo prompts, review and logs

**Demo.** Act as a reviewer.

```
Act as a critical but fair referee for the following methods section.
List 5 specific concerns and 3 concrete suggestions to improve validity.
```

**Demo.** Replication log template.

```
Create a csv template for a replication log with columns for coding
errors, computational reproducibility, two robustness checks, and notes.
Keep column names short and consistent.
```

# Deep Research, when to use it

- Use when you want structured web research with sources collected first.
- Ask for constraints. Scope, credibility criteria, and deliverable format.
- Expect longer runs. Use it for scoping reviews or policy scans.

# Demo prompts, Deep Research

**Demo.** Scoping review pattern.

```
Use Deep Research to scan policy reports and peer reviewed papers on
municipal radio and governance outcomes in Latin America after 2010.
Return a matrix of findings by country and outcome with links.
```

# Agent Mode, supervised autonomy

- Plan and execute multi step tasks across tools, under your supervision.
- Good for browse, download data, analyze, and assemble a draft.
- Start with small, auditable tasks. Keep humans in the loop.

# Demo prompts, Agent Mode

**Demo.** End to end micro task.

```
Plan and run a short workflow. Find two official data sources on radio
coverage by municipality in Colombia, download small samples, summarize
coverage over time for Antioquia, and draft a 5 bullet summary with links.
Ask before any login or large downloads.
```

# Responsible and reproducible use

- Track provenance. Record what was prompted, what data was used, and when.
- Never paste sensitive data unless policy allows it. Prefer local tools for secrets.
- Verify everything that affects results or claims.
- Save outputs in plain text files. Scripts, logs, and markdown notes.
- Be clear about where AI helped, in acknowledgments or methods.

# A tiny checklist

1. Did I provide sufficient context and constraints.
2. Did I verify claims and numbers against sources.
3. Are scripts and logs saved with dates and versions.
4. Can a colleague rerun this without me.
5. Is sensitive data handled according to policy.

# Prompt bank, quick copy

```
Rewrite this paragraph for clarity and precision. Keep technical terms.
Return two variants and a diff style list of changes.
```

```
You are a code reviewer. Read this Stata do file and list 5 risks.
Then propose 3 refactors that reduce runtime or errors.
```

```
Given this figure and caption, propose one alternative visualization
and write the exact matplotlib or ggplot code.
```

# Codex, what it is and when to use it

- Codex is OpenAI's coding agent for working with your repos.
- Use chat for questions and small snippets. Use Codex when you want edits in files and simple automations.
- Keep your repo under git. Review diffs. You stay in control.

# Codex CLI, install and start

```
# macOS or Linux
npm install -g @openai/codex
# or
brew install codex

# run
codex
```

*On Windows, prefer WSL.*

# Two demos you can run

**Demo.** Summarize a csv.

```
Read data.csv here. Report column types, missingness,
and five key stats per numeric column. Save a markdown summary.
```

**Demo.** Explain code you did not write.

```
Explain what analysis.py does, step by step, and list two risks.
```

# Tiny command cheat sheet

```
codex "explain this codebase"
codex exec "fix the failing test"
/model gpt-5-codex
/approvals
```

# Q and A

Thank you.