APEC 8221: Programming for Econometrics

Fall 2025, 2 credits, In-Person

Course description

This 7-week, 2-credit course provides a rigorous, hands-on introduction to modern computational tools for applied econometrics research. The curriculum moves beyond point-and-click software to develop a professional, reproducible research workflow using R—a powerful, flexible, and open-source programming language and environment specifically designed for statistical computing and graphics. Students will learn to manage projects with RStudio and Git/GitHub, author dynamic documents with Quarto, acquire data from diverse sources (APIs, web scraping), and perform complex data manipulation using `tidyverse`. Emphasis is placed on producing publication-quality outputs, including figures, tables, and choropleth maps for spatial data analysis. The course culminates in a final project that synthesizes these skills by replicating and extending a core result from an economics research paper. Upon completion, students will be equipped to conduct independent, computationally sophisticated, and reproducible research.

Instructor: Ali Joglekar
Research Scientist, GEMS Informatics
Adjunct Assistant Professor, APEC
248G Ruttan Hall | joglekar@umn.edu

Schedule: Tuesday and Thursday, 11:45 A.M.–1:00 P.M.
Ruttan Hall 135B
September 2 – October 21, 2025

Office hours: Thursdays, 10:30–11:30 A.M., Ruttan 248G
Appointments by request are welcomed

Prerequisites: A graduate-level course in applied econometrics covering causal inference techniques (e.g., IV, fixed effects, DiD), or permission from the instructor.

This publication/material is available in alternative formats upon request. Please contact Jenna Mead (jmead@umn.edu, 612-625-3777, 231H Ruttan Hall), the Graduate Program Coordinator for the Department of Applied Economics, if needed.

Learning objectives

Through active participation in this course, you will be able to:

- **Manage a Reproducible Research Workflow**: Independently manage a complete research project using R Projects for organization, Quarto for dynamic document generation, and Git/GitHub for version control, adhering to best practices for commit history and documentation.
- **Programmatically Acquire and Assemble Data**: Write R code to automatically acquire data by querying REST APIs directly, scraping structured data from HTML web pages, and reading multiple data files.
- **Master Advanced Data Manipulation**: Restructure, join, and clean complex, messy datasets using `dplyr` for readable data pipelines. This includes cleaning text data using regular expressions.
- **Create Publication-Quality Outputs**: Generate publication-ready visualizations using `ggplot2` and produce formatted regression and summary tables using packages like modelsummary and `gt`.
- **Produce Maps and Analyze Spatial Data**: Create choropleth maps by joining attribute data to spatial vector data contained in `sf` objects and visualize the results with `ggplot2`.
- **Develop Modular and Efficient Code**: Author custom functions to eliminate code redundancy and use iterative structures (e.g., loops, `purrr` functionals) to automate repetitive analytical tasks.
- **Synthesize Skills for Empirical Replication**: Integrate all learned skills to execute a full research workflow, from data acquisition to final output, by replicating the primary findings of a published economics paper.

Course materials

Required

- Wickham, H., Çetinkaya-Rundel, M., and G. Grolemund (2023). *R for Data Science: Import, Tidy, Transform, Visualize, and Model* (2nd ed.). O'Reilly Media, Inc. Free online access
- Gillespie, C., and Lovelace, R. (2021). *Efficient R Programming*. O'Reilly Media, Inc. Free online access

Supplementary

- Cunningham, S. (2021). *Causal Inference: The Mixtape*. Yale University Press. Free online access
- Heiss, F. (2020). Using R for Introductory Econometrics. 2nd Ed. Free online access. (Computer code used in this book can be accessed via GitHub)

Technology requirements

- Laptop computer capable of running R and RStudio (required for all class sessions)
- R (version ≥ 4.4.0) and RStudio (version ≥ 2024.04) (free software—installation instructions will be provided)
- Git and GitHub account (free—setup instructions provided in first assignment)
- Reliable internet connection for accessing online materials and collaborative work

Course schedule

- **R4DS:** Wickham et al. (2023), *R for Data Science*
- **ERP:** Gillespie and Lovelace (2021), *Efficient R Programming*

Wk	Lecture Topic		Required Reading	Suggested Reading	Assignment
Wk 1	1	Modern Research Workflow	R4DS: Introduction, Ch 2 (basics)	ERP: Ch 2 (setup) RStudio IDE cheatsheet Base R cheatsheet	
	2	Literate Programming & Version Control	R4DS: Ch 4 (style), Ch 6 (projects), & Ch 28 (Quarto) ERP: Ch 9 (collaboration)	Happy Git with R Quarto cheatsheet R4DS: Ch Ch 8 (help) ERP: Ch 10 (learning)	*A0 posted*
Wk 2	3	Data Manipulation (`dplyr`)	R4DS: Ch 3 (transform), Ch 7 (import), Ch 27.2 – 27.3 (selecting elements)	R4DS: Ch 12 (logical), Ch 13 (numbers), Ch 16 (factors), Ch 17 (dates) `dplyr` cheatsheet `readr` cheatsheet `forcats` cheatsheet	**A0 due**

Wk	Lecture Topic		Required Reading	Suggested Reading	Assignment
	4	Intro to Visualization	R4DS: Ch 1 (visual)	`ggplot2` tutorial `ggplot2` cheatsheet	*A1 posted*
Wk 3	5	Automation – Functions & Iteration	R4DS: Ch 25 (functions), Ch 26 (iteration), Ch 27.4 (`apply`), Ch 27.5 (`for`)	ERP: Ch 3 (programming) vector & lists tutorial `purrr::map()` tutorial `purrr` cheatsheet	
	6	Data Acquisition I (APIs)	An intro to APIs (Chs 1–4)	R4DS: Ch 23 (hierarchical) `httr2` get started `httr2` wrapping APIs	*A2 posted* **A1 due**
Wk 4	7	Reshaping and Joining Data	R4DS: Ch 5 (tidy), Ch 19 (joins)	R4DS: Ch 18 (missing) ERP: Ch 6.2–6.5 (carpentry) `tidyr` cheatsheet	
	8	Web Scraping	R4DS: Ch 24 (scraping)	`rvest` tutorial	*A3 posted* **A2 due**
Wk 5	9	Text Cleaning with Regular Expressions	R4DS: Ch 14 (strings), Ch 15 (regex)	`stringr` cheatsheet Regex101.com (for practice)	
	10	Building Publication-Ready Tables	Arel-Bundock, V. (2022). `modelsummary`: Data & model summaries in R. *Journal of Statistical Software, 103*(1), 1-23.	`modelsummary` getting started vignette `modelsummary` model summaries vignette `gt` vignette	*A4 posted* *Final project posted* **A3 due**
Wk 6	11	Advanced Visualization	R4DS: Ch 9 (layers), Ch 10 (communication)	R4DS: Ch 10 (explore) `patchwork` tutorial	
	12	Introduction to Spatial Data (`sf`)	Pebesma, E. (2018). Simple Features for R: Standardized support for	`sf` vignettes ggplot2-book.org: Ch 6	*A5 posted* **A4 due**

Wk	Lecture Topic		Required Reading	Suggested Reading	Assignment
			spatial vector data. *The R Journal, 10*(1): 439–446.	(maps)	
Wk 7	13	High-Performance Data Wrangling (data.table)	`data.table` intro & vignettes	ERP: Ch 6.7 (carpentry) `data.table` cheatsheet `data.table` vs `base` vs `dplyr` post	
	14	A Tour of Advanced Topics & Final Project Workshop			**A5 due**
Wk 8					**Final project due**

What I expect from you

- Come to class prepared having completed the assigned readings
- Actively participate in hands-on exercises and team activities
- Submit assignments that reflect your own understanding, properly citing all sources
- Communicate proactively if you're struggling with material or facing challenges

Student workload and time commitment

This is an accelerated, 2-credit, PhD-level course condensed into a 7-week term. The pace and depth of the material require a significant time investment.

University policy defines one (undergraduate) credit as equivalent to approximately three hours of student work per week. However, the quantitative, hands-on nature of this course necessitates a workload well above that baseline. To succeed, you should expect to dedicate 13 to 17 hours per week to this course. This time is an estimate for an average student and will be spent on the following activities:

- **Instruction (2.5 hours):** Attending and actively participating in lectures and live-coding sessions.

- **Readings (3-4 hours):** Carefully studying the assigned book chapters and technical documentation to understand the conceptual foundations of the tools we use.
- **Assignments & Project (8-10 hours):** Applying concepts by completing weekly programming assignments and making steady progress on the final replication project. This includes writing code, debugging, and interpreting results.

This course is designed to be an immersive experience that equips you with foundational skills for a career in modern empirical research. The substantial time commitment reflects the high value of these practical skills. Please plan your quarter accordingly to ensure you have adequate time to engage fully with the material.

What you can expect from me

- I will send an announcement if this syllabus or course schedule changes
- I will welcome you to my office hours and give you my full attention
- I will provide timely feedback on assignments and be available to help you succeed
- Statement on pronouns: I will gladly honor your request to address you by your chosen name and pronouns. Please advise me of this at any point in the semester so that I may make appropriate changes to my records.

What about AI?

I embrace the use of AI tools as learning aids in this course, similar to how I encourage peer collaboration. AI tools like Gemini, ChatGPT, GitHub Copilot, and Claude can be valuable for understanding code, debugging, and exploring programming concepts. However, all non-original work must be properly cited, including:

- AI assistance (specify which tool and how it was used)
- Peer collaboration (acknowledge classmates who helped)
- Online resources and tutorials

Important: AI and peer learning should enhance your understanding, not bypass the learning process. Submitting work you don't understand violates academic integrity. You should be able to explain your code and approach. Consider AI as a tutor—use it to learn, but ensure the final work represents your own understanding.

Assignments

The course includes one simple set-up assignment, five weekly assignments, and one final project. As part of your reproducible workflow, you will commit your assignments to a private GitHub repository (shared with me, the instructor) as fully reproducible, well-commented Quarto (.qmd) documents that knits to a PDF document. Quarto templates will be provided for the earlier assignments to establish a standard format; subsequent assignments will require students to structure their own documents.

Assignment	Points	Grade Share	Description	Approximate Due Date
Assignment 0	5 pts	6.7%	Setting Up Your Workspace	Sep 8
Assignment 1	10 pts	13.3%	Tidyverse Fundamentals	Sep 20
Assignment 2	10 pts	13.3%	Automation and API Data Acquisition	Sep 27
Assignment 3	10 pts	13.3%	Web Scraping and Text Wrangling	Oct 4
Assignment 4	10 pts	13.3%	Advanced Wrangling and Publication Tables	Oct 11
Assignment 5	10 pts	13.3%	Visualization and Spatial Analysis	Oct 18
Final Project	20 pts	26.7%	Replicating and Extending a Core Result from a Published Paper	Oct 25
Total	**75 pts**			

Note: Assignment dates and lecture topics are subject to adjustment based on the flow of the class.

Assignment requirements

All assignments must be submitted as a PDF file via Canvas which contains:

1. The URL to your GitHub repository containing your Quarto documents for review
2. Proper citations for all assistance (AI, peers, online resources)
3. Brief reflection (2-3 sentences) explaining your approach or key challenges

To prevent academic misconduct, assignments will include code commenting requirements and process documentation through GitHub commits showing your work progression.

Assessment criteria

Assignments are evaluated based on both functionality and code quality. I look for:

- **Correctness:** Does your code produce the expected results?
- **Clarity:** Is your code well-organized and commented?
- **Understanding:** Can you explain your approach and decisions?
- **Reproducibility:** Does your code run without modification?
- **Academic integrity:** Are all sources properly cited?

Grading scale

Grade	Percent	Points
A	[93 - 100]	[69.8 - 75]
A-	[90 - 93)	[67.5 - 69.8)
B+	[87 - 90)	[65.3 - 67.5)
B	[83 - 87)	[62.3 - 65.3)
B-	[80 - 83)	[60 - 62.3)
C	[70 - 80)	[52.5 - 60)
D	[60 - 67)	[45 - 52.5)
F	[0 - 60)	[0 - 45)

Deadlines and Late Work

All assignments must be submitted on Canvas before 8:00 AM on the due date.

I understand that graduate study involves balancing many commitments. To provide some flexibility, each student will have a total of three "grace days" to use throughout the semester on any assignment without penalty. You may use these days one at a time or all at once.

Once your grace days are exhausted, any assignment submitted after the deadline will be penalized by 10% per day. Please be aware that no work will be accepted more than five days after the due date. Plan your use of grace days accordingly.

If a legitimate reason prohibits you from finishing an assignment on time, please contact me in advance and I will make appropriate accommodations. Legitimate circumstances include verified illness, participation in intercollegiate athletic events, subpoenas, jury duty, military service, bereavement, and religious observances. For complete information, please see Administrative Policy: Excused Absences and Makeup Work.

Attendance

Class attendance is expected and essential for success. This hands-on course relies heavily on in-class programming exercises and team activities that cannot be easily made up. While you won't be graded on attendance, material covered in class is fair game for assignments. If

you must miss class, lecture materials will be available online, but you're responsible for catching up on missed activities.

Extra credit

There are no opportunities for extra credit in this course.

Disability accommodations

I am committed to creating an inclusive learning environment within my course. Inclusivity and accessibility are ongoing community processes, and I hope that you as a member of our class share my commitment to creating a classroom experience that fosters belonging. Please contact me immediately if you become concerned—for any reason—about your capacity to fully participate in our course due to the structure of the course, activities, or assignments. If you work with the Disability Resource Center (DRC), please notify me as soon as possible so that we can discuss access (https://disability.umn.edu/, drc@umn.edu). If you do not work with the DRC, but know that access barriers may arise (due to undiagnosed health conditions, mental health, learning style, life circumstances, etc.), please reach out to me as soon as possible so that we can work together to support your learning. I welcome the conversation.

You may also be interested in the complete Board of Regents Policy: Disability Services.

University policies

Please review the following University of Minnesota policies as they apply to your experiences as a student. University policies are taken from Recommended Policy Statements for Syllabi.

Scholastic Dishonesty

As a university student, you are expected to complete your own academic work and properly cite all sources used. Scholastic dishonesty—including plagiarism, unauthorized collaboration beyond what is permitted, misuse of course materials, or falsifying records—can result in serious consequences. For programming assignments, submitting substantially similar code without proper attribution constitutes plagiarism. When collaborating with peers or using AI assistance, you must cite these sources and ensure you understand the code you submit. If you're unsure about what is allowed, consult me for guidance. Additional resources about academic integrity can be found through the Office for Community Standards.

Student Conduct Code

The University seeks an environment that promotes academic achievement and integrity, that is protective of free inquiry, and that serves the educational mission of the University. To support this environment, students are expected to follow the Student Conduct Code, which requires a community that is free from violence, threats, and intimidation; that is respectful of the rights, opportunities, and welfare of students, faculty, staff, and guests of the University; and that does not threaten the physical or mental health or safety of members of the University community.

Sexual Harassment, Sexual Assault, Stalking, and Relationship Violence

The University prohibits sexual misconduct and encourages those affected to seek support, explore reporting options, and contact a confidential resource on campus if desiring to speak confidentially with someone. Instructors are required to share reports of possible sexual misconduct with the campus Title IX office. For confidential help, to report sexual misconduct, or to access more information, contact your campus Title IX office or consult the Board of Regents policy on sexual misconduct.

Diversity, Equity, Inclusion, and Equal Opportunity

The University provides equal access to and opportunity in its programs and facilities, without regard to race, color, creed, religion, national origin, gender, age, marital status, familial status, disability, public assistance status, membership or activity in a local commission created for the purpose of dealing with discrimination, veteran status, sexual orientation, gender identity, or gender expression. For more information, please consult Board of Regents Policy: Diversity, Equity, Inclusion, and Equal Opportunity.

Discrimination

All University members are prohibited from engaging in, or assisting or abetting another's engagement in, discrimination and related retaliation (collectively "prohibited conduct"). The University of Minnesota (the "University") will take prompt and effective steps intended to end prohibited conduct; prevent its recurrence; and, as appropriate, remedy its effects. For more information, consult Administration Policy: Discrimination.

Mental Health and Stress Management

Students may face challenges like anxiety, relationship issues, substance use, difficulty concentrating, or lack of motivation that can affect academic and daily life. Confidential

mental health services are available through the Student Mental Health Website. As an instructor, I care about student wellbeing, and if concerns arise, I may consult with campus support offices to help connect students with appropriate resources.

Academic Freedom and Responsibility

Academic freedom is central to the University, allowing students to explore, discuss, and research relevant topics within a course's scope. With this freedom comes the responsibility to engage critically, respect differing views, develop the capacity for critical judgment, engage in a sustained and independent search for truth, and meet course requirements. Concerns about academic freedom are taken seriously—students can reach out to me, department leadership, advisors, or the Office of the Provost for support. Read more about Academic Freedom and Responsibility.

Use of Personal Electronic Devices in the Classroom

Laptops are required for this course as we'll be programming during class time. Please use devices respectfully and avoid non-course related activities during instruction. For complete information, please review the Administrative Policy: Teaching and Learning: Student Responsibilities.

Appropriate Student Use of Class Notes and Course Materials

Taking notes helps you record, integrate, and connect information as part of your learning. Sharing notes beyond the class or for compensation violates academic norms and undermines the instructor's intellectual work. For more details, see the Administrative Policy: Teaching and Learning: Student Responsibilities.