SYRACUSE UNIVERSITY, FALL 2026 SYLLABUS

MAT 422, Statistical Computing

General Information

MySlice listing: MAT 422 Section/course ID/Days/Times/Room: TBA
Instructor: Thomas John, Ph.D., Carnegie-313D, email: thjohn@syr.edu, Office Hours: TBA.

Course Catalogue Description:

Characteristics of data, data preparation, and reformatting, accessing data from various sources, programming principles for statistical computing, data summary and visualization, statistical study design aspects of sampling and randomization, introduction to statistical simulation.
Prereq: MAT 222 or permission of instructor

Shared Competencies:

MAT 422 supports two of Syracuse University's Shared Competencies:

- Scientific Inquiry and Research Skills
- Information Literacy and Technological Agility

Course Overview:

Our goal for the course and this semester will be for you to become familiar with dealing with aspects of statistical computing. These aspects will have the following major themes:

(1) data preparation and/or reading data from delimited files; (2) organizing data by recoding, relabeling, stacking, and unstacking; (3) basic database queries; (4) basic programming concepts such as functions and loops; (5) data in matrix and multidimensional array forms; (6) data visualization; (7) data summary and basic statistical analyses; (8) basic sampling size determinations; (9) basic randomization schemes; and (10) basic statistical simulation.

Pre-Course Preparation (Week 0 Module)

Due: Before the first day of class (August 25, 2026)
Before our first-class meeting, all students must complete the Week 0 Preparation Module on Blackboard. This self-paced module ensures everyone starts with basic technical setup completed.

Module Contents:
- Installing R and RStudio (or accessing via campus computers)
- Overview of the RStudio interface
- Running your first R commands
- Introduction to R Commander as an alternative interface
- Completion quiz (pass/fail, not counted in final grade)

Purpose: This module separates basic computer literacy from statistical computing concepts, allowing us to focus on meaningful content from Day 1. If you encounter technical difficulties completing this module, please contact me or visit ITS before the first class.

For students with prior R experience: You may skip directly to the completion quiz if you are already comfortable with R/RStudio.

Text:

There is no official textbook for this course. We will select from abundant (reputable) resources available online. Links to those and additional resources will be posted on the course blackboard page.

Exam Dates:

- Semester Exam 1: Tuesday, TBA
- Semester Exam 2: Tuesday, TBA
- **CUMULATIVE FINAL EXAM:**
  - **TBA**
  - for class periods: TTH TBA

Syllabus is to be updated with specific exam dates when the semester specifics are decided.

Special Notes on the Final Exam:

- The final exam will be comprehensive.
- **All students must take the cumulative final exam at their scheduled time.**
- There will be no exceptions and so **you should not make any other plans on (or plan to leave campus before) before the scheduled final exam date.**

Grading:

Your final grade will be based on the following components:

Semester Exam 1	15%
Semester Exam 2	15%
Cumulative Final Exam	15%
Skills Checkpoints (6 total)	15%
Projects (2 total)	25%
Homework/Engagement	15%

Final letter grades will be given according to the following scale:
A (93-100); A- (90-92); B+ (87-89); B (83-86); B- (80-82);
C+ (77-79); C (73-76); C- (70-72); D (60-69); F (0-59)

Learning Outcomes: By the end of this course, successful students will be able to:

1. Execute and explain R code using either R Commander or RStudio to import, clean, and summarize datasets with at least 1000 observations and 10 variables
2. Implement iterative algorithms (loops, conditional logic, functions) in R to solve numerical problems, with line-by-line documentation explaining the code's logic
3. Apply appropriate statistical methods by identifying variable types (categorical vs. quantitative) and selecting corresponding analyses (t-tests, ANOVA, chi-square, regression)
4. Create effective data visualizations (histograms, boxplots, scatterplots) appropriate to the data type and analytical goal, with proper labels and interpretations
5. Generate reproducible statistical reports using R Markdown that integrate code, visualizations, statistical output, and written interpretation for non-technical audiences
6. Debug and modify existing code by reading through R scripts, identifying errors, and making appropriate corrections

Course Structure: Progressive Skill Development -- This course builds computational skills incrementally through four components:

1. Live Demonstrations (In-Class)
   Each class includes live coding demonstrations with extensively commented code. These demonstrate problem-solving processes and computational thinking. You are not required to code along during class but should observe carefully and take notes on the logic and approach.
2. Skills Checkpoints (15% of grade)

Six short coding assignments (due Weeks 3, 5, 7, 9, 11, 13) provide scaffolded practice with immediate feedback:
- Checkpoints 1-2: Using built-in R functions for probability and statistics
- Checkpoints 3-4: Writing and modifying custom functions
- Checkpoints 5-6: Implementing iterative algorithms (loops, convergence)

These are low-stakes assessments (2.5% each) designed to normalize errors and provide practice. They build progressively toward the skills needed for major projects.

3. Homework Assignments (15% of grade)
   Weekly homework through WeBWorK provides targeted practice with specific R functions and statistical concepts. Submit R code with sufficient comments as PDF files.
4. Integration Projects (25% of grade)
   Two major projects require complete data analysis workflows:
   - Project 1 (Week 6): Conduct a statistical hypothesis test with full report
   - Project 2 (Week 11): Implement numerical algorithms and compare methods

   Projects assess your ability to integrate multiple skills into coherent analyses with professional-quality reports in R Markdown.

Learning Pathways: RCommander vs. RStudio -- Students may choose between two approaches for completing coursework:

Pathway A: R Commander (RCmdr) - Recommended for beginners
- Graphical user interface with menus and buttons
- Generates R code automatically
- Requirement: You must understand and explain the generated code
- Best for: Students with limited or no programming experience
- Transition goal: Aim to move toward RStudio by Week 8 (after Exam 1)

Pathway B: RStudio - Traditional coding environment
- Write R scripts from scratch
- Full control and flexibility
- Best for: Students comfortable with programming concepts
- Used for: Advanced assignments (particularly algorithmic implementations)

Important notes:
- Both pathways lead to the same learning outcomes
- You may switch pathways at any time
- Some assignments (particularly algorithmic implementations) will require RStudio
- Office hours and resources support both approaches

Getting Help and Building Skills

If you are struggling or feeling lost:
- This is normal, especially if you are new to programming
- Attend office hours early and often—do not wait until you are completely stuck
- Use the Week 0 module and recommended online resources
- Ask questions in class about concepts (save debugging for office hours)
- Start assignments early so you have time to get help before deadlines

Available resources:
- Instructor office hours (TBA)
- ITS computer labs and help desk (help@syr.edu, 315.443.2677)
- Online R documentation and tutorials (links on Blackboard)
- Classmate study groups (encouraged for understanding concepts)

Academic integrity in collaborative work:

You may discuss homework problems and concepts with classmates, but each student must write and submit their own code independently. Understanding comes from doing, not from copying.

Makeup Policy

No makeups or extensions will be granted for any assignments or exams except in cases of documented emergencies verified by:

- Medical documentation from a physician, OR
- Student Outreach and Support absence notification (see https://experience.syracuse.edu/student-outreach/student-support/absence-notifications/)

If you miss an assessment due to documented emergency:

- Your grade will be calculated based on performance in corresponding sections of remaining exams
- This requires that your absence covers the entire assignment period, not just the due date

Professional responsibility perspective:

Data analysts work under firm deadlines from clients, teams, and regulatory requirements. Learning to plan ahead and submit work on time is an essential professional skill. Technical issues ("internet down," "laptop broken," "Blackboard not loading") are not valid excuses—start assignments early.

Homework and Collaboration

Homework will be assigned regularly through WeBWorK and other platforms. You may discuss homework concepts and approaches with classmates, but each student must write and submit their own code and solutions independently.

The best way to learn this material is to work through problems yourself. If you are stuck:

- Review lecture notes and examples first
- Try breaking the problem into smaller steps
- Come to office hours for guidance on your approach
- Discuss concepts with classmates, but write your own code

Collaboration is encouraged for learning; copying is not permitted.

Assignment Deadlines

Assignments have firm deadlines with no extensions except for documented emergencies (as described in the Makeup Policy above).

Why this policy exists:

- In professional settings, data analysts face unmovable deadlines from clients, teams, and regulatory requirements
- Learning to manage time and anticipate problems is a crucial professional skill
- Starting assignments early allows time to encounter problems and seek help

Plan ahead:

If your laptop breaks, internet fails, or Blackboard has issues on the due date, you should have already started the assignment days earlier. These situations, while frustrating, are not valid excuses for late work.

Also, note that the University offers several computers at different locations (e.g., libraries). Further laptop loan options are also available at the Bird and Carnegie library locations.

If you are concerned about meeting a deadline, come to office hours or email me well before the due date to discuss your approach.

Attendance:

You are expected to attend every class and every exam. If you miss a class, it is your responsibility to obtain a copy of the lecture notes for that class from another student. You are also responsible for any

announcements about changes to the course schedule, the exam schedule, or the course requirements that were made during that class.

It is a federal requirement that faculty promptly notify the university of students who do not attend or cease to attend any class. Faculty will use Early-Semester Progress Reports and Mid-Semester Progress Reports in Orange SUccess to alert the Registrar and Financial Aid Office on non-attendance.

Technology Policies

During Class:

Phones, laptops, tablets, and similar electronics must be put away during class time. This policy is strictly enforced.

Rationale:

Live coding demonstrations require your focused attention on problem-solving logic and computational thinking. Research shows that multitasking with devices significantly reduces learning outcomes for this type of material.

Consequences:

Students who repeatedly violate this policy will be asked to leave the classroom and their overall course grade will be negatively affected.

Note:

Calculator apps on phones cannot be used during exams or quizzes.

Calculators:

You will need a scientific calculator for computations throughout the course. Calculators must be your own (no sharing during exams). Phone calculators are not permitted on exams.

Statistical Software:

Students will use R (free, open-source statistical software) for this course.

R can be accessed through:

- Your personal computer (download from r-project.org and rstudio.com)
- ITS Public Computer Labs (multiple campus locations with R, SPSS, Minitab, SAS)
- Visit answers.atlassian.syr.edu/wiki and search “ITS lab locations”

R Commander (RCmdr) is also available as a graphical interface within R (see Learning Pathways section above).

Learning R:

You are expected to develop R skills using online resources, lecture materials, and practice assignments. The Week 0 Module provides initial orientation. Office hours are available for troubleshooting and conceptual questions.

Important Syllabus Reminders from the Office of Academic Affairs

Following are the language the University asks the faculty to provide in their syllabi: [from academicaffairs.syracuse.edu/important-syllabus-reminders/]

These as well as the important semester dates will be updated closer to the beginning of semester.

What Changed and Why (also the reason behind late submittal)

Since CRI (summer '25), our department (Mathematics) has started restructuring the statistics programs. MAT 422 was built as a bridge between MAT 222 (intro stats) and 500-level courses like 524 and 527. But now the stats BA/BS programs and several of these courses are probably (extremely likely) getting cut. So, I have been redesigning a course that might only run once or twice more. Still, doing the redesign was useful. It helped me work through teaching problems that apply to other courses too.

The Problem:

- Students show up to MAT 422 all over the map.
- Some already code in R.
- Others do not know what a zip file is.
- My old syllabus assumed everyone had some basic skills.
- That meant beginners were lost from day one, and advanced students were bored.

What I Did:

- Added a "Week 0 Module"
    - Students have to complete an online module before class starts.
    - It covers installing R, finding their way around RStudio, and running basic commands.
    - Students who already know this can skip to a quick quiz.
    - This way we can actually start teaching on day one instead of fixing computer problems.
- Added "Skills Checkpoints (15% of grade)"
    - Six short coding assignments spread across the semester
        - At weeks 3, 5, 7, 9, 11, 13.
    - They start easy (e.g., use this function)
        - and get harder (e.g., write an algorithm with loops).
    - Before, students did homework and then suddenly had to do big projects.
    - The jump was too big
        - like going from "type pnorm()"
        - to "build Newton's method from scratch."
    - The checkpoints should fill that gap.
    - Each is only 2.5% of the grade, so making mistakes does not kill you.
    - I made room by
        - dropping exams from 50% to 45% and
        - homework from 25% to 15%.
    - Clear Instructions About RCmdr-package ("R-Commander") vs. RStudio
        - My old syllabus mentioned RCmdr-package ("R-Commander") once.
        - Students had no idea when to use it.
        - Now there is a whole section:
            - RCmdr-package ("R-Commander") (for beginners):
            - Click buttons, it writes code.
            - But you have to understand what the code does.
        - RStudio (if you can already code):
            - Write code yourself.
            - I tell them to try moving from RCmdr-package ("R-Commander") to RStudio by midterm.
        - Some assignments require RStudio anyway.
- Clearer Learning Outcomes

    - Changed fuzzy stuff
        - like "Use R...slightly above beginner level"
        - to concrete stuff like "Import and analyze a dataset with 1000+ rows."
    - Now everyone knows what they actually need to do.
- Less Harsh Tone
    - Same policies (no makeups, firm deadlines, no phones in class) but I explain why instead of just saying no.
    - Added a section that says, "It's normal to struggle" and lists where to get help.
    - The rules are the same, just less "tough luck, figure it out yourself."

Connection to CRI:

- This uses stuff from the institute
    - start with what I want students to learn,
    - build toward it in small steps,
    - give options for different starting points, and
    - make it okay to struggle.
- If I End Up Teaching This:
    - I would need to record about 25 minutes of videos for Week 0 (do it once, use forever),
    - write 6 checkpoint assignments, and
    - grade with simple rubrics (does it work? can you explain it?).
- I would check if it worked by looking at:
    - Week 0 completion rates,
    - whether checkpoint scores get better over time,
    - whether projects look better than before, and
    - what students say about whether the pathways and checkpoints helped.

Even if MAT 422 does not stick around,

- this redesign clarified some things I can use elsewhere:
    - break hard skills into smaller pieces,
    - let students choose their path,
    - admit that struggling is part of learning while still expecting a lot.
- I will use these ideas in other courses.