Quantitative Methods for Linguistics
R, Bayesian Statistics, and Reproducible Research

LING 3XXX – Fall 2026

Course information

Time	TBD
Location	TBD
Instructor	Utku Turk
Email	utkuturk@umd.edu
Office hours	By appointment
Course site	Canvas

Prerequisites: Basic statistics (e.g., intro stats, research methods). No prior programming experience required.

1 Course description

How do we understand what statistical models actually do? How can we test whether our models are appropriate for our data? What does it mean for a model to "learn" from data?

This course takes a **simulation-first approach** to quantitative methods in linguistics. I know statistics can be intimidating—I've been there! My goal is to make it approachable by peeling back the math and showing you what's really happening under the hood. Rather than treating statistical models as black boxes that magically produce p-values, we'll build models from the ground up by simulating data. This approach makes abstract statistical concepts concrete: you'll see exactly how regression works, what mixed-effects models assume, and why Bayesian inference is powerful.

Following the approach of McElreath's *Statistical Rethinking*, we emphasize understanding models as generative processes. You'll learn to: (i) simulate data from a specified model, (ii) fit models to recover the parameters you simulated, (iii) check whether your model assumptions are met, and (iv) use simulations for power analysis and model validation.

By the end of the semester, you'll be able to design and execute quantitative studies, analyze complex linguistic datasets, fit appropriate statistical models, and—most importantly—understand what those models are actually doing. I'm excited to work through this journey with you.

No prior programming or statistics experience required—we build everything from scratch through simulation.

2 Learning objectives

Upon successful completion of this course, students will be able to:

1. Program in R for data analysis and simulation
2. Simulate data from statistical models (regression, mixed-effects, etc.)
3. Understand models as generative processes

4. Fit and interpret linear and logistic regression models
5. Understand and apply mixed-effects models for repeated measures data
6. Use simulation to check model assumptions and validate results
7. Conduct power analysis through data simulation
8. Fit Bayesian models using Stan and brms
9. Specify informative priors based on domain knowledge
10. Create reproducible research documents using Quarto
11. Critically evaluate statistical claims in linguistic research

3 Course structure

Weekly rhythm

Component	What it looks like
Lecture	Concepts, statistical theory, and when to use which methods (Tue/Thu first half).
Coding lab	Hands-on R work: data wrangling, visualization, model fitting (Tue/Thu second half).
Homework	Weekly problem sets with coding + interpretation questions.
Simulation Challenge	Each week features a "Simulation Challenge"—can you generate data that breaks a model's assumptions? Can you recover parameters from messy data? These challenges build intuition for what models actually do.

Why simulation-first?

Traditional stats courses teach you to run tests on data. This course teaches you to **think like a statistician**: if I believe X about the world, what data would I expect to see? What patterns would surprise me? By simulating data first, you develop deep intuition about what models assume and when they fail.

Simulation Challenges (weekly coding puzzles):

- Week 2: Simulate 100 coin flips—how often do you get 5 heads in a row?
- Week 4: How many participants do you need to detect a 50ms difference in reading times?
- Week 6: Can you trick a regression model by adding a correlated predictor?
- Week 8: Simulate crossed random effects—what happens when you ignore them?

Tools we use

- **R:** primary programming language for statistical computing
- **RStudio:** integrated development environment
- **tidyverse:** data wrangling and visualization (dplyr, ggplot2, tidyr)
- **lme4 & brms:** mixed-effects models (frequentist and Bayesian)
- **Stan:** Bayesian inference engine
- **Quarto:** reproducible research documents

4 Course requirements

4.1 Grading

Item	%	What counts
Participation + in-class coding	15	Active engagement in labs; completion of in-class exercises.
Homework (10 assignments)	50	Coding + interpretation questions (5% each).
Midterm project	10	Data analysis report using regression or mixed-effects models.
Final project	25	Complete data analysis with Quarto report and presentation.

4.2 Homework policy

Homework is due on Canvas by 11:59pm on the specified date. Late submissions lose 10% per day (up to 3 days). After 3 days, maximum 50% credit without prior arrangement.

You may collaborate on homework, but you must write your own code and answers. Copying code from classmates or online sources without attribution is academic dishonesty.

4.3 Projects

Midterm project (Week 9): Analyze a linguistic dataset using regression or mixed-effects models. Write a 3–4 page report including: research question, data description, exploratory visualizations, model specification, results, and interpretation. Submit code + report.

Final project (Finals week): Complete data analysis project on a topic of your choice. This should include: (i) clear research question, (ii) data collection or curation, (iii) exploratory data analysis with visualizations, (iv) appropriate statistical models, (v) model diagnostics and validation, (vi) interpretation of results, and (vii) discussion of limitations. Create a reproducible Quarto document (8–10 pages) + 10-minute presentation.

Example topics: analyzing corpus data for syntactic variation, modeling reaction times in psycholinguistic experiments, investigating sociolinguistic patterns, or testing phonological predictions with acoustic data.

5 Course schedule (16-week semester)

Schedule subject to change. Canvas is the live version.

Wk	Topic	Readings / Resources	Lab / due
1	Introduction; why simulation?; R basics	**Read:** McElreath (2020) *Statistical Rethinking* Ch. 1–2; Wickham & Grolemund (2017) Ch. 1–4. **Optional:** DeBruine & Barr (2021) "Understanding mixed-effects models through data simulation".	Installing R and RStudio; first simulations
2	Simple simulations in R; sampling	**Read:** McElreath (2020) Ch. 3; Wickham & Grolemund Ch. 5. **Optional:** Winter (2020) Ch. 1–2.	Simulating coin flips; random sampling; distributions **Due:** HW1
3	Simulations, p-values, and NHST	**Read:** McElreath (2020) Ch. 3; Gelman & Hill (2007) Ch. 1–2. **Optional:** Wasserstein & Lazar (2016) "The ASA statement on p-values".	Simulating null distributions; understanding p-values
4	Power analysis through simulation	**Read:** Green & MacLeod (2016) "SIMR"; DeBruine & Barr (2021). **Optional:** Brysbaert & Stevens (2018) "Power analysis and effect size".	Estimating power; sample size planning **Due:** HW2
5	Simulating and examining simple regression	**Read:** McElreath (2020) Ch. 4; Gelman & Hill (2007) Ch. 3. **Optional:** Winter (2020) Ch. 5–6.	Generating data from linear models; recovering parameters
6	Multiple regression; interactions	**Read:** McElreath (2020) Ch. 5; Gelman & Hill (2007) Ch. 4. **Optional:** Schad et al. (2020) "How to capitalize on a priori contrasts".	Simulating interactions; centering predictors **Due:** HW3
7	Generalized linear models; logistic regression	**Read:** McElreath (2020) Ch. 10–11; Gelman & Hill (2007) Ch. 5. **Optional:** Jaeger (2008) "Categorical data analysis: Away from ANOVAs".	Simulating binary outcomes; link functions
8	Simulating hierarchical data structures	**Read:** McElreath (2020) Ch. 13; DeBruine & Barr (2021). **Optional:** Baayen et al. (2008) "Mixed-effects modeling with crossed random effects".	Nested and crossed random effects **Due:** HW4
9	Mixed-effects models I: random intercepts	**Read:** McElreath (2020) Ch. 13; Winter (2020) Ch. 9–10. **Optional:** Gelman & Hill (2007) Ch. 11–12.	Simulating and fitting random intercept models **Due:** Midterm project
10	Mixed-effects models II: random slopes	**Read:** McElreath (2020) Ch. 13; Barr et al. (2013) "Random effects structure". **Optional:** Bates et al. (2015) "Fitting linear mixed-effects models using lme4".	Simulating varying slopes; maximal models **Due:** HW5
11	Introduction to Bayesian inference	**Read:** McElreath (2020) Ch. 1–3. **Optional:** Kruschke (2015) Ch. 2; Gelman et al. (2014) Ch. 1.	Prior distributions; posterior inference; grid approximation
12	Bayesian regression with brms and Stan	**Read:** McElreath (2020) Ch. 4, 9; Bürkner (2017) "brms". **Optional:** Vasishth et al. (2018) "Bayesian data analysis in the phonetic sciences".	Fitting Bayesian models; prior specification **Due:** HW6
13	MCMC diagnostics; posterior predictive checks	**Read:** McElreath (2020) Ch. 9; Gelman et al. (2014) Ch. 6. **Optional:** Gabry et al. (2019) "Visualization in Bayesian workflow".	Checking convergence; validating models through simulation
14	Model comparison; information criteria	**Read:** McElreath (2020) Ch. 7; Vehtari et al. (2017) "Practical Bayesian model evaluation". **Optional:** Burnham & Anderson (2002) *Model Selection*.	WAIC, LOO-CV; cross-validation **Due:** HW7
15	Reproducible research with Quarto	**Read:** Quarto documentation `https://quarto.org/docs/guide/`; Marwick et al. (2018) "Packaging data analytical work reproducibly". **Optional:** Wilson et al. (2017) "Good enough practices in scientific computing".	Creating Quarto documents; code chunks; citations
16	Final project presentations	–	10-minute presentations; peer feedback
Finals	–	–	**Due:** Final project (Quarto report + code)

6 Policies

6.1 What you might struggle with (and how to succeed)

This course involves learning both statistics and programming simultaneously. Here's what students typically struggle with and evidence-based advice:

Time management:

- Expect 8–10 hours/week outside class: 2–3 hours reading, 5–6 hours coding and homework, 1 hour reviewing notes
- Don't cram: Distributed practice works better than massed practice (Dunlosky et al., 2013)
- Start homework early: Debugging R code takes longer than you think

Learning to code:

- Everyone struggles at first: Programming is a skill that requires practice, not innate talent
- Read error messages carefully: They tell you what went wrong
- Google is your friend: Most R errors have been encountered before; search for solutions
- Practice regularly: Code a little bit every day rather than in long sessions

Note-taking:

- Take notes by hand during lectures: Handwriting improves retention (Mueller & Oppenheimer, 2014)
- Type code during labs: You learn by doing, not by watching
- After class: Annotate your code with comments explaining what each line does

Understanding statistics:

- Focus on concepts, not formulas: Understand what a model does, not just how to fit it
- Visualize everything: Plots help you understand patterns in data
- Ask "why?": Why this model? Why these predictors? Why this interpretation?
- Connect to linguistics: Every statistical concept has a linguistic application

Debugging code:

- Start simple: Test small pieces of code before combining them
- Print intermediate results: Use `print()` or `View()` to check what your code is doing
- Read documentation: Use `?function_name` to see how functions work
- Ask for help: Come to office hours, post on discussion board, work with classmates

Resources:

- R for Data Science (free online): `https://r4ds.had.co.nz/`
- RStudio cheatsheets: `https://www.rstudio.com/resources/cheatsheets/`
- Stack Overflow: `https://stackoverflow.com/questions/tagged/r`
- Learning How to Learn (free Coursera course): `https://www.coursera.org/learn/learning-how-to-learn`

6.2 Attendance

Attendance is expected. Labs are hands-on and difficult to replicate remotely. If you must miss class, notify me in advance and complete the lab exercises on your own.

6.3 Academic integrity

Do your own work; cite sources. You may collaborate on homework, but submitted code and answers must be your own. Copying code from classmates or online sources without attribution is plagiarism.

When using code from Stack Overflow, tutorials, or documentation, cite the source with a comment and demonstrate understanding by explaining what it does.

6.4 Use of ChatGPT and LLMs

LLMs may be used for **support** (debugging, learning syntax, explaining error messages) but not to generate solutions to homework problems. Any use must be documented.

Why this matters: You need to understand the statistics and code yourself. LLMs make mistakes in statistical reasoning, and you need to be able to catch them.

6.5 Accessibility & wellness

If you need accommodations, please contact the relevant campus office and talk to me as early as possible in the semester. I'm committed to making sure everyone can succeed in this course.

If you're struggling—whether it's with the material, coding, or anything else in life—please reach out. I really appreciate when students communicate with me, and I'm happy to work with you to make a plan together. Your wellbeing matters more than any assignment deadline.

7 Resources

Textbooks (all free online)

- **Primary:** McElreath (2020). *Statistical Rethinking* (2nd ed.). `https://xcelab.net/rm/statistical-rethinkin`
- Wickham & Grolemund (2017). *R for Data Science*. `https://r4ds.had.co.nz/`
- Winter (2020). *Statistics for Linguists: An Introduction Using R*. `https://doi.org/10.4324/9781315165547`
- Gelman & Hill (2007). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. (library access)
- Healy (2019). *Data Visualization: A Practical Introduction*. `https://socviz.co/`

Technical resources

Category	Links
R & RStudio	Download R `https://cran.r-project.org/`; RStudio `https://www.rstudio.com/products/rstudio/download/`
tidyverse	Documentation `https://www.tidyverse.org/`; Cheatsheets `https://www.rstudio.com/resources/cheatsheets/`
Mixed models	lme4 `https://github.com/lme4/lme4`; brms `https://paul-buerkner.github.io/brms/`
Bayesian	Stan `https://mc-stan.org/`; Bayesian workflow `https://github.com/betanalpha/knitr_case_studies`
Quarto	Documentation `https://quarto.org/`; Gallery `https://quarto.org/docs/gallery/`

This syllabus is a living document and may be updated during the semester. Last updated: January 3, 2026

Appendix: Quarter-system (10 weeks)

Wk	Topic	Readings	Lab / due
1	R basics; data wrangling	**Req:** Wickham & Grolemund Ch. 1–5; Winter Ch. 1.	R, RStudio, dplyr basics
2	Data visualization	**Req:** Wickham & Grolemund Ch. 3; Healy Ch. 1–3.	ggplot2; publication-quality figures **Due:** HW1
3	Linear regression	**Req:** Winter Ch. 5–7; Gelman & Hill Ch. 3–4.	Simple and multiple regression
4	Logistic regression	**Req:** Winter Ch. 8; Gelman & Hill Ch. 5.	Categorical outcomes; predictions **Due:** HW2
5	Mixed-effects models	**Req:** Winter Ch. 9–11; Baayen et al. (2008); Barr et al. (2013).	Random intercepts and slopes **Due:** Midterm project
6	Data simulation	**Req:** DeBruine & Barr (2021); Green & MacLeod (2016).	Simulating data; power analysis **Due:** HW3
7	Bayesian inference	**Req:** McElreath Ch. 1–3; Bürkner (2017).	Priors, posteriors, brms
8	Model comparison	**Req:** Gelman et al. Ch. 6–7; Vehtari et al. (2017).	Cross-validation; information criteria **Due:** HW4
9	Reproducible research	**Req:** Quarto documentation; Marwick et al. (2018).	Quarto documents; final project work
10	Final presentations	–	10-min presentations **Due:** Final project

Quarter grading:

Item	%	What counts
Participation + in-class coding	15	Active engagement in labs.
Homework (4 assignments)	40	Coding + interpretation (10% each).
Midterm project	15	Data analysis with regression/mixed models.
Final project	30	Complete analysis with Quarto report + presentation.