# Bayesian Cognitive Modeling for Linguistics
## From Theory to Implementation

LING 4XXX – Fall 2026

## Course information

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

**Prerequisites:** Introduction to programming (R or Python) and basic statistics.

## 1 Course description

How do we test theories of cognition? How can we formalize our hypotheses about how the mind works? This course introduces computational modeling as a tool for understanding cognitive processes, with a specific focus on **Bayesian Cognitive Modeling**.

Unlike data analysis, which asks "is there an effect?", cognitive modeling asks "what underlying process generated this data?". We will use the probabilistic programming language **Stan** to build, fit, and evaluate generative models of cognition. We will cover the full workflow: from defining a mathematical model of a cognitive process, to implementing it in code, to recovering parameters from simulated data, and finally applying it to real human behavioral data.

## 2 Learning objectives

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

1. Explain the purpose and value of quantitative modeling in cognitive science, distinguish among descriptive, pure predictive, process-characterization, and explanatory models, and critically evaluate model scope, testability, and identifiability.
2. Critically evaluate and implement simulations of cognitive models to explore their behavior under different parameter settings. Assess model performance, and understand the role of parameters in determining a model's behavior and explanatory power.
3. Implement Bayesian cognitive models using a probabilistic programming language (Stan) and interpret the structure of existing models.
4. Evaluate the robustness of Bayesian cognitive models, particularly in how they replicate human behavior.
5. Clearly communicate the structure of a Bayesian cognitive model, how it performs relative to human data, and interpret the model performance.

# 3 Course content & Structure

The course is divided into five main modules:

1. **Foundations:** Role of computational modeling in cognitive science; Simulation and model fitting.
2. **Probability:** Probability theory and Bayes rule; Generative processes.
3. **Tools:** Bayesian models in the probabilistic programming language Stan.
4. **Modeling:** Cognitive modeling in a Bayesian framework (Signal Detection, Categorization, Memory).
5. **Evaluation:** Model comparison, parameter recovery, and robustness checks.

# 4 Course requirements

## 4.1 Grading

| Item | % | What counts |
|---|---|---|
| Participation | 10 | Active engagement in labs and discussions. |
| Weekly Labs (10x) | 40 | Coding assignments in R/Stan (4% each). |
| Midterm Project | 20 | Replicating a simple cognitive model from the literature. |
| Final Project | 30 | Implementing and testing a novel or extended cognitive model. |

## 4.2 Labs

Weekly labs are the core of the course. You will implement models in ‘R‘ and ‘Stan‘, run simulations, and visualize results. Labs are due one week after they are assigned.

## 4.3 Projects

**Midterm:** Replicate a standard model (e.g., Signal Detection Theory, GCM) using data provided in class. Focus on correct implementation and parameter recovery.

**Final:** Choose a cognitive phenomenon (e.g., decision making, sentence processing, memory retrieval) and implement a Bayesian model for it. You must simulate data, fit the model, and evaluate its ability to recover parameters.

## 5 Course schedule (16-week semester)

*Readings based on Vasishth et al. (Bayesian Cognitive Modeling) and Lee & Wagenmakers.*

| Wk | Topic | Readings / Concepts | Lab / Assignment |
|---|---|---|---|
| **MODULE 1: Foundations** | | | |
| 1 | Intro to Cognitive Modeling | Descriptive vs. Explanatory models; The generative stance. | Lab 1: Simulating simple processes in R |
| 2 | Probability & Bayes Rule | Joint, marginal, conditional probability; Bayes' Theorem. | Lab 2: Probability math & simple updating |
| 3 | Intro to Stan | The concept of probabilistic programming; Stan syntax. | Lab 3: Hello World in Stan (Binomial model) |
| **MODULE 2: Basic Models (Lee & Wagenmakers)** | | | |
| 4 | Parameter Estimation | Binomial rate; Gaussian parameters; Difference between rates. | Lab 4: Inferring basic parameters |
| 5 | Signal Detection Theory (SDT) | Equal variance SDT; Discriminability ($d'$) and Bias ($c$). | Lab 5: Implementing SDT in Stan |
| 6 | Latent Variables & Categorization | GCM; Change detection; Exemplar theory. | Lab 6: GCM implementation |
| **MODULE 3: Hierarchical Structures** | | | |
| 7 | Hierarchical Models I | Pooling; Shrinkage; Hyperparameters. | Lab 7: Hierarchical Binomial Model |
| 8 | Hierarchical Models II | Individual differences; Log-normal RT models. | **Due:** Midterm Project |
| 9 | Model Comparison | Bayes Factors; CV; WAIC; LOO. | Lab 8: Comparing models via LOO |
| **MODULE 4: Advanced Architectures (Vasishth et al.)** | | | |
| 10 | Multinomial Processing Trees | Vasishth Ch 16; Modeling categorical responses; Picture naming. | Lab 9: MPTs for Aphasia |
| 11 | Mixture Models | Vasishth Ch 17; Fast-guess model; Speed-accuracy trade-off. | Lab 10: Mixture modeling |
| 12 | Accumulator Models (LBA) | Vasishth Ch 18; Log-normal race model; Lexical decision. | – |
| **MODULE 5: Evaluation & Project** | | | |
| 13 | Robustness & Predictive Checks | Posterior predictive checks; Parameter recovery. | – |
| 14 | Final Project Workshop I | Troubleshooting Stan code. | – |
| 15 | Final Project Workshop II | Peer review of preliminary results. | – |
| 16 | Final Presentations | Student presentations of final models. | **Due:** Final Project Report |

# 6 Policies

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

**Coding in Stan:** Stan can be challenging because it is declarative (you declare the model) rather than imperative. But don't worry—we will tackle this together, starting simple and building up complexity line by line.

- **Start simple:** Build the simplest possible model that compiles, then add complexity one line at a time.
- **Use the manual:** The Stan User Guide is incredibly good. Read it.
- **Simulate first:** Always simulate data from known parameters to check if your model can recover them.

**Bayesian concepts:**

- Focus on the generative process: "If I were a god designing a universe that follows these rules, how would data appear?"
- Don't worry about the integrals: Focus on interpreting the posterior distribution.

## 6.2 Academic integrity

You may collaborate on code, but you must write your own write-ups and understand every line of code you submit.

# 7 Resources

1. **Textbook 1:** Lee, M. D., & Wagenmakers, E. J. (2014). *Bayesian Cognitive Modeling: A Practical Course*. Cambridge University Press.
2. **Textbook 2:** Vasishth et al. (2024). *Introduction to Bayesian Data Analysis for Cognitive Science*. Online: `https://bruno.nicenboim.me/bayescogsci/`
3. **Stan User Guide:** `https://mc-stan.org/users/documentation/`

# Appendix: Quarter-system (10 weeks)

| Wk | Topic | Readings | Assignment |
|---|---|---|---|
| 1 | Foundations & Probability | Generative stance; Bayes Rule. | Lab 1: Simulating processes |
| 2 | Intro to Stan | Stan syntax; Binomial model. | Lab 2: Hello World in Stan |
| 3 | Parameter Estimation | Binomial & Gaussian estimation. | Lab 3: Estimating parameters |
| 4 | Signal Detection Theory | SDT model; Derived parameters. | Lab 4: SDT in Stan |
| 5 | Hierarchical Models I | Intro to pooling & shrinkage. | Lab 5: Hierarchical Binomial |
| 6 | Hierarchical Models II | Continuous hierarchies; RTs. | – |
| 7 | Model Comparison | Bayes Factors, LOO, WAIC. | Lab 6: Model comparison |
| 8 | Memory & Categorization | GCM or Retention models. | – |
| 9 | Project Workshop | Debugging & fitting. | – |
| 10 | Final Presentations | – | **Due:** Final Project |