SML 320: Bayesian Analysis

Instructor: Derek Sollberger (dsollberger@princeton.edu)

Course Description: This course provides an introduction to Bayesian analysis—a powerful statistical framework for making inferences and modeling uncertainty in a wide range of applications. Students will explore the fundamental principles of Bayesian statistics, probability theory, Bayesian inference, and practical applications of Bayesian modeling. The course will cover both the theory and hands-on implementation using data science software and the R programming language.

Prerequisites:

- Basic knowledge of probability distributions (binomial, normal)
- Understanding of statistical concepts (sampling, confidence intervals)
- Familiarity with R or Python programming is assumed
- One semester of Calculus or instructor approval

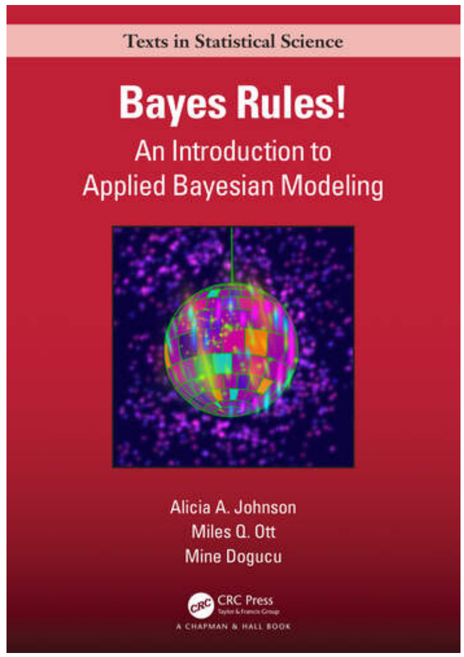
Course Learning Outcomes: By the end of this course, students will be able to:

1. Understand the foundational principles of Bayesian analysis and probability theory
2. Apply Bayesian techniques for parameter estimation and regression models
3. Create and interpret Bayesian models for various types of data
4. Effectively communicate results and conclusions from Bayesian analyses
5. Gain practical experience using Bayesian analysis software

Assessment The course grades will be calculated by the following weights:

- Before Lecture (10%): These tasks will ask students to review or preview statistics concepts and will be handled by Canvas quizzes. These tasks will have due dates at the start of each lecture session (unless otherwise announced).
- Computation (25%): Weekly problem sets (about 10 to 15 mathematical and computer programming tasks from the textbook and/or created by the instructor) will be assigned. Students are encouraged to discuss these tasks during precepts and study groups.
- Inference (15%): Students will be presented scenarios of model summaries and results and will be asked to answer inference tasks. These assignments will be handled through GradeScope.
- Project Benchmarks (10%): As seen in the schedule below, students will maintain progress toward their semester project. Students are encouraged to discuss these plans during precepts and office hours.
- Progress Presentation (10%): Students will update instructors and their peers on their data search, exploratory data analysis, and statistical modeling
- Final Presentation (10%): Students will present their application of Bayesian techniques and literature review toward their semester project
- Semester Project (20%): The culminating semester project will be due at the end of the semester examination period and will be assessed according to a rubric that will be provided in advance.

Reading List: This course will closely follow the *Bayes Rules!* textbook by Alicia A Johnson, Miles Q Ott, and Mine Dogucu. It is, in my opinion, the best blend of Bayesian thought, mathematical background, computer processes, and relevant applications. The authors have made the materials of their textbook available online at `https://www.bayesrulesbook.com/`



The following list of books is optional for student studies, but the instructor may use some materials to add depth and interest to the course.

- *Statistical Rethinking* by Richard McElreath is the premier body of work in the field of Bayesian analysis. This resource is great for people who want to build a strong foundation in philosophy and theory in this branch of mathematics.
- *Bayesian Data Analysis* by Andrew Gelman, et al., is the classic textbook (available online) in this field that is used in several university courses. The authors' approach work well for people looking to quickly add Bayesian approaches to their research skills.
- *Bayesian Statistics the Fun Way* by Will Kurt brings Bayesian notions to a broad audience and its presentation blends will with an introductory course in statistics.
- *Bayesian Thinking in Biostatistics* by Gary L Rosner, et al., provides rigorous applications in bioinformatics along with strong software use.

Calendar and List of Topics:

Week	**Date**	**Topic**	**Textbook Chapter**	**Precept**	**Project Benchmark**
1	Jan 30	Conditional Probability	1		
1	Feb 1	Bayes' Rule	2	Problem Set 1	Software Installation
2	Feb 6	Beta-Binomal Model	3		
2	Feb 8	Balance and Sequentiality	4	Problem Set 2	Topic Proposal
3	Feb 13	Conjugate Families	5		
3	Feb 15	Approximating the Posterior	6	Problem Set 3	Data Search
4	Feb 20	MCMC	7		
4	Feb 22	MCMC	7	Problem Set 4	Explore Data
5	Feb 27	Posterior Inference	8		
5	Feb 29	Posterior Prediction	8	Problem Set 5	
6	Mar 5	**Progress Presentations**			
6	Mar 7	**Progress Presentations**			
	Mar 11-15	*Spring Recess*			
7	Mar 19	Normal Regression	9		
7	Mar 21	Evaluating Regression Models	10	Problem Set 6	Literature Search
8	Mar 26	Extension of Normal Regression	11		
8	Mar 28	Poisson and Negative Binomial	12	Problem Set 7	
9	Apr 2	Logistic Regression	13		
9	Apr 4	Naive Bayes Classification	14, 15	Problem Set 8	Literature Review
10	Apr 9	Bayesian Neural Networks			
10	Apr 11	Hierarchical Models	16, 17	Problem Set 9	
11	Apr 16	Bayesian Neural Networks 2			
11	Apr 18	Non-Normal Hierarchical Models	18, 19	Problem Set 10	
12	Apr 23	**Final Presentations**			
12	Apr 28	**Final Presentations**			

Computation problem sets, inference written answers, and project benchmarks will be due at Friday, 11:59 PM EST on their respective weeks.

Class Policies:

1. Lecture sections: Please keep extra noise to a minimum. Cell phones may be used as long as they are on silent or vibrate. Please also review the Cooperative Classroom statement below.

2. Precepts will be held for 80 minutes per week. Students will develop problem-solving skills through collaborative work on the computer programming and written assignments while also working toward their semester projects.

3. Computers: Use of a laptop computer is highly recommended for this course, and students are asked to bring their laptop computer to every lecture and precept session.

   - More information about computer needs can be found at `https://princeton.service-now.com/service?id=kb_article&sys_id=KB0013768`
   - While Chromebooks (or other systems that discourage installation of software) can access cloud software, intensive calculations in this course may merit the use of a personal computer and downloaded software rather than server access.

4. Special Accommodations: Students must register with the Office of Disability Services (ODS) (ods@princeton.edu; 258-8840) for disability verification and determination of eligibility for reasonable academic accommodations. Requests for academic accommodations for this course need to be made at the beginning of the semester, or as soon as possible for newly approved students, and again at least two weeks in advance of any needed accommodations in order to make arrangements to implement the accommodations. Please make an appointment to meet with me in order to maintain confidentiality in addressing your needs. No accommodations will be given without authorization from ODS, or without advance notice.

5. Academic Integrity: You are allowed to read text books and resources online. You may not ask other individuals questions (e.g., you may not ask questions on Stack Exchange or R help discussion groups). In accordance with the honor code, you must cite all sources of external information used in your work. This can be a book or a web site. Part of being a successful data scientist is having the ability to leverage existing information and techniques, so it is okay to do so in this course as long as you cite the reference. University policies can be reviewed at

   `https://ua.princeton.edu/policies-resources/undergraduate-honor-system`

Disclaimer: Due to the adaptive nature of the course and learning environment, this document and schedule is subject to change.

Cooperative Classroom: Learning in a cooperative environment should be stimulating, demanding, and fair. Because this approach to learning is different from the competitive classroom structure that many other courses used to be based on, it is important for us to be clear about mutual expectations. Below are my expectations for students in this class. This set of expectations is intended to maximize debate and exchange of ideas in an atmosphere of mutual respect while preserving individual ownership of ideas and written words. If you feel you do not understand or cannot agree to these expectations, you should discuss this with your instructor and classmates.

1. Students are expected to work cooperatively with other members of the class and show respect for the ideas and contributions of other people.
2. When working as part of a group, students should strive to be good contributors to the group, listen to others, not dominate, and recognize the contributions of others. Students should try to ensure that everyone in the group is welcome to contribute and recognize that everyone contributes in different ways to a group process.
3. Students should explore data, make observations, and develop inferences as part of a group. If you use material from published sources, you must provide appropriate attribution.[1]

Inclusion and Diversity: I value all students regardless of their background, country of origin, race, religion, ethnicity, gender, sexual orientation, disability status, etc. and am committed to providing a climate of excellence and inclusiveness within all aspects of the course. If there are aspects of your culture or identity that you would like to share with me as they relate to your success in this class, I am happy to meet to discuss. Likewise, if you have any concerns in this area or facing any special issues or challenges, you are encouraged to discuss the matter with me (set up a meeting by e-mail) with an assurance of full confidentiality (only exception being mandatory reporting of academic integrity code violations or sexual harassment).[2]

Pep Talk! Learning R can be difficult at first—it is like learning a new language, just like Spanish, French, or Chinese. Hadley Wickham—the chief data scientist at RStudio and the author of some amazing R packages you will be using like `ggplot2`—made this wise obesrvation:

> It's easy when you start out programming to get really frustrated and think, "Oh it's me, I'm really stupid," or, "I'm not made out to program." But, that is absolutely not the case. Everyone gets frustrated. I still get frustrated occasionally when writing R code. It's just a natural part of programming. So, it happens to everyone and gets less and less over time. Don't blame yourself. Just take a break, do something fun, and then come back and try again later.

If you are finding yourself taking way too long hitting your head against a wall and not understanding, take a break, talk to classmates, ask questions ... e-mail me, etc. I *promise* you can do this.[3]

[1] This document has been adapted from *Scientific Teaching* by Jo Handelsman, Sarah Miller, and Christine Pfund

[2] This inclusion statement was written by chemistry professor Dr. Steve Zimmerman at the University of Illinois at Urbana-Champaign `https://mobile.twitter.com/steveczimmerman/status/1161019135251353606`

[3] This pep talk comes from data science instructor Andrew Heiss at Georgia State University. Source: `https://mobile.twitter.com/andrewheiss/status/1165310391750189063`