Topics in Politics: Applied Statistical Programming (Version 0.2)
L32 4625

Tuesdays and Thursdays
1:00-2:30
Applied Statistics Classroom
Seigle Hall L016

Instructor Information

Jacob M. Montgomery, Ph.D.
Assistant Professor, Department of Political Science
Office: Seigle 242
E-mail: jacob.montgomery@wustl.edu
Telephone: None
Office Hours: Wed. 9:00-11:00 and by appointment

Course description

Statistical computing is a quickly changing field. Standard techniques of today would have been difficult to execute fifteen years ago and impossible even in the late 1990s. Rapid improvements in computing power have been accompanied by swift changes in standard statistical methods. In just the last decade, techniques ranging from Markov chain Monte Carlo (MCMC) simulation, randomization inference, network analysis, and non-parametric matching have moved from being novel, advanced applications to commonplace across the social sciences.

This class is designed to achieve two broad objectives. More narrowly, it aims to guide students as they learn the specifics of the R programming language, a powerful statistical computing environment widely used in the fields of political science, network analysis, machine learning, and statistics. Achieving this goal will require students to learn commands, best practices, and work-arounds specific to the sometimes idiosyncratic R language.

More broadly, however, this course aims to provide students some of the foundational concepts and skills needed to engage in modern statistical computing generally. No course can teach you *everything* there is to know about R even as it exists today, and certainly no class can teach you every piece of software you will need to use in your career. Some of the tools that will be in wide use in ten years do not even exist today. Thus, this course aims to give you the more foundational meta-skills from computer science and statistics you need to teach *yourself* how to develop software to execute specific tasks in R or similar computer languages. Learning at this level will also better equip you to understand software written by others. In addition, the course will introduce a few widely used computational methods common to statistics (e.g., E-M algorithms, and numerical optimization).

The course will focus on helping students to understand the core concepts behind the R language, gain practical programming skills, and learn to apply both appropriately in a real-world setting. A major component of the course includes learning how to plan and execute a collaborative, complex programming project and how to effectively document and communicate the capabilities of the resulting software to others.

Learning objectives

By the end of this course, you should be able to:

- Explain the basic components of the R working environment
- Understand object-oriented programming (or at least R's version of this)
- Understand the basic control functions, flow functions, and data structures of R
- Functionalize complex and/or repetitive code
- Clearly document software during and after development
- Debug and analyze your code for speed
- Read-in and write-out data and text of any format, including information collected online
- Create custom data visualizations
- Implement class structures
- Create a simple R package capable of passing the CRAN checks
- Document and test an R package for general distribution
- Improve the performance of your code via parallelization and integration of `C++` routines
- Apply common computational statistics techniques and simulations methods
- Work collaboratively with colleagues to plan and execute complex software development

Team-based learning

This course will feature as little traditional lecturing as I can manage. Students will be expected to learn the basic content of the readings before class so that the majority of class time can be dedicated to discussion, group work, and hands-on demonstrations, which are more likely to facilitate successful learning. We will work in teams throughout the semester to maximize active engagement with the course material. By working in teams, students will not only develop communication and collaboration skills but assist each other in understanding and applying concepts successfully.

Early in the semester, you will be assigned to a team of three to five students. You will work with this team throughout the semester on both in-class assignments, (some) homework assignments, and your final project. To ensure that each student contributes the group's success, your contributions will be assessed via the self- and peer-evaluation components discussed below.

Requirements and Evaluation

Grading in this class will be based on the components described below. **Late work will not be accepted without prior permission**. Makeup exams will not be given, and students who miss exams will receive a score of 0 absent extraordinary circumstances.

Grading scale

Score	Grade	Score	Grade	Score	Grade	Score	Grade
$\geq$94	A	$\geq$83	B	$\geq$ 73	C	$\geq$63	D
$\geq$90	A-	$\geq$80	B-	$\geq$ 70	C-	$\geq$60	D-
$\geq$87	B+	$\geq$77	C+	$\geq$ 67	D+	<60	Fail

Peer assessments - 10%

Early in the semester, you will be assigned into a team of 3-5 individuals. You will work with this team throughout the semester on in-class assignments and your final research project. To help ensure that all members of the team are actively contributing, students will be asked to evaluate their teammates' contributions, effort, and performance. You will receive ungraded midterm evaluations from your team to help you know how well you are doing and identify areas in need of improvement. You will also complete a midterm self-evaluation of your own contributions, effort, and performance using an identical form to help you reflect on your own effort and performance. (All peer and self-evaluation forms are provided at the end of the syllabus.)

Problem sets and in-class work - 30%

Problem sets, or homeworks, will be distributed throughout the course (20%). *Unless otherwise specified*, these are individual assignments that you should prepare yourself, though you may ask your colleagues for help. To be clear, *every single keystroke for these assignments should be your own*. Please turn them in at the on the specified date **at the beginning of class**. If you have a printing problem, you are responsible for emailing it to me or the graduate TA before class starts. Each student's lowest homework grade will be dropped in the final grade calculations. This option should be reserved for illness, family emergencies, broken alarm clocks, or other unforeseen events. No additional waivers will be granted.

In-class assignments will be completed during class with your team (10%). All members will turn in a single assignment at the end of class and will share their grade. However, **all absent students will receive a zero**. Students missing more than five minutes of class time will be counted as absent. Each student's two lowest in-class assignment grades will be dropped in the final grade calculations. This option should be reserved for illness, family emergencies, broken alarm clocks, or other unforeseen events. No additional waivers will be granted.

Midterm exam - 25%

The midterm exam will be a take-home exam where you will be expected to independently create an R package to accomplish a specific task. The exam will be due at the beginning of class on March 24. Specific rules will be explained at the time of the exam.

Project - 35%

After the midterm exam, teams will be assigned a specific programming task of interest either to myself or another faculty member in the department. Working with your assigned teams, and under the close supervision of the faculty member, students will be responsible for planning, creating, and documenting a software package that meets the specified needs of the faculty member. This will include outlining the software, developing a plan, monitoring the progress of the software development, and evaluating the final product. The course will culminate with turning over the resulting software packages to me for grading at the time of the regularly scheduled final for this course.

Extra Credit

No adjustments will be made to final grades under any circumstances and no incompletes will be granted absent extraordinary circumstances. Students will have the opportunity to earn extra credit over the course of the semester to provide an extra cushion:

- Students can increase their final grade 1% by completing their official online course evaluations for both Professor Montgomery and the graduate TAs.
- Students may earn up to another 1% for creating a video tutorial or other online content to be added to the course website for the undergraduate QPM class. The topic of this video must be approved in advance, and the final product must be delivered to me before the final projects are handed in. You can find example videos here: `https://pages.wustl.edu/montgomery/articles/3458`

Class policies

Technology in the classroom

You will frequently make use of computers in this course. Please be respectful to your instructors and your peers by using your computers only for class-related purposes. Please put your phone away before class starts and don't bring it out.

Academic Honesty

Cheating and plagiarism will not be tolerated. I strongly encourage you to review the University's policies regarding academic honesty, which you can read at: `http://www.wustl.edu/policies/undergraduate-academic-integrity.html`.

In general, if you have any question, please feel free to ask your TA or Professor Montgomery. Specific rules for this course:

- You may work together on homework in small groups, but you should each prepare your answers separately unless otherwise instructed.
- The homeworks and in-class work are "open book" and "open notes."
- You are to consult *only* with Professor Montgomery or a TA during exams.

All cases of cheating or plagiarism will be referred to Washington University's Committee on Academic Integrity. If the Committee on Academic Integrity finds a student guilty of cheating, then the penalty will be (without exception) automatic failure of the course.

Students with disabilities

Students with disabilities enrolled in this course who may need disability-related classroom accommodations are encouraged to make an appointment to see me before the end of the second week of the semester. All conversations will remain confidential. Please also arrange to have the required documentation sent to me for any accommodations *at least two weeks prior to the first exam.*

Religious observances

Some students may wish to take part in religious observances that occur during this semester. If you have a religious observance that conflicts with your participation in the course, please meet with me *before the end of the second week of the semester* to discuss accommodations.

Teaching Assistant

There are is two graduate teaching assistants. They will work closely in conjunction with Professor Montgomery on all issues of grading, but all grading decisions will be mine.

Mr. David Carlson	Mr. Jonathan Homola
carlson.david@wustl.edu	homola@wustl.edu
Office Hours: TBA	Office Hours: TBA
Office: Seigle 276	Office: Seigle 277

Course materials

Textbooks

In addition to assigned readings that will be posted on Blackboard, the following books are required and can be purchased at the bookstore.

de Vries, Andrie and Joris Meys. 2015. *R for Dummies* (2nd Edition). Wiley.

Wickham, Hadley. 2015. *Advanced R*. CRC Press.

Wickham, Hadley. 2015. *R Packages: Organize, Test, Document, and Share Your Code*. OReilly.

The following books are recommended, but should be available through Springer Link (via the library).

Eddelbuettel, Dirk. 2013. *Seamless R and C++ Integration with Rcpp*. Springer.

Nolan, Deborah and Duncan Temple Lang. 2014. *XML and Web Technologies for data Sciences with R*. Springer.

The following books are suggested for additional readings.

> Braun, W. John and Duncan J. Murdoch. 2007. *A First Course in Statistical Programming with R*. Cambridge.
>
> Jones, Owen, Robert Maillardet, and Andrew Robinson. 2009. *Introduction to Scientific Programming and Simulation Using R*. Chapman and Hall (CRC)

Software and hardware

You will be using the R statistical package (`http://www.r-project.org/`). While R is available for every computing platform, many of the more advanced tasks performed in this class will be taught based on the assumption that you are working on a Mac machine. While I will *not* require that you purchase a Mac for this class, I strongly recommend that any student considering advanced methods training and work to do so in the near future. Once you are set up with R, R-Studio, and Git, do NOT update your operating system (OS) for the remainder of the semester. If you are considering updating your OS, do it now.

Plan of the course

The basic outline of the course is divided into four components. In Section 1, we will introduce the R computing environment and develop some basic and advanced skills and topics. In this portion of the class, we will begin with short lectures and discussions of the assigned readings. To encourage engagement with these materials, students will be asked to accomplish assigned programming tasks both inside and outside of the class period.

In Section 2, we will study the structure and development of R packages, which are collections of related functions needed to achieve specific analytical tasks. In Section 3, we will move from the abstract to the applied as the class takes on several real-world statistical programming challenges. Under the close supervision of me (or another designated faculty member), each team will be presented with a project requiring the development of a complex set of code. As part of this Section, several lecture periods will be dedicated to covering some advanced topics that will be needed to build the packages. By the end of this Section, each team should have a detailed plan for the software they wish to develop including a method for monitoring progress and evaluating the final product.

In Section 4, class periods will be dedicated to covering material to help student build the software, write documentation, develop detailed vignettes and demos for its component functionality, and unit testing.

Very tentative Schedule

All told, this is an ambitious project that will require a substantial intellectual engagement from each student. It will also require *flexibility* since the course may evolve – perhaps substantially – during the semester in response to the needs of the project, our results, and issues raised by students.

In particular, you will be expected to work with your teams throughout the semester both inside and outside the class. You will be involved in many collaborative projects in your career, so consider building a positive working dynamic within your team to be part of the assignment. I also expect that students be quick to inform me and/or the TA when assignments seem vague, overly difficult, confusing, or incomplete. **The schedule below should be viewed as no more than suggestive**.

Date	Topic	Reading	Assignments	Notes
Section 1				
1/19	`''Hello world''`	DVM Chpt 1-2, Appendix *Advanced R* Chapter 1		
1/21	Calculation/Structures 1	DVM Chpt 4,5 *Advanced R* Chapter 2	Complete survey	
1/26	Structures 2	DVM Chpt 7 *Advanced R* Chapters 3, 4		
1/28	Control/flow	DVM Chpt 9	PS 1 Due	
2/2	Functions	DVM Chpt 8 *Advanced R* Chpt 6		
2/4	Version Control Documentation	DVM Chpt 11 Bowers (2011) *R Packages* Chpt 13 *Advanced R* Chapter 5	PS 2 Due	
2/9	Classes	*Advanced R* Chpt 7		
2/11	Environments	*Advanced R* Chpt 8	PS 3 Due	
2/ 16	Debugging	DVM Chpt 10 *Advanced R* Chpt 9		
2/18	Plotting	DVM Chpts 16-18		
2/23	Webscraping 1	NL Chpts 1-3, 7-9	PS 4 Due	
2/25	Apply/Parallel			
3/1	Catch up		PS 5 Due	

Date	Topic	Reading	Assignments	Notes
Section 2				
3/3	Package structure	*R Packages* Chpts 1-4		
3/8	Documentation	*R Packages* Chpts 5,6		
3/10	Unit testing	*R Packages* Chpts 7,8	PS 6 Due	
3/22	Package misc	*R Packages* Chpts 14-15		
Section 3				
3/24	Improving code	*Advanced R* 16, 17	Midterm due	
3/29	Web scraping 2	TBA		
3/31	`Rccp` *Advanced R* 19	*Rccp* 1,2, App.	PS 7 Due	
4/5	`Rccp` 2	*Rccp* Chpt 3-4		
4/7	`Rccp` and packages *R Packages* Chpt 10	*Rccp* Chpt 5-7	PS 8 Due	
Section 4				
4/12	Numerical integration	JMR 9, 11		
4/14	Numerical optimization	JMR 10	PS 9 Due	
4/19	E-M algorithm			
4/21	Team work day		Software plan due	
4/26	Team work day			
4/28	NO CLASS (SLAMM)			

Self evaluation form (mid-semester; ungraded)

Team #:
Your name:

Part 1: Quantitative assessment (check one box for each item)

Cooperative learning skills	Never	Sometimes	Often	Always
Arrives on time and remains with team during activities				
Demonstrates a good balance of active listening and participation				
Asks useful or probing questions				
Shares information and personal understanding				

Self-directed learning	Never	Sometimes	Often	Always
Is well-prepared for team activities				
Shows appropriate depth of knowledge				
Identifies limits of personal knowledge				
Is clear when explaining things to others				

Interpersonal skills	Never	Sometimes	Often	Always
Gives useful feedback to others				
Accepts useful feedback from others				
Is able to listen and understand what others are saying				
Shows respect for the opinions and feelings of others				

Part 2: Qualitative assessment (1–3 sentences each)

1) What is the single most valuable contribution you make to your team?

2) What is the single most important way you could alter your behavior to more effectively help your team?

Peer evaluation form (mid-semester; ungraded)

Team #:
Colleague you are evaluating:
Your name (evaluator):

Part 1: Quantitative assessment (check one box for each item)

Cooperative learning skills	Never	Sometimes	Often	Always
Arrives on time and remains with team during activities				
Demonstrates a good balance of active listening and participation				
Asks useful or probing questions				
Shares information and personal understanding				

Self-directed learning	Never	Sometimes	Often	Always
Is well-prepared for team activities				
Shows appropriate depth of knowledge				
Identifies limits of personal knowledge				
Is clear when explaining things to others				

Interpersonal skills	Never	Sometimes	Often	Always
Gives useful feedback to others				
Accepts useful feedback from others				
Is able to listen and understand what others are saying				
Shows respect for the opinions and feelings of others				

Part 2: Qualitative assessment (1–3 sentences each)

1) What is the single most valuable contribution this person makes to your team?

2) What is the single most important way this person could alter their behavior to more effectively help your team?

Peer evaluation form (end of semester)

Name/team #:

Please assign scores that reflect how you really feel about the extent to which the other members of your team contributed to your learning and/or your teams performance. This will be your only opportunity to reward the members of your team who worked hard on your behalf. (Note: If you give everyone pretty much the same score, you will be hurting those who did the most and helping those who did the least.)

Instructions: In the space below, please rate each of the other members of your team. Each member's peer evaluation score will be the average of the points they receive from the other members of the team. To complete the evaluation you should: 1) List the name of each member of your team in the alphabetical order of their last names and, 2) assign an average of ten points to the other members of your team and, 3) differentiate some in your ratings; for example, you must give at least one score of 11 or higher (maximum = 15) and one score of 9 or lower.

Team member	Score
1.	
2.	
3.	
4.	

Additional feedback

Please briefly describe the reasons for your highest and lowest ratings in the space below. These comments will be shared anonymously. Note: Your comments should be descriptive, not evaluative; as clear and specific as possible; phrased in constructive terms; and focused on areas in which the student has made especially valuable contributions or could improve in the future.

Reason(s) for your highest rating(s):

Reason(s) for your lowest rating(s):