Course	STAT535-FA22 – Statistical Computing
Instructor	Patrick Flaherty <flaherty@math.umass.edu>
Meeting	TuTh 2:30–3:45pm at LGRT 141
Office Hours	Tu 3–4pm at LGRT 1444

Course Description

This course provides an introduction to fundamental computer science concepts relevant to the statistical analysis of large-scale data sets. Students will collaborate in a team to design and implement analyses of real-world data sets, and communicate their results using mathematical, verbal and visual means. Students will learn how to analyze computational complexity and how to choose an appropriate data structure for an analysis procedure. Students will learn and use the python language to implement and study data structure and statistical algorithms.

Course Objectives

At the end of the course, you should be able to:

1. Design and implement an end-to-end statistical analysis of a data set,
2. Connect notions of computational complexity to statistical methods for data analysis,
3. Collaborate effectively in a team to develop a solution to real-world problems using statistics,
4. Evaluate and critique the choice data structure for an algorithm, and
5. Communicate statistical analysis results using mathematics, verbal and visual means.

This course aims to not only give you the opportunity to learn fundamental computer science concepts critical for understanding machine learning algorithms, but also brings you into contact with real data and allows them opportunities to make meaningful quantitative contributions to problems in genetics, social science, and other disciplines.

Prerequisites

You must have completed STAT516 and CS121 or equivalent to take this course.

Resources

Required Texts

- *Introduction to Computation and Programming Using Python: With Application to Understanding Data* by John V. Guttag

Optional Texts

- *Problem Solving with Algorithms and Data Structures using Python* by Bradley Miller and David Ranum
- *Python for Data Analysis* by Wes McKinney
- *Introduction to Algorithms* by Cormen, Leiserson, Rivest, and Stein

You are expected to have read the assigned readings *before* the class session for which it is assigned.

Other Resources

This course make use of the MIT OCW *Introduction to Computer Science and Programming in Python* [1] and the MIT OCW *Introduction to Algorithms* [2]. You can find lots of videos, slides and resources at the websites for those courses (shown in references).

You will best learn programming through a lot of practice. Here are a few links for gaining more practice opportunities:

- MIT OCW Suggestions

Piazza

This term we will be using Piazza for class discussion. The system is catered to getting you help fast and efficiently from classmates, the TA, and myself. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza. If you have any problems or feedback for the developers, email team@piazza.com.

Find our class page at: `https://piazza.com/class/jz1m043kb7z7m4`

Assessment

This course assessments consist of seven problem sets, nine quizzes, two exams, six group labs, and a final group project. The point distribution is

Assessment	Percentage
Problem sets	30%
Quizzes	20%
Exams	25%
Project	20%
Class participation	5%

Grading thresholds are

	A	A-	B+	B	B-	C+	C	C-	D+	D
$\geq$	90	87	83	79	75	71	67	63	59	55

Problem Sets

You must follow the google python style guide for your coding assignments [3]. Assignments that do not follow the style guide will have points deducted. You can find additional tips and tricks in the MIT 6.001 style guide [4].

We will be using the gradescope tool to manage grading assignments `https://gradescope.com`. You should be enrolled in gradescope and received a welcome email. If you have not, contact the instructor to get and entry code.

Quizzes

The quizzes require short answers to problems and will be time limited to 5 minutes at the start or end of class.

Exams

The midterm exam will require you to translate and solve a statistical computing problem stated in words. You must state any assumptions that you make in your solution. You may be asked to write code or modify code without a computer. It is typical in a interview situation to sketch out your solution on paper or a whiteboard and the exams replicate this kind of assessment.

Project

You and your group members will do a detailed write-up and analysis of one of the group labs for your final project. The project will be presented to the class in a short presentation and submitted in a report at the end of the semester.

Class Participation

Attendance is required for all meetings. At selected points in the semester, participation will be assessed in terms of attendance and active participation. If you are not in class for a reason that's not accommodated, you will not receive class participation credit.

Makeup Policy

If you will be absent due to extenuating circumstances-including jury duty, military obligations, scheduled activities for other classes, the death of a family member, or verifiable health-related incapacity-remain responsible for meeting all class requirements and contacting the me in a timely fashion about making up missed work. Timely means within one week of the event. We'll meet to work out a make-up or alternative to make-up work.

Late Policy

If your assignment is late and you do not have an approved extenuating circumstance a deduction of 10% of the total for the assignment will be applied for each 24 hours late rounded up. After 72 hours if the assignment is not submitted, a grade of zero will be assigned.

Accomodation Statement

The University of Massachusetts Amherst is committed to providing an equal educational opportunity for all students. If you have a documented physical, psychological, or learning disability on file with Disability Services, you may be eligible for reasonable academic accommodations to help you succeed in this course. If you have a documented disability that requires an accommodation, please notify me within the first two weeks of the semester so that we may make appropriate arrangements.

Academic Honesty Statement

Since the integrity of the academic enterprise of any institution of higher education requires honesty in scholarship and research, academic honesty is required of all students at the University of Massachusetts Amherst.

Academic dishonesty is prohibited in all programs of the University. Academic dishonesty includes but is not limited to: cheating, fabrication, plagiarism, and facilitating dishonesty. Appropriate sanctions may be imposed on any student who has committed an act of academic dishonesty. Instructors should take reasonable steps to address academic misconduct. Any person who has reason to believe that a student has committed academic dishonesty should bring such information to the attention of the appropriate course instructor as soon as possible. Instances of academic dishonesty not related to a specific course should be brought to the attention of the appropriate department Head or Chair.

Since students are expected to be familiar with this policy and the commonly accepted standards of academic integrity, ignorance of such standards is not normally sufficient evidence of lack of intent.

Follow the link below for detailed information on the Academic Honesty Policy
`http://www.umass.edu/dean_students/codeofconduct/acadhonesty/`

References

[1] Ana Bell, Eric Grimson, and John Guttag. *6.0001 Introduction to Computer Science and Programming in Python. Fall 2016.* Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu. License: Creative Commons BY-NC-SA. URL: `https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/index.htm`.

[2] Eric Demaine and Srinivas Devadas. *6.0006 Introduction to Computer Algorithms. Fall 2011.* Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu. License: Creative Commons BY-NC-SA. URL: `https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/index.htm`.

[3] *Google Python Style Guide*. URL: `https://google.github.io/styleguide/pyguide.html`.

[4] *MIT Style Guide*. URL: `https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/assignments/MIT6_0001F16_StyleGuide.pdf`.

Course Schedule

No.	Date	DoW	Unit	Topic	Reading
1	09/06/2022	Tu	Func. Prog.	Hello World	ch 1 and 2.1
2	09/08/2022	Th	Func. Prog.	Branching Iterations and Functions	ch2.2 + 2.3.1 + 2.4 + 3.2 + 4.1-4.2+4.4-4.6
3	09/13/2022	Tu	Func. Prog.	Tuples Lists Aliasing	ch 5.1-5.3.1 + 5.4-5.5
4	09/15/2022	Th	Func. Prog.	Recursion and Dictionaries	ch 4.3 + 5.6
5	09/20/2022	Tu	Func. Prog.	Testing and Debugging	ch 6 + 7
6	09/22/2022	Th	Sim. and Vis	Generating simulation data	ch 14 + 16
7	09/27/2022	Tu	Sim. and Vis	Visualizing data	ch 11
8	09/29/2022	Th	Sim. and Vis	Company Valuation Lab	
9	10/04/2022	Tu	Sim. and Vis	Model selection	
10	10/06/2022	Th	Sim. and Vis	Casino lab	ch 10.3 (clrs ch 11)
11	10/11/2022	Tu			
12	10/13/2022	Th	RDBs	Constructing SQL queries	
13	10/18/2022	Tu	RDBs	Join Operations	
14	10/20/2022	Th		Midterm exam	
15	10/25/2022	Tu	RDBs	Movie recommendation lab	
16	10/27/2022	Th	Algorithms	Computational complexity	ch 9 (clrs ch 3)
17	11/01/2022	Tu	Algorithms	Searching and sorting algorithms	ch 10.1-10.2 (clrs ch 2)
18	11/03/2022	Th	Algorithms	Dynamic programming	
19	11/08/2022	Tu	Algorithms	Hash tables	
20	11/10/2022	Th	Data Structures	Heaps	(clrs ch 6.1-6.3)
21	11/15/2022	Tu	Data Structures	Heap sort	(clrs ch 6.4-6.5)
22	11/17/2022	Th	Data Structures	Binary search trees	(clrs ch 10.4 + 12.1)
23	11/29/2022	Tu	Data Structures	BST sort	(clrs ch 12.2-12.3)
24	12/01/2022	Th		Class presentation	
25	12/06/2022	Tu		Class presentation	
26	12/08/2022	Th		Final Review	
27		F		Final Exam	

Assessment Schedule

No	Date	Distr.	Coll.
1	09/06/2022	Pset 0 – Python setup	
2	09/08/2022		Pset 0 – Python setup
3	09/13/2022	Pset 1 – Functions	Branching quiz
4	09/15/2022		Tuples quiz
5	09/20/2022	Pset 2 – Testing	Pset 1 – Functions
6	09/22/2022		Recursion quiz
7	09/27/2022	Pset 3 – Simulation and visualization	Pset 2 – Testing
8	09/29/2022		Simulation quiz
9	10/04/2022	Pset 4 – Model selection	
10	10/06/2022		Pset 3 – Simulation and visualization
11	10/11/2022		Model selection quiz
12	10/13/2022		Pset 4 – Model selection
13	10/18/2022		Relational database quiz
14	10/20/2022	Pset 5 – Relational databases	Midterm exam
16	10/27/2022	Pset 6 – Complexity	Pset 5 – Relational databases
18	11/03/2022		Computational complexity quiz
19	11/08/2022		Pset 6 – Complexity
21	11/15/2022		Hash tables quiz
22	11/17/2022		Heaps quiz
24	12/01/2022		Group project
25	12/06/2022		Binary search tree quiz
27			Group project evaluation