Introduction to Computation for the Humanities and Social Sciences

CS 3
Chris Tanner
Lecture 24

I Got a [CS3] Story to Tell*

* the rare, unreleased original song
ANNOUNCEMENTS

- Projects **must** be submitted by Dec 3 at 11:59pm

TODAY

- complete in-class assignment and show it to a TA in order to get full HW credit (i.e., HW10)
- To have your project graded, sign up for a session w/ your Mentor TA (Dec 3 - Dec 10)
- Sign up for a presentation slot (Dec 4, 6, 11)
Lecture 24

- Everything we’ve learned
- Further options in Computation
- Further options in education
- Final Project Details
Lecture 1

What this course teaches

• Problem-solving workflow to build defensible arguments backed by relevant data sources and appropriate methods

• How to find and process various types of data (e.g., pre-formatted text, unstructured web-based text like Twitter feeds, etc).

• How to solve computational problems using Python (programming language)

• High-level understanding of core concepts (e.g., data structures, algorithms, computer science vs programming, machine learning, deep learning)
Lecture 1 — Course Topics

1. Problem Solving and Python
   - Forming computational problems
   - Finding datasets
   - Getting started with Python
   - Designing and writing programs
   - How data is stored and changed
   - Project 1: Computation

2. Textual Analysis
   - Iterating through text data
   - Data Structures
   - Building a concordance of large texts
   - Sentiment Analysis
   - Project 2: Text Analysis

3. Data Visualization and using APIs
   - Plotting Data in informative ways
   - Visualizing data on maps
   - Accessing Data from APIs
   - Project 3: Final Project

4. The Extended Landscape
   1. What is Machine Learning?
   2. Naive Bayes Classifier
   3. The Academic Landscape
1. Problem Solving and Python
   • Forming computational problems
   • Finding datasets
   • Getting started with Python
   • Designing and writing programs
   • How data is stored and changed
   • **Project 1:** Computation

2. Textual Analysis
   • Iterating through text data
   • Data Structures
   • Building a concordance of large texts
   • Sentiment Analysis
   • **Project 2:** Text Analysis

3. Data Visualization and using APIs
   • Plotting Data in informative ways
   • Visualizing data on maps
   • Accessing Data from APIs
   • **Project 3:** Final Project

4. The Extended Landscape
   1. What is Machine Learning?
   2. Naive Bayes Classifier
   3. The Academic Landscape
Computation vs Computer Science vs Programming vs Python
Computation vs Comp Sci vs Programming vs Python

- **Computation** — a calculation; a specific operation, or set of operations, performed by a well-defined model or function.

- **Computer Science** — a field of study which is deeply centered around computation, including the interest of both the computation itself, and applying computation towards various problems and tasks.

- **Programming** — the act of typing words (instructions) in a specific computer language, in a way that a computer can understand and execute (i.e., compute). This is the vehicle of communication within the field of computer science.

- **Python** — a specific, high-level yet powerful and easy-to-use programming language
What is Computer Science?

- How do you calculate which a photo from an unprecedented point-of-view would look like? (graphics)
- How can you very quickly calculate the sub-sections of human DNA which are most likely to concern pancreatic cancer? (comp. bio)
- How can you robustly handle internet traffic when 1,000 computers try to simultaneously connect to a website? (networking)
- How can you store 1 billion names such that one can quickly determine if any queried name exists in it? Or the closest match? (databases)
- How can you determine if two names within a document refer to the same underlying person or not? (natural language processing. my dissertation)
Problem Solving Workflow: Define Question

- Problem Solving Workflow
- Our Data

1. Define question
2. Select appropriate dataset
3. Design a valid method
4. Analyze your results
5. Communicate your findings
Python Programs

- Data Structures: single-valued, lists, dictionaries, sets, np.arrays, np.matrices
- Data Types: strings, ints, floats, booleans
- Control Flow: if-statements, for-loops, while-loops, functions
- Regex

Input() from cmd line

Files on the computer

Web pages

External apps (APIs)
Lecture 24

- Everything we’ve learned
- Further options in Computation
- Further options in education
- Final Project Details
Life after CS3
A soft, high-level overview of a bunch of topics, from the Internet, Databases, Multimedia, What is Programming?, What is AI?
A soft, middle-level overview of a bunch of topics, but includes slightly more technical things like Machine Learning, NLP, Computer Vision.
Similar to CS3 but is geared toward engineering concentrators, scientific problems, and using an annoying language called MATLAB, too.
One of the most popular Intro sequence classes. A lot of the material we learned about! They take an object-oriented approach to programming (uses Classes, like we touched on last lecture) and other data structures (e.g., Trees). Covers making graphical programs, too.
One of the popular Intro Sequence courses to CS. Less about programming, more about theoretical stuff, like data structures and algorithms.
One of the popular Intro Sequence courses to CS.

Functional programming course that introduces students to data types, recursion, and basic data structures and algorithms. The course starts in Racket and transitions into OCaml, which are both functional programming languages. The course does not expect any prior experience and is a great introduction to programming and how a program flows. The difficulty of the course significantly picks up in the latter half of the semester as the harder projects are assigned. The course is a significant time commitment, but I would definitely recommend it to people who are interested in computer science and like to solve problems logically.
One of the popular Intro Sequence courses to CS.

is the second semester course that is paired with cs17. It teaches **object-oriented programming (OOP)** and focuses significantly on **data structures, algorithms, and good coding principles**. The course starts by teaching OOP and some data structures in Java and later transitions into Scala, a programming language based on Java but also supports functional programming. cs18 also has projects that are similar to real-world programming and software engineering that incorporate the data structures and algorithms that students learn in the class. Projects include: a client and server, a search engine, and a GUI browser.

Take this course so you can bother Dylan Sam for another semester.
One of the popular Intro Sequence courses to CS.

Fast-paced course in Pyret, a **functional programming language**. Professor is an amazing lecturer and you'll learn a lot, but has no real applications unless you take upper div CS courses or work at Jane Street. Technically no prior experience necessary and works well as an independent course, but not for the faint of heart.
CS111

One of the popular Intro Sequence courses to CS.

cs111 is the first course in a three part intro sequence. It is slower-paced than cs15 and cs17, and teaches Pyret and Python. The next courses in the sequence (cs112 and cs113) are not being taught next semester, though students who wish to continue an intro sequence can take cs18 if they do some work over winter break.
Lecture 24

- Everything we’ve learned
- Further options in Computation
  - Further options in education
- Final Project Details
Further education

Master’s

• Take harder classes which **build upon a foundation**. Easier to get A’s, though. Nobody gets C’s. Professors treat you more like an adult, and learning is way more of a focus than grades.

• Some experience with research.

• Degree is useful for (1) career advancement, to show some distinction from having “only” a Bachelors; (2) your own intellectual curiosity, especially with the opportunity to see what a further career in research (PhD) would look like.
Further education

PhD

- Typically 5-8 years. It’s a **research** degree. Almost like a job, and you’re paid a yearly salary of $20k - $40k to learn how to do independent research.
- Point is to extend the world’s knowledge about a very, very specific problem that interests you. Your dissertation is your document showing new/better stuff that **nobody** else has done before.
- Classes have almost no importance other than your own individual growth and interest. Same grad classes as Master’s. Typically only take courses your first 2 years.
- Your relationship with your adviser means everything. It’s kind of like a marriage. Picking a good fit for an adviser is more important and correlated to your success than the prestige of the school. It’s okay to change advisers, but if so, should happen within the first 2.5 years.
Further education

Choices

• Once you leave academia, it’s hard to get back in, especially for PhD.

• Obviously, the choice of your field highly dictates which level of education is necessary for whatever you want to do — largely influenced by how much your field emphasizes using a skillset vs expanding upon current knowledge (not just a product of needing to distinguish yourself from your peers).
Lecture 24

- Everything we’ve learned
- Further options in Computation
- Further options in education
- Final Project Details
Final Project

You decide the best projects
Final Project

The Dundies!

Holiday Toy List: Kid Picks

You purchased this item on November 28, 2018.
View this order

Gold Award Trophies, 6" Trophy Statues – Oscar Trophy Award for Party Celebrations, Ceremonies, Appreciation Gift, Sport Awards, Olympic Award, Academy Awards, Oscar Party Supplies, (Set of 6)

by Nellblu

4.5 out of 5 stars – 182 customer reviews | 7 answered questions

prime

Your cost could be $0.00. Eligible customers get a $10 bonus when reloading

- EVERYONE'S A WINNER WHEN YOU HAND OUT THESE LITTLE TROPHIES! Crafted to resemble the one and only Oscar. You too can feel like a STAR a recognition for winners and participants alike.
- THESE TROPHY AWARDS ARE A GREAT WAY TO SHOW APPRECIATION WITHIN YOUR BUDGET. Appreciate your contributors and players with awards they will be proud to display!
The Dundies!

- **The RISD-Level Award** (Best Visualization)
- **The Winston Churchill Award** (Best Oral Presentation)
- **The Ben Franklin Award** (Most impressive Problem Solving)

Cast your vote for each of these three categories (can’t vote yourself).

The winners will receive 3 extra credit points.
LAB TIME