Welcome to W200!
Introduction to Data Science Programming Live Session

Week 1

Taylor Martin

Section 8

Remind me to start recording! ☺

Berkeley
SCHOOL OF INFORMATION
MIDS • DATA SCIENCE PROGRAM

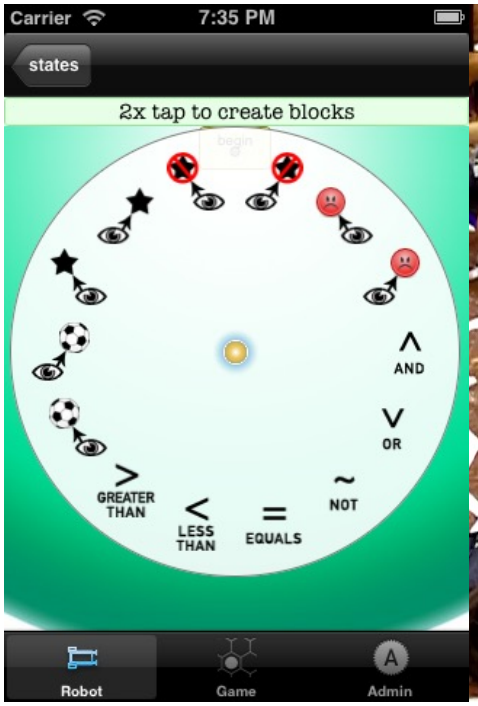
Today

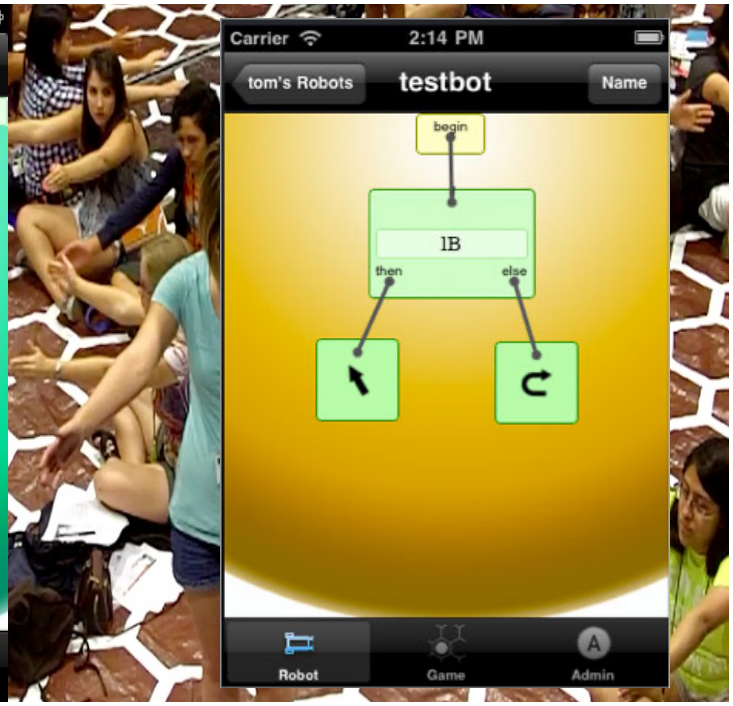
- Introductions
- Survey results
- DEIB
- Gen AI
- How we'll work in this class
- Breakouts to dive deeper into:
  - Command Line
  - Manipulating Files & Directories
  - DS Interview Questions CLI & git

Berkeley
SCHOOL OF INFORMATION
MIDS • DATA SCIENCE PROGRAM

Introductions

- Name
- Semester started MIDS?
- Where are you?
- What do you do?
- Quick version of how you came to data science





Berkeley
SCHOOL OF INFORMATION
MIDS • DATA SCIENCE PROGRAM

Survey Results

Variety of Experience

- Fast pace
- New
  - Grab it while it flies by
  - Keep learning
    - Simple daily practice
    - Projects
    - Python Crash Course by Eric Matthes
- Experienced
  - Learn by teaching
  - Challenge yourself
  - Challenge projects in solid-garbanzo

Course Sequence

- Week 1: Computing Environment
  - *OS/Unix, Github, IDEs*
- Weeks 2-6: Fundamentals
  - *Data and sequence types; control of flow, algorithms & functions, complexity; Big Theta ‣*
- Weeks 7-8: Object-Oriented Design
  - *Classes, OOD, OOP ‣ Project 1*
- Weeks 9-14: Data Analysis
  - *Text & Binary Data ‣ Numpy ‣ Pandas ‣ Plotting ‣ Files & Testing ‣ Team Data Analysis & Presentation ‣ Project 2*

Course repo

- https://github.com/htmartin/solid-garbanzo
- A fork = a copy of a repository that you manage.
- Lets you
  - experiment with the code & make changes
  - Practice git add, commit, push flow
  - Without affecting main repo
  - propose those changes to the original project (we'll address this later)
- Demo
- See forking-guide.md in solid-garbanzo

Diversity, Equity, Inclusion, and Belonging

- I'm always advised to make a statement regarding DEIB at the start of my classes.
- That feels like CYA/checking a box.
- More importantly, it takes all of us,
- So instead:
- We're going to build an agreement about what we all need to do to ensure that we co-construct a classroom experience that:
  - Celebrates diversity
  - Prioritizes equity
  - Enables inclusion and belonging for everyone.
- Activity: See deib-activity.md
  - List what students and instructors need to do.
  - I'll run some nlp on your versions and mine.
  - Next week we'll discuss the result to adopt as our class DEIB agreement.

Berkeley
SCHOOL OF INFORMATION
MIDS • DATA SCIENCE PROGRAM

Gen AI Tools Breakout

- How can they help you for prepare for working as a data scientist?
- How can they hinder you?

Use Gen AI for learning basic Python

- Experiment with your own prompts for
  - Error interpretation
  - Explanation for any concept you're stuck on, give other examples
  - Practice problems
  - Extensions
    - How would you extend this concept?
    - Project or problem to practice that
  - Tons more when we get to the data science section of the course.

Notes about our section

- Live session is just that – live
- Class participation
- A caveat about experimenting
- Office hours M 11-12 PT
  - Please slack me if you're coming by Sunday night.
  - Take pity so I don't have to sit there for an hour for maybe someone to show up ☺
  - If possible, slack your main questions too
- Order of activities
  - Read & watch async
  - Live session
  - HW

Breakout: CLI scavenger hunt

- Create a scavenger hunt with 4-5 clues
- Use commands you learned this week and/or ones you already knew.
- without the answers :-)
- We'll check in at 10 min, hopefully groups are ready to trade
- Trade with another group.
- Go hunting! cli-scavenger-hunt.md

Or Breakout: htop exploration

- Install htop if needed- htop-exploration.md
- Write a set of questions that can be answered with htop.
  - Example: Which processes are using the most CPU?
- We'll check in at 10 min, hopefully groups are ready to trade
- Trade with another group.
- Go htoping! installing -`htop`.md

CLI breakout wrap up

- The Linux Command Line by William Shotts (https://linuxcommand.org/tlcl.php)
- ls –al
  - Files and directories that start with a dot (.git) are hidden by default.
  - These files are usually configuration files or directories used by the system or various applications.
- -r
  - Did you try copying or removing directories? (cp or rm)
  - What happened?
  - Need to be recursive
    - cp -r
    - rm -r
- Control c will get you out of many a sticky situation
- Other super handy critters:
  - which _____ (vim, python…)
  - tree
  - top and htop

Breakout: Interview Questions

- General Knowledge
  - 1. What is version control, and why is it important in data science?
  - 2. How can you resolve merge conflicts in Git?
  - 3. Explain the difference between Git and GitHub.
- Coding/Scripting
  - 1. Write a Bash command to find and count the number of `.csv` files in a directory and its subdirectories.
  - 2. How can you revert the last commit in Git while keeping the changes in your working directory?
- Work on the CLI/Revision Control Interview Questions in your groups - cli-revision-control-interview-questions.md

Interview Questions Breakout wrap up

- These were a subset of the questions
- I'll add the full list to the repo after class fyi.

Async presentation Week 2

- X,Y, Z
- Don’t try to recap everything
- Looking for 5 min presentation (10 max)
- What are the 3-5 Big Ideas of the material?
  - Provide a focusing conceptual “lens” for any study
  - Connect and organize many facts, skills, and experiences
  - Point to ideas at the heart of expert understanding of the subject
- For example:

Week 1 Big Ideas

- The command line interface (CLI) offers a more direct and scriptable way to control and automate tasks compared to graphical user interfaces (GUIs) which rely on visual navigation and mouse clicks by allowing users to interact with the computer through text commands.
- Revision control (e.g. Git) enables tracking and managing changes to code or files over time in a distributed manner.
- Programming languages, like recipes and knitting patterns are tools used to create and express algorithms. They have sequential steps and the properties of clarity and precision, reproducibility, and modularity.
- Computer architecture layers (hardware at the bottom, the operating system in the middle, and applications on top) simplify user interaction with the computer by providing standardized abstractions at each level.

Expansion example:

- Computer architecture layers simplify user interaction with the computer by providing standardized abstractions at each level.
  - 1. Hardware Abstraction: The operating system provides a simplified and consistent interface to interact with various hardware components (e.g., CPU, memory, storage) without needing to know the specifics of the hardware.
  - 2. File System Abstraction: The operating system abstracts the underlying data storage into files and directories, allowing users to organize and manage data in a standardized way regardless of the actual storage medium.
  - 3. Process Abstraction: The operating system manages the execution of programs, allowing multiple applications to run simultaneously without interfering with each other.
  - 4. User Abstraction: The operating system defines different types of users (regular users, system users, superusers) and manages permissions and access controls, ensuring secure and controlled access to system resources.
  - 5. Network Abstraction: The operating system provides a consistent way to interact with network resources, abstracting the complexities of underlying network protocols and hardware.

W200.8 Martin Week 1 Survey

- https://forms.gle/v234WXcqeh5dNUGN7

A final word

- Week 2 async is a whopper.
- Most intro python courses would cover the material in at least 2 weeks sometimes more...
- You'll use it the rest of the course, so lots of chances to practice.
- Don't think you need to completely get it all

Extras

Breakout: CLI from Jupyter Notebooks

- What's your current working directory?
- List of all files in this directory
- And a list that shows the full file names, in alphabetical order, with file permissions (read, write, execute) for owner, group, world.
- Who are you logged in as? Where are you in the file directory?
- Compare changing directory using ! and %
  - What's the difference?

Breakout follow up

- Issue commands to python
  - .py files
  - interactive terminal
  - IDEs
  - or Jupyter notebook
- Commands: OS -> Python
- Python executes the commands
- Results: Python -> application.
- Jupyter Notebook is an application
- From JNB to bypass python and speak directly with the OS using Jupyter we need the ! or % commands.
  - ! vs % in JNB - What's the difference?