W200 Python Fundamentals for Data Science

Course Overview

Course Overview | Instruction Team

Course Overview | Agenda

Review Goals

Course Basics

The Student Experience

The Curriculum

Discussion

Activity

Student Showcase

Wrap Up

Review Goals | The Wisdom of Drew

Develop a shared understanding of program curriculum

Identify outcomes for each course

Identify gaps and overlaps between courses

Share strategies around instruction and tech use

Assume others know very little about your course

Course Overview | Agenda

Review Goals

Course Basics

The Student Experience

The Curriculum

Discussion

Activity

Student Showcase

Wrap Up

Course Basics | Big Picture

An introduction to programming ***and*** an introduction to dataset manipulation, all in Python

Focus is on theory as well as syntax

Students leave able to solve programming problems, comfortable in a variety of Python tools, and able to to troubleshoot and learn new libraries on the fly

Students leave comfortable using git and GitHub on team projects

Course Basics | First 8 Weeks - Programming

Unit 1 | Introduction, the Command Line, Source Control

Unit 2 | Starting Out with Python

Unit 3 | Sequence Types and Dictionaries

Unit 4 | More About Control and Algorithms

Unit 5 | Functions

Unit 6 | Complexity

Unit 7 | Classes

Unit 8 | Object-Oriented Programming

Individual Object Oriented Project

Course Basics | Last 6 Weeks - Data Analysis

Unit 9 | Working With Text and Binary Data

Unit 10 | NumPy

Unit 11 | Data Analysis With Pandas

Unit 12 | More Analysis With Pandas

Unit 13 | Testing

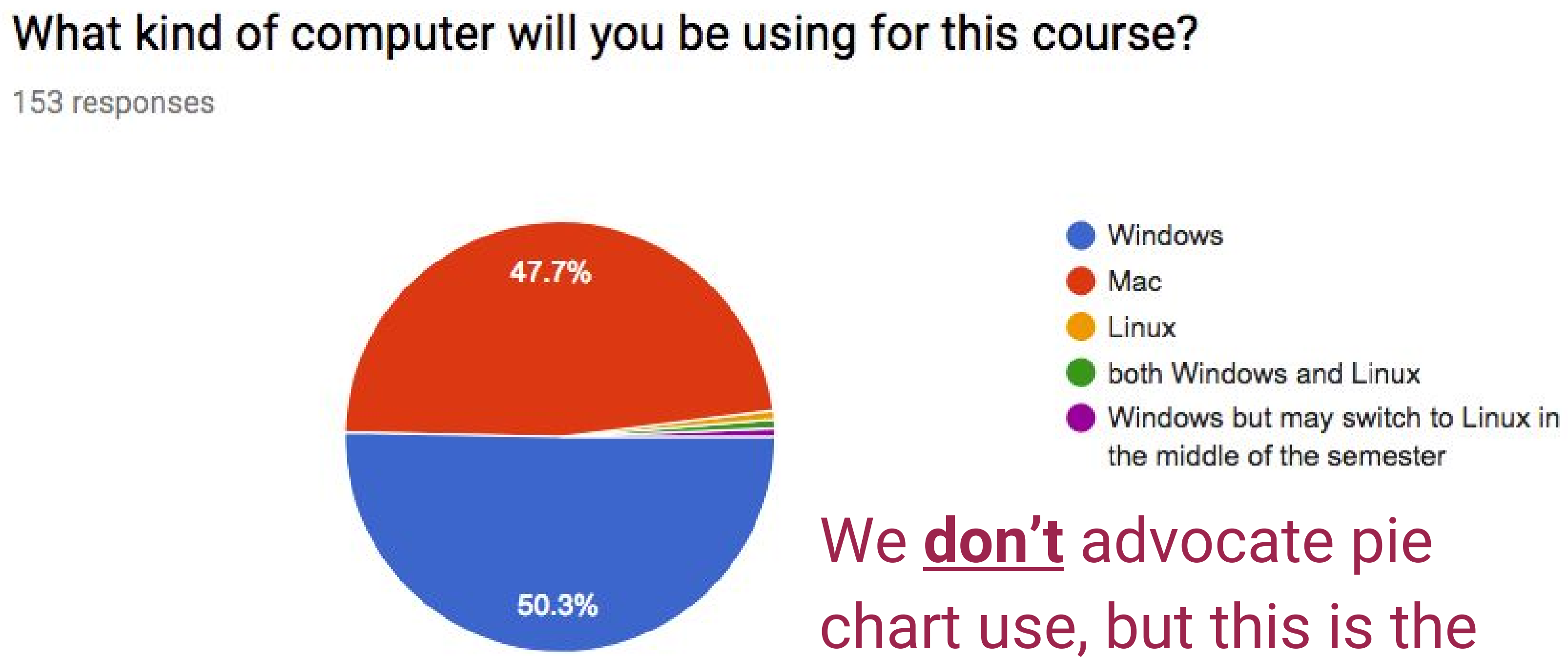
Data Analysis Group Project

Course Basics | Student Profile

Students come from a wide variety of programming backgrounds

We start each semester with a “diagnostic survey” to get a handle on the current class

Course Basics | Student Profile



What kind of computer will you be using for this course?

153 responses

We **don’t** advocate pie chart use, but this is the Google Sheets default.

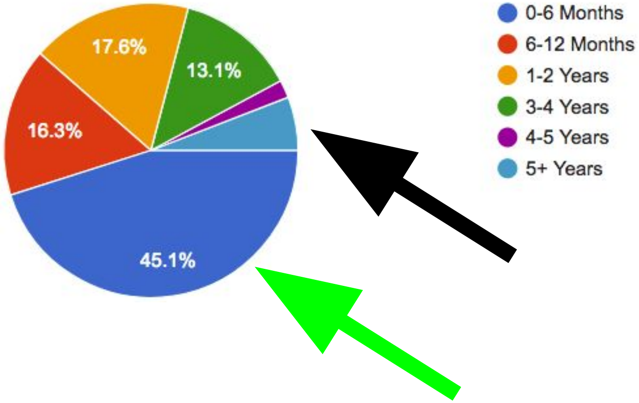
Course Basics | Student Profile

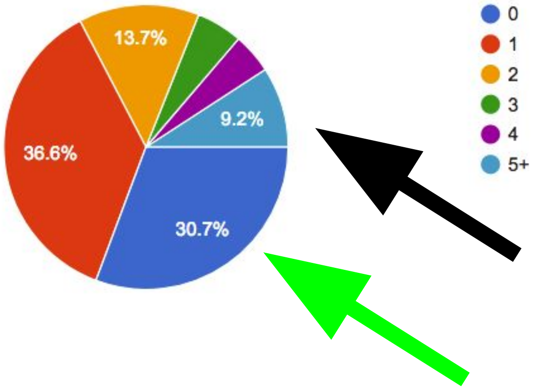
How many programming classes have you taken in a school environment?

153 responses

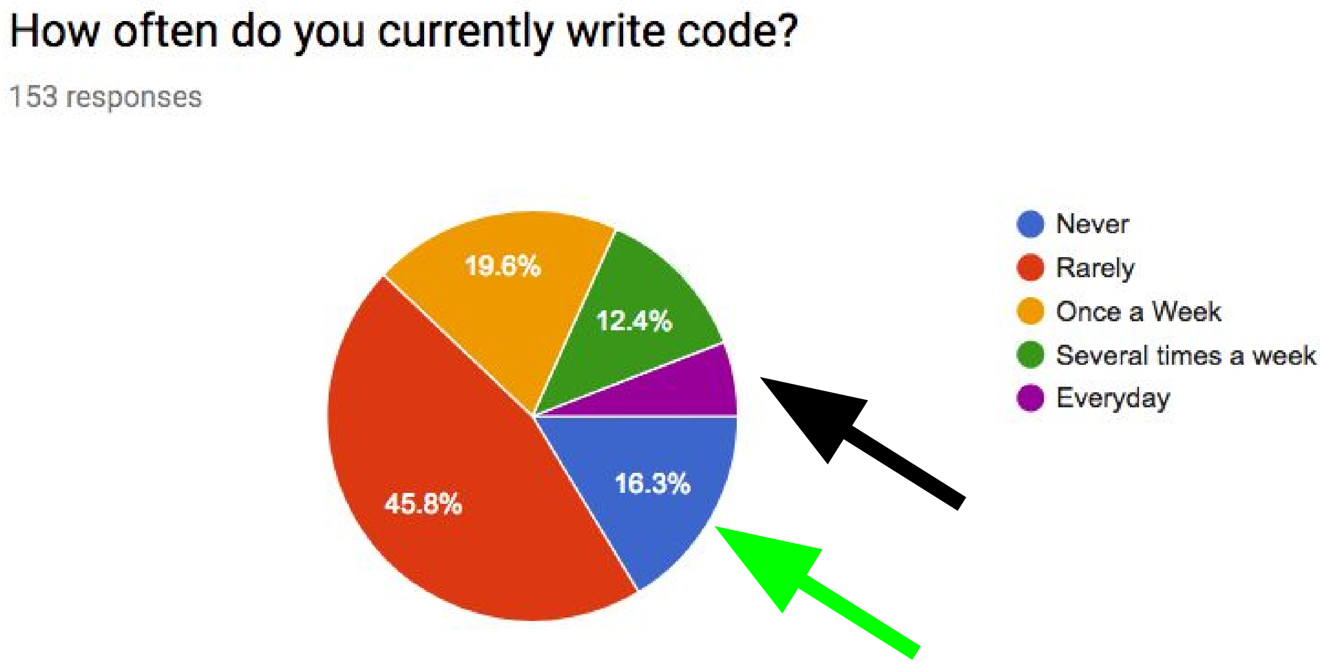
How much programming experience do you have?

153 responses





Course Basics | Student Profile



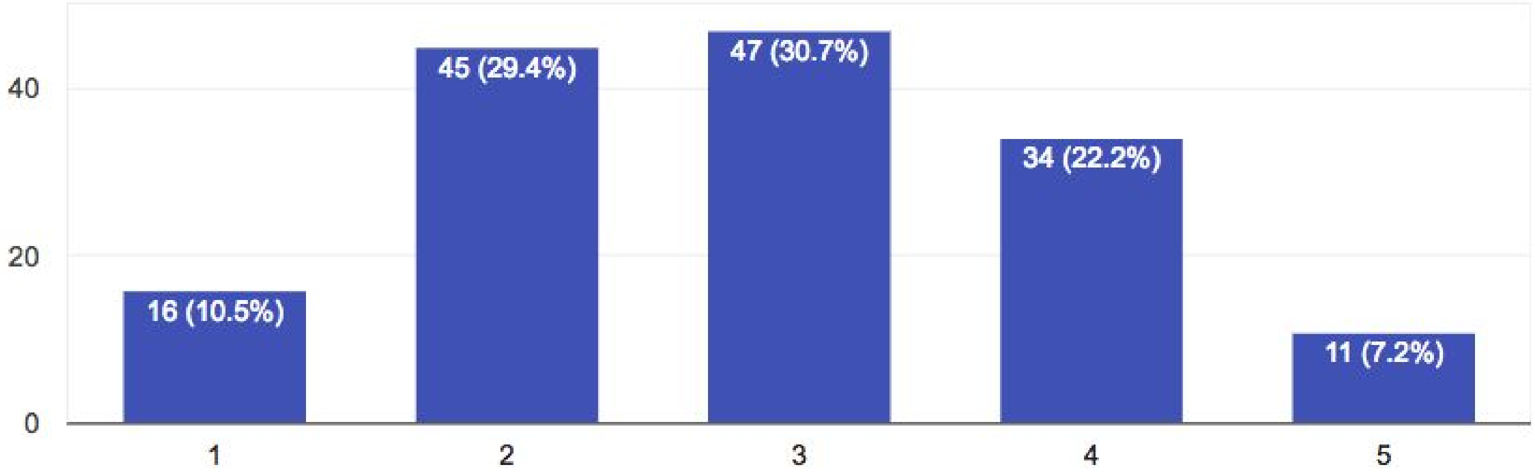
How often do you currently write code?

153 responses

Course Basics | Student Profile

How would you rate your experience with troubleshooting computer problems?

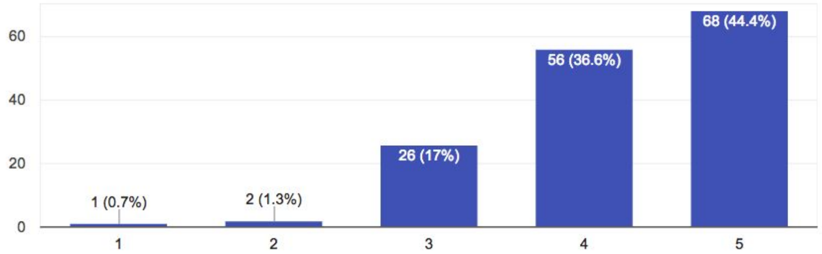
153 responses



Course Basics | Student Profile

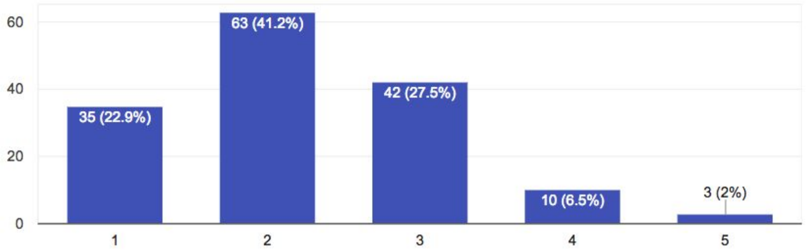
How would you rate your experience with excel?

153 responses



How would you rate your experience with programming?

153 responses



Course Overview | Agenda

Review Goals

Course Basics

The Student Experience

The Curriculum

Discussion

Activity

Student Showcase

Wrap Up

The Student Experience | Grading

Weekly Homework (30%)

Project 1 object oriented, individual (20%)

Project 2 data analysis, group (20%)

Participation (10%)

Midterm (10%)

Final (10%)

The Student Experience | Schedule

https://docs.google.com/spreadsheets/d/1s1FZzV-8Mw4FjYZc_2j-knD76P8ehLMKt4w7UHYN5_I/edit?usp=sharing

The Student Experience | GitHub

Installation:
https://github.com/MIDS-INFO-W18/Installation

All Course Asynchronous Materials:
https://github.com/MIDS-INFO-W18/Course-Syllabus

Homework:
https://github.com/MIDS-INFO-W18/assignments_upstream_fall17

Free for All:
https://github.com/MIDS-INFO-W18/github-playground

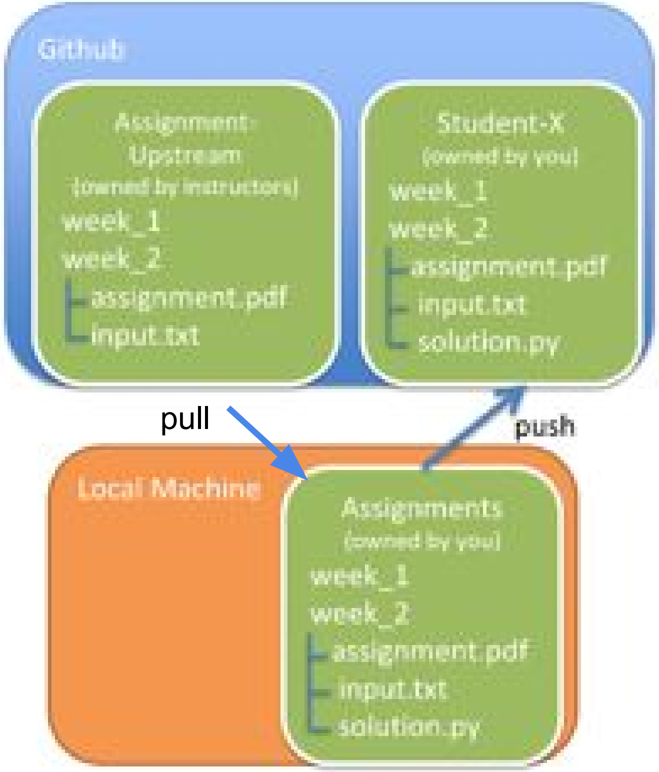
The Student Experience | Submitting Homework

Students **pull** homework from a shared repository, and **push** their solutions to a private repository that only they can see.

We use GitHub “teams” to control access across our repositories.

Demo:
https://github.com/MIDS-INFO-W18

The Student Experience | Submitting Homework



The Student Experience | Behind the Scenes

Instructor Full Content:
https://github.com/MIDS-INFO-W18/assignments-instructors

Includes:

- Upcoming homework assignments, projects, and exams (not yet released)
- Solution keys
- Files under development
- “Fun” code we accumulate over the years!

The Student Experience | Truly Cross-Section

Students have access to a shared Google Group across all sections. Instructors promise to respond to questions in a timely fashion

https://groups.google.com/forum/#!forum/w200-python-2017-fall

Students are welcome to join any office hours for the course

Gunnar/Chris get to know each other’s students, to a point

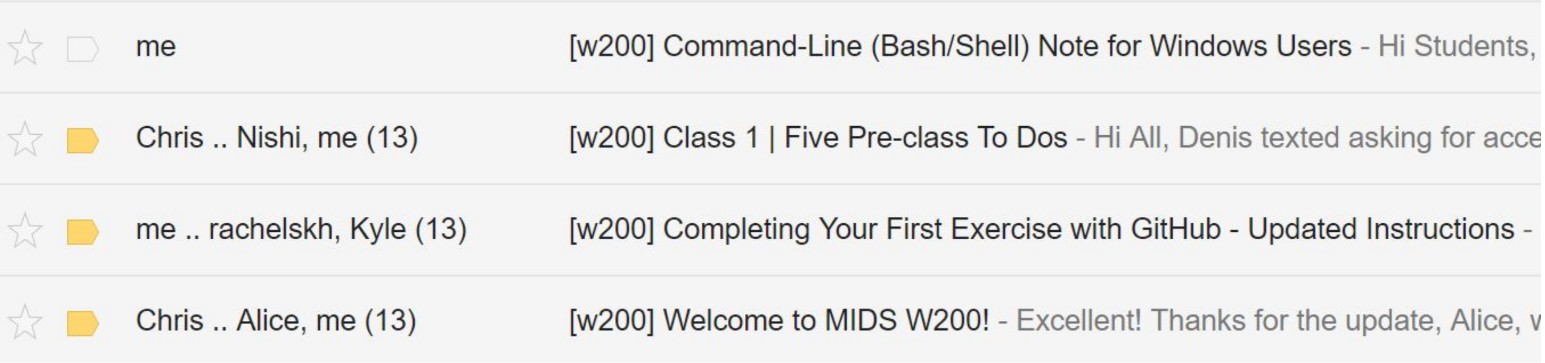
Paul joins for final project presentations when possible

Students encouraged to use Slack to meet each other

https://ucbischool.slack.com/messages/C5AL99BU6/

The Student Experience | Course Kickoff

Course is kicked off each year using a set of email “templates” developed over time.



me	[w200] Command-Line (Bash/Shell) Note for Windows Users - Hi Students,
Chris .. Nishi, me (13)	[w200] Class 1 | Five Pre-class To Dos - Hi All, Denis texted asking for acce
me .. rachelskh, Kyle (13)	[w200] Completing Your First Exercise with GitHub - Updated Instructions -
Chris .. Alice, me (13)	[w200] Welcome to MIDS W200! - Excellent! Thanks for the update, Alice, v

The Student Experience | Course Kickoff

Students complete a pre-class 1 assignment

We pay close attention in the first few weeks to make sure nobody gets left behind, including individual follow-up as needed

No Python taught in the first week

Students learn how to use git, GitHub, and the command line. These tools are then used throughout the course

Students who fail HW 1 are asked to resubmit

The Student Experience | Live Sessions

In live sessions, we:

- Use a presentation such as this one
- Recap major points of the course async
- Demo important concepts on our own machines
- Start discussion around tricky concepts
- Provide breakout activities for most weeks

New this semester: The submarine

Course Overview | Agenda

Review Goals

Course Basics

The Student Experience

The Curriculum

Discussion

Activity

Student Showcase

Wrap Up

The Curriculum | First 8 Weeks - Programming

Unit 1 | Introduction, the Command Line, Source Control

Unit 2 | Starting Out with Python

Unit 3 | Sequence Types and Dictionaries

Unit 4 | More About Control and Algorithms

Unit 5 | Functions

Unit 6 | Complexity

Unit 7 | Classes

Unit 8 | Object-Oriented Programming

Individual Object Oriented Project

The Curriculum | Programming

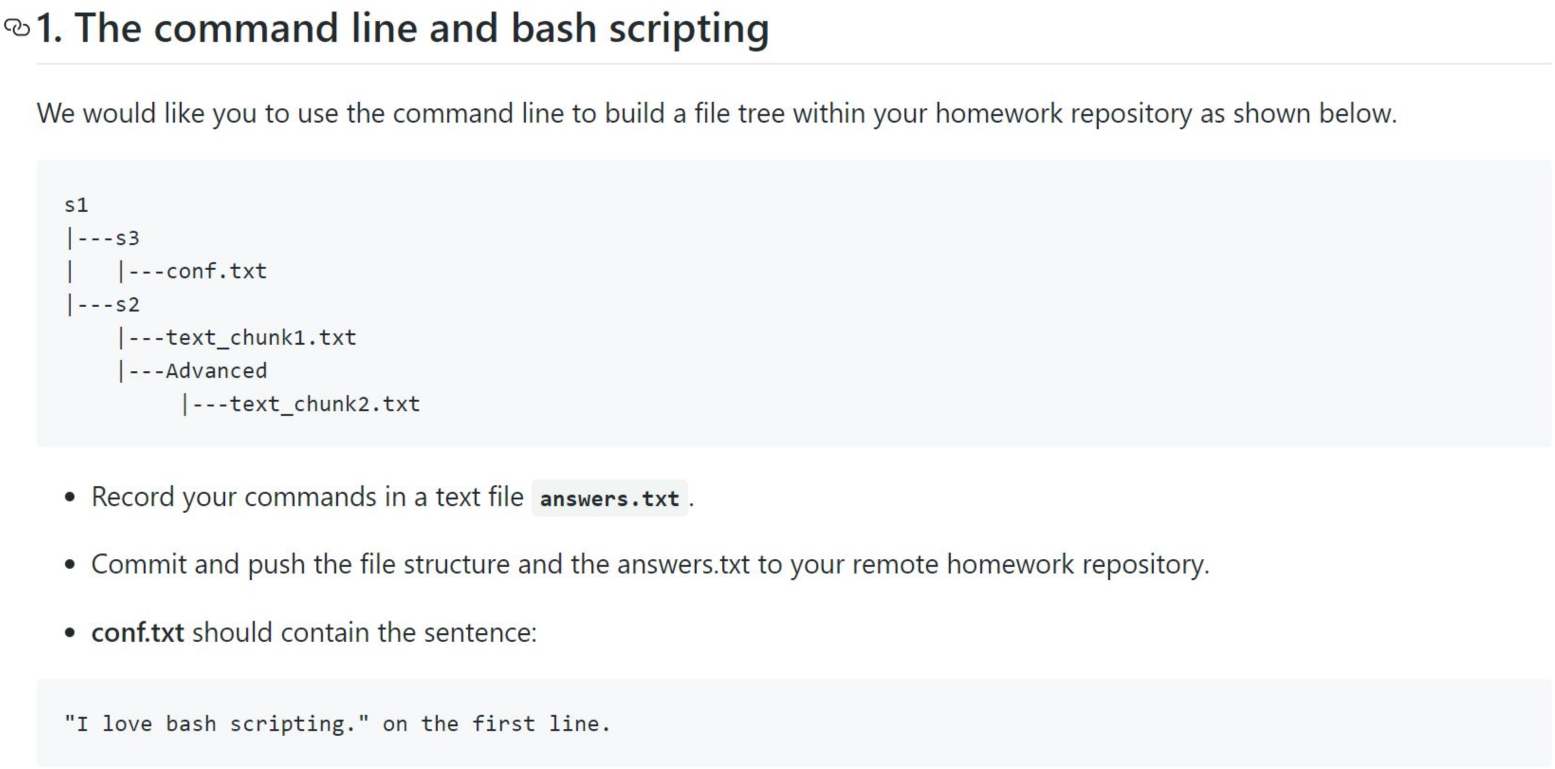
The first eight weeks of our course are designed to challenge students, culminating in a student-designed project and a midterm.

Students are exposed to a wide range of programming tools, and the homework gets progressively harder as more tools must be integrated to solve problems

Important theory is also covered: pseudocoding, algorithm complexity, recursion

We thought it would be fitting (and fun!) to show you context into the course through an example homework problem from each week.

Programming | Homework Examples - Unit 1



1. The command line and bash scripting

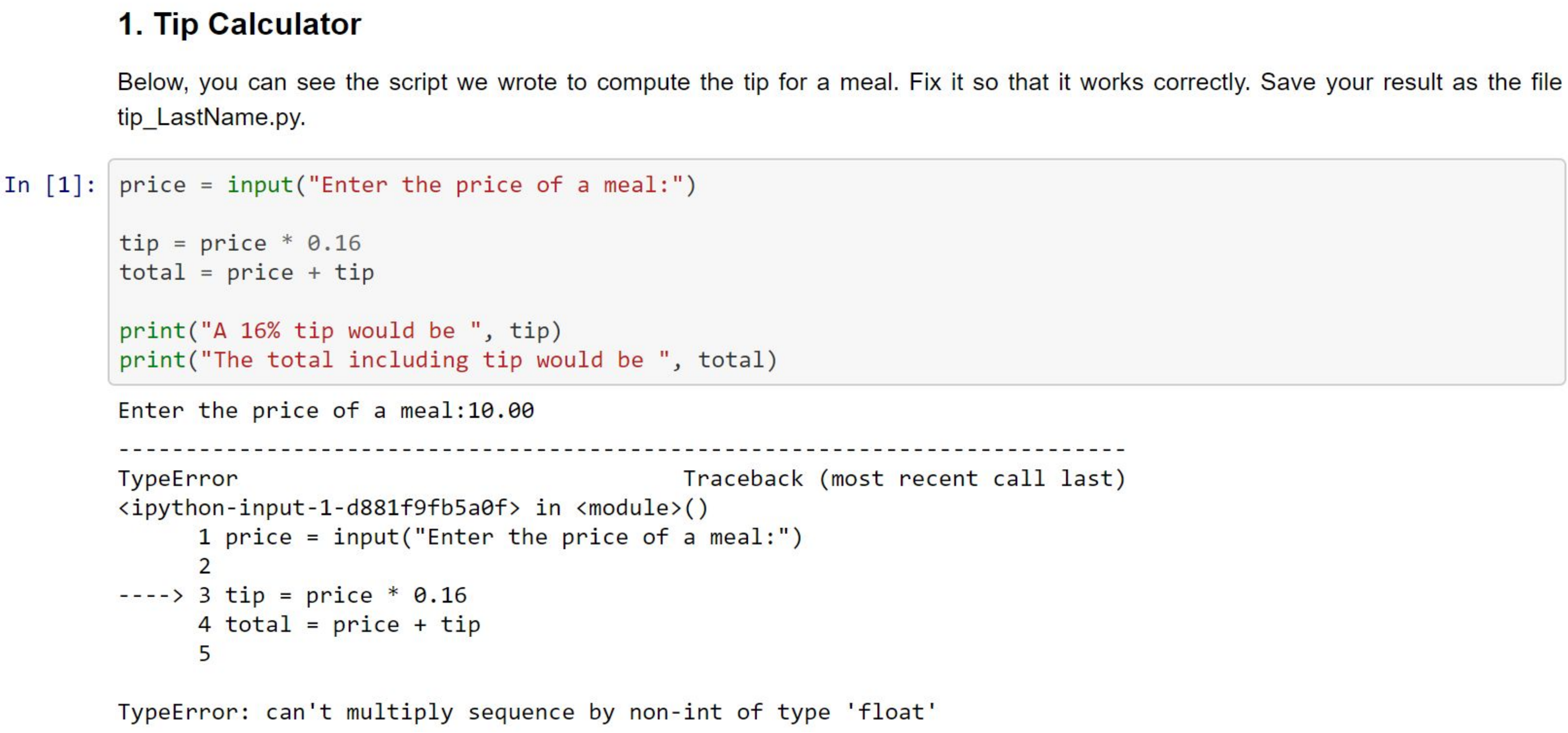
We would like you to use the command line to build a file tree within your homework repository as shown below.

```
s1
|---s3
|   |---conf.txt
|---s2
    |---text_chunk1.txt
    |---Advanced
         |---text_chunk2.txt
```

- Record your commands in a text file `answers.txt` .
- Commit and push the file structure and the answers.txt to your remote homework repository.
- **conf.txt** should contain the sentence:

```
"I love bash scripting." on the first line.
```

Programming | Homework Examples - Unit 2



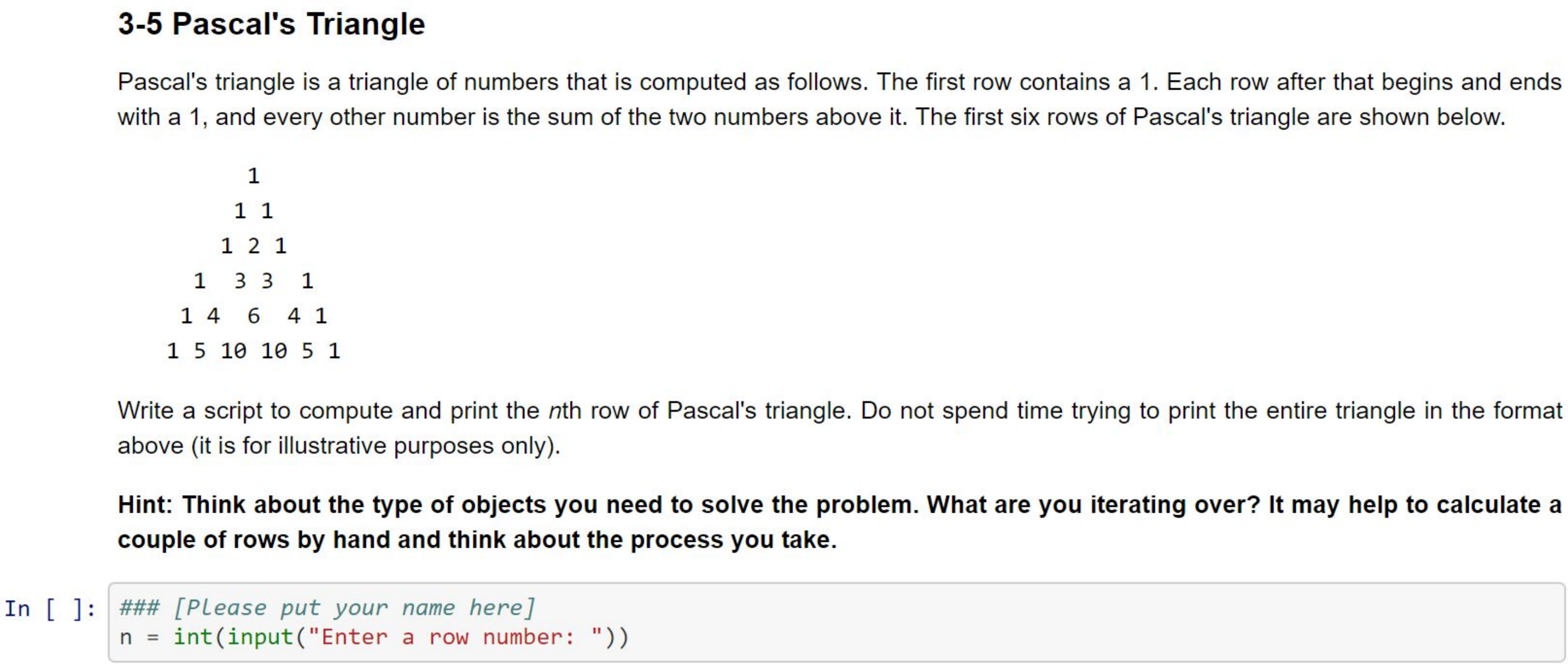
1. Tip Calculator

Below, you can see the script we wrote to compute the tip for a meal. Fix it so that it works correctly. Save your result as the file tip_LastName.py.

```
In [1]: price = input("Enter the price of a meal:")

        tip = price * 0.16
        total = price + tip

        print("A 16% tip would be ", tip)
        print("The total including tip would be ", total)
```

```
Enter the price of a meal:10.00

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-1-d881f9fb5a0f> in <module>()
      1 price = input("Enter the price of a meal:")
      2 
----> 3 tip = price * 0.16
      4 total = price + tip
      5 

TypeError: can't multiply sequence by non-int of type 'float'
```

Programming | Homework Examples - Unit 3



3-5 Pascal's Triangle

Pascal's triangle is a triangle of numbers that is computed as follows. The first row contains a 1. Each row after that begins and ends with a 1, and every other number is the sum of the two numbers above it. The first six rows of Pascal's triangle are shown below.

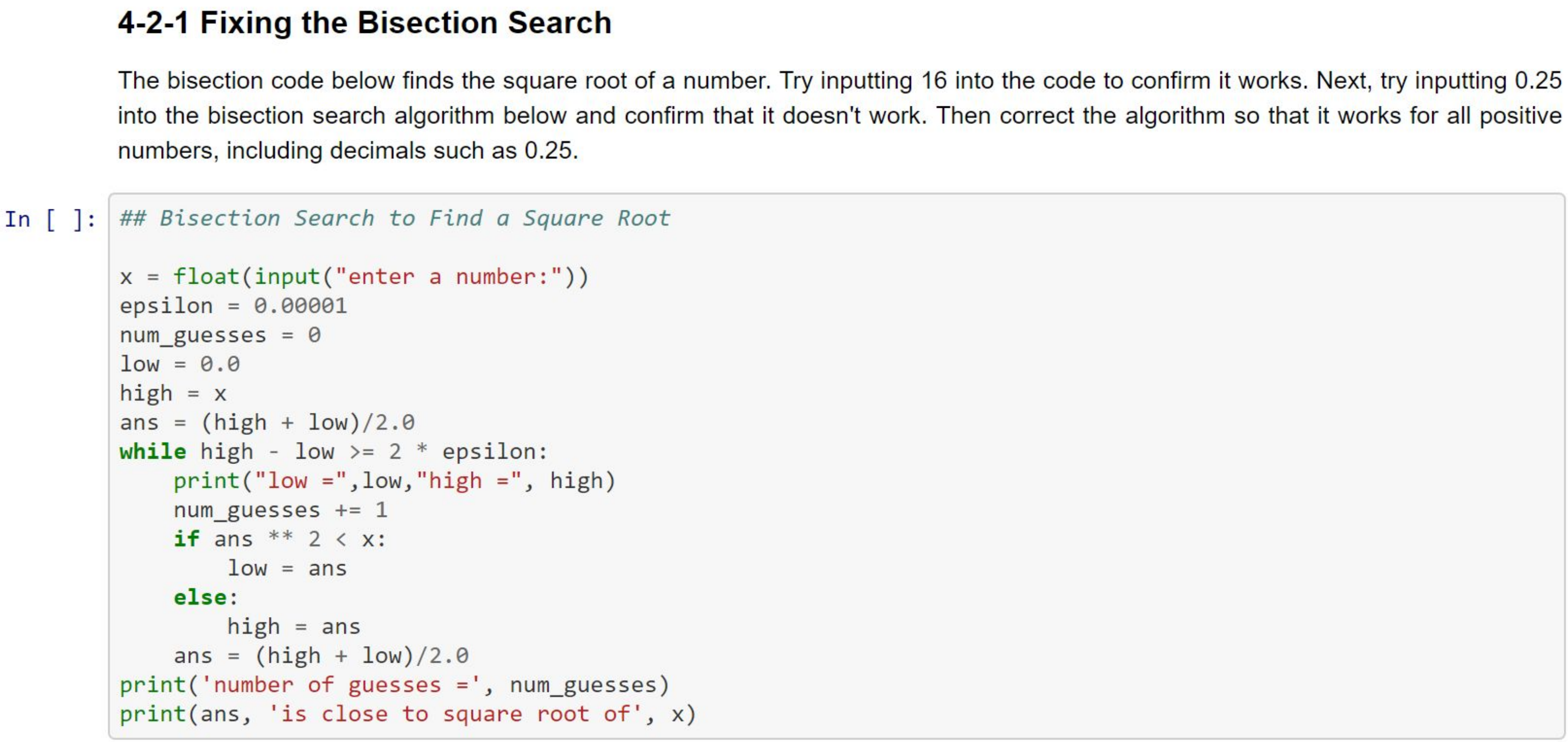
```
      1
     1 1
    1 2 1
  1  3 3  1
 1 4  6  4 1
1 5 10 10 5 1
```

Write a script to compute and print the *n*th row of Pascal's triangle. Do not spend time trying to print the entire triangle in the format above (it is for illustrative purposes only).

Hint: Think about the type of objects you need to solve the problem. What are you iterating over? It may help to calculate a couple of rows by hand and think about the process you take.

```python
### [Please put your name here]
n = int(input("Enter a row number: "))
```

Programming | Homework Examples - Unit 4



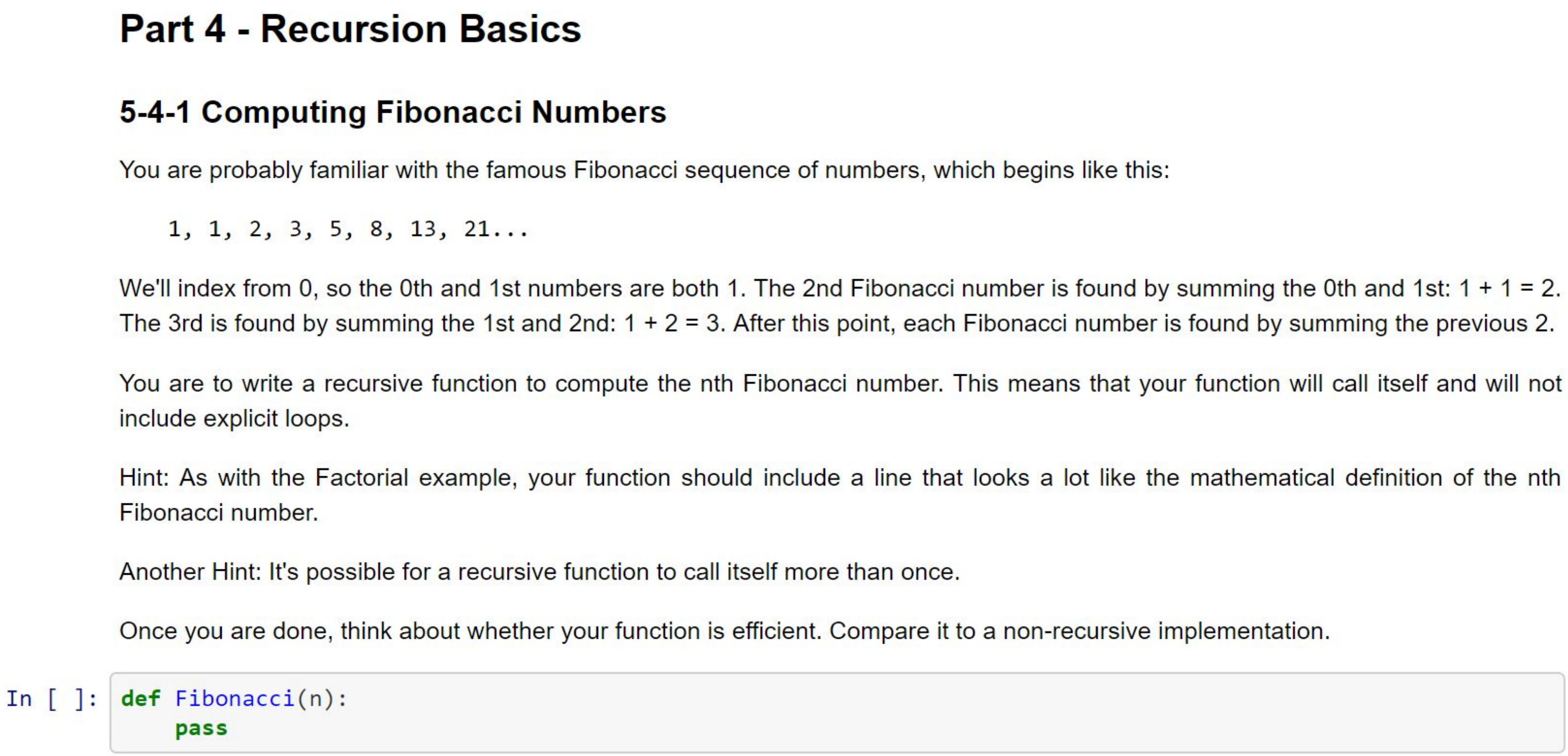
4-2-1 Fixing the Bisection Search

The bisection code below finds the square root of a number. Try inputting 16 into the code to confirm it works. Next, try inputting 0.25 into the bisection search algorithm below and confirm that it doesn't work. Then correct the algorithm so that it works for all positive numbers, including decimals such as 0.25.

In [ ]:

```
## Bisection Search to Find a Square Root

x = float(input("enter a number:"))
epsilon = 0.00001
num_guesses = 0
low = 0.0
high = x
ans = (high + low)/2.0
while high - low >= 2 * epsilon:
    print("low =",low,"high =", high)
    num_guesses += 1
    if ans ** 2 < x:
        low = ans
    else:
        high = ans
    ans = (high + low)/2.0
print('number of guesses =', num_guesses)
print(ans, 'is close to square root of', x)
```

Programming | Homework Examples - Unit 5



Part 4 - Recursion Basics

5-4-1 Computing Fibonacci Numbers

You are probably familiar with the famous Fibonacci sequence of numbers, which begins like this:

```
1, 1, 2, 3, 5, 8, 13, 21...
```

We'll index from 0, so the 0th and 1st numbers are both 1. The 2nd Fibonacci number is found by summing the 0th and 1st: 1 + 1 = 2. The 3rd is found by summing the 1st and 2nd: 1 + 2 = 3. After this point, each Fibonacci number is found by summing the previous 2.

You are to write a recursive function to compute the nth Fibonacci number. This means that your function will call itself and will not include explicit loops.

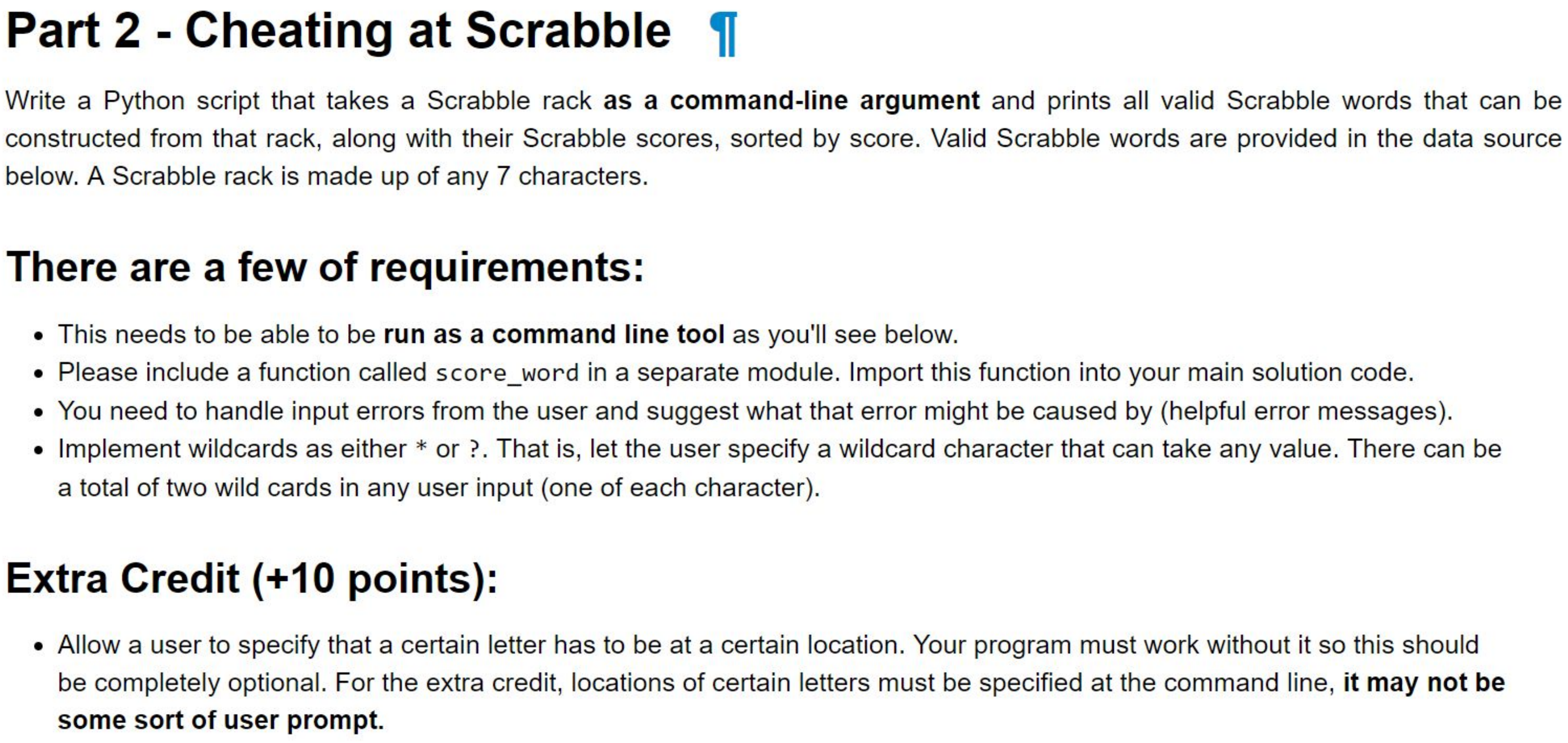
Hint: As with the Factorial example, your function should include a line that looks a lot like the mathematical definition of the nth Fibonacci number.

Another Hint: It's possible for a recursive function to call itself more than once.

Once you are done, think about whether your function is efficient. Compare it to a non-recursive implementation.

In [ ]:
```
def Fibonacci(n):
    pass
```

Programming | Homework Examples - Unit 6



Part 2 - Cheating at Scrabble ¶

Write a Python script that takes a Scrabble rack **as a command-line argument** and prints all valid Scrabble words that can be constructed from that rack, along with their Scrabble scores, sorted by score. Valid Scrabble words are provided in the data source below. A Scrabble rack is made up of any 7 characters.

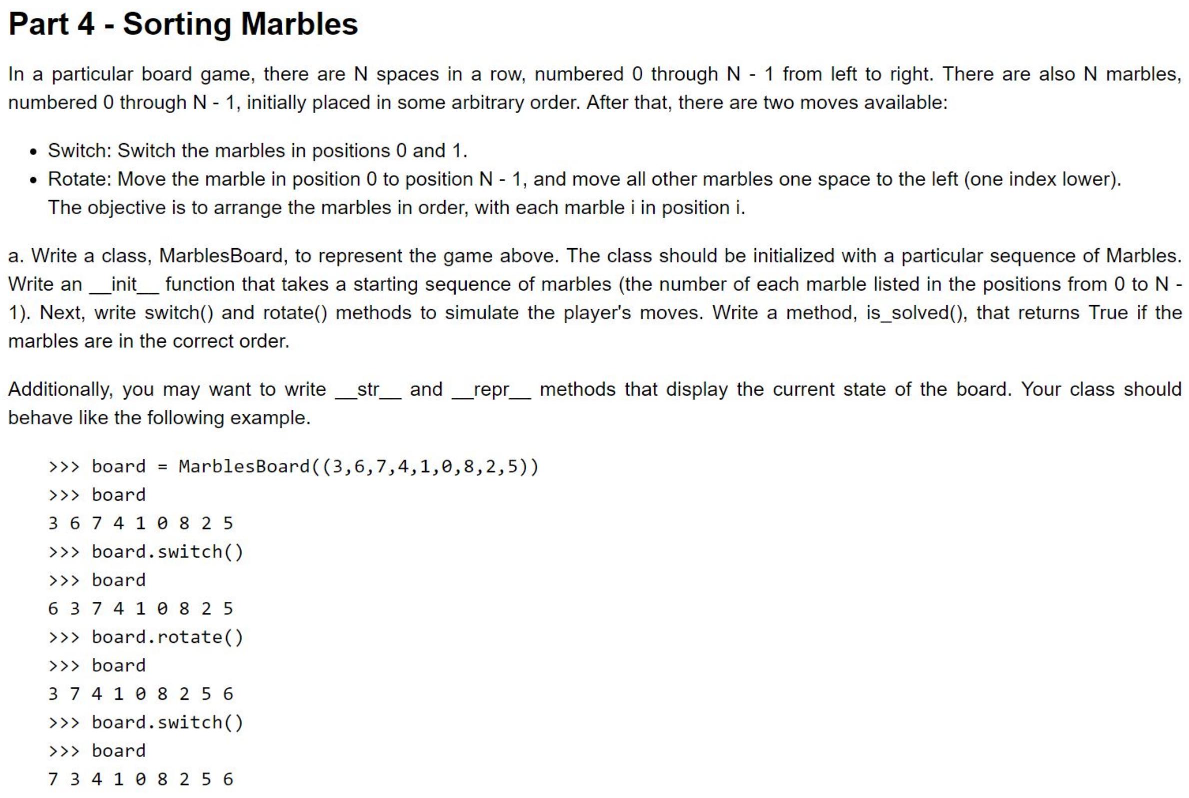
There are a few of requirements:

- This needs to be able to be **run as a command line tool** as you'll see below.
- Please include a function called `score_word` in a separate module. Import this function into your main solution code.
- You need to handle input errors from the user and suggest what that error might be caused by (helpful error messages).
- Implement wildcards as either `*` or `?`. That is, let the user specify a wildcard character that can take any value. There can be a total of two wild cards in any user input (one of each character).

Extra Credit (+10 points):

- Allow a user to specify that a certain letter has to be at a certain location. Your program must work without it so this should be completely optional. For the extra credit, locations of certain letters must be specified at the command line, **it may not be some sort of user prompt.**

Programming | Homework Examples - Unit 7



Part 4 - Sorting Marbles

In a particular board game, there are N spaces in a row, numbered 0 through N - 1 from left to right. There are also N marbles, numbered 0 through N - 1, initially placed in some arbitrary order. After that, there are two moves available:

- Switch: Switch the marbles in positions 0 and 1.
- Rotate: Move the marble in position 0 to position N - 1, and move all other marbles one space to the left (one index lower).
  The objective is to arrange the marbles in order, with each marble i in position i.

a. Write a class, MarblesBoard, to represent the game above. The class should be initialized with a particular sequence of Marbles. Write an __init__ function that takes a starting sequence of marbles (the number of each marble listed in the positions from 0 to N - 1). Next, write switch() and rotate() methods to simulate the player's moves. Write a method, is_solved(), that returns True if the marbles are in the correct order.

Additionally, you may want to write __str__ and __repr__ methods that display the current state of the board. Your class should behave like the following example.

```
>>> board = MarblesBoard((3,6,7,4,1,0,8,2,5))
>>> board
3 6 7 4 1 0 8 2 5
>>> board.switch()
>>> board
6 3 7 4 1 0 8 2 5
>>> board.rotate()
>>> board
3 7 4 1 0 8 2 5 6
>>> board.switch()
>>> board
7 3 4 1 0 8 2 5 6
```

Project 1 | Your Mission

Create a small, object-oriented program of your choosing:

Examples:

- An ATM
- A flower shop
- An adventure game
- Something relating to your everyday work

Project 1 | Code

Python 3 code, 300-500 lines (750 max)

All code should be well commented!

Must use Object Oriented design and classes

Demonstrate various flow controls and data types

Robust to common user errors and exceptions

Project 1 | Expectations

The user will interact with your program via Terminal/Shell

Three documents due before your class on 3/14 or 3/16:

1. Proposal (10%)
2. Code(s) (80%)
3. Reflective Summary (10%)

You will demo your progress in a breakout room (3/7 or 3/9)

You may only use Python libraries that come installed with Anaconda

Project 1 | Proposal

Describe your project concept

Pseudocode your major classes and functions

1. Briefly describe the purpose of each class
2. List expected functions belong to each class
3. List inputs and outputs for each function

Instructors will “approve” your draft proposal

Coding is **iterative**. Your final code may not match the proposal exactly

Project 1 | Reflection

Submit a 1-page reflection with your code

Instructors will read your reflection before grading your project

Tell us how to use your project!

Discuss challenges you faced and how you overcame them

The Curriculum | Last 6 Weeks - Data Analysis

Unit 9 | Working With Text and Binary Data

Unit 10 | NumPy

Unit 11 | Data Analysis With Pandas

Unit 12 | More Analysis With Pandas

Unit 13 | Testing

Data Analysis Group Project

The Curriculum | Data Analysis

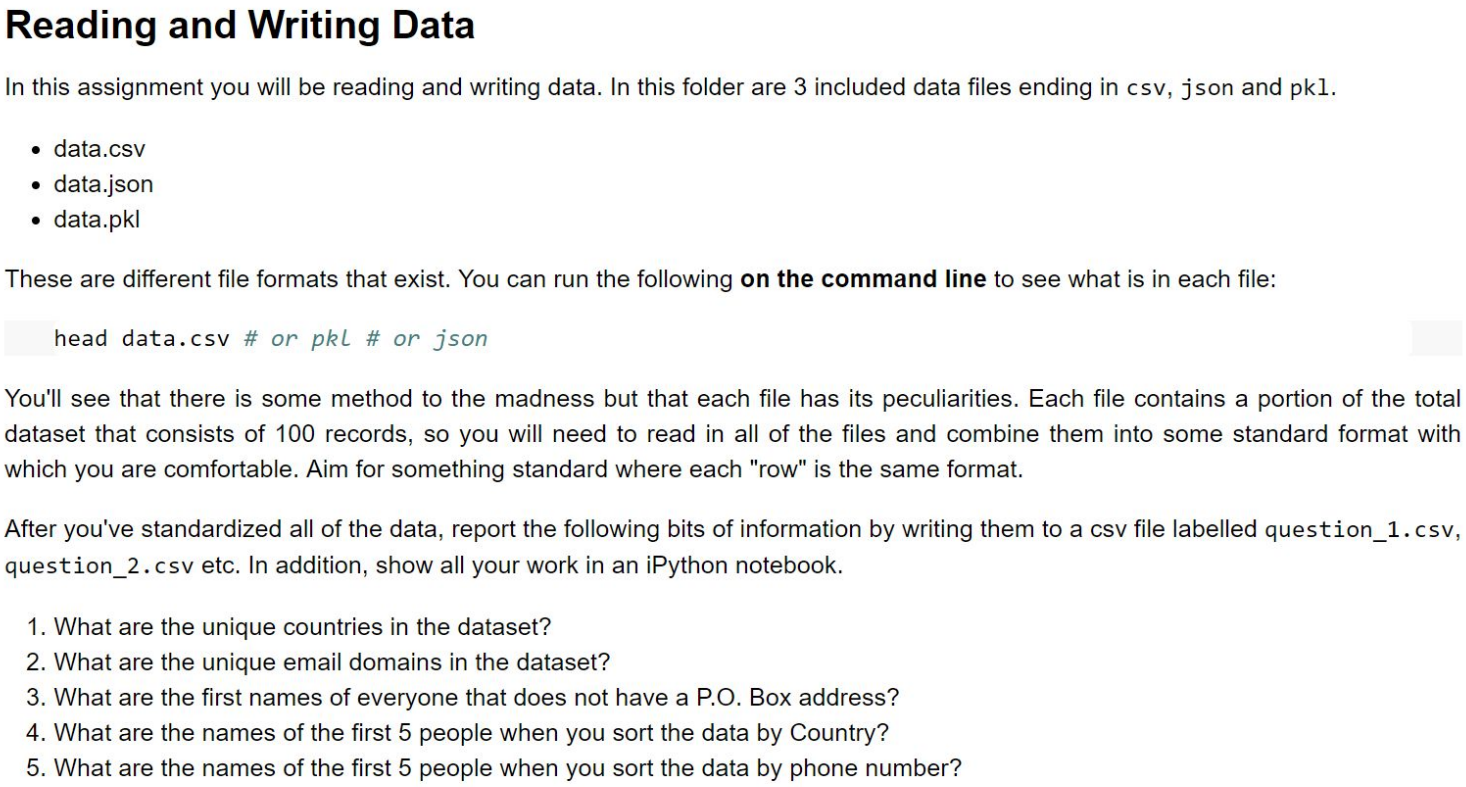
The last six weeks of our course are designed to teach students how to start thinking about analyzing data. This includes developing questions, sanity checking and exploring, transforming, and basics of plotting and visualization. Students then apply everything they have learned in their second project

We often discuss how the lessons of class extend beyond Python, and share stories from our personal experience in the classroom setting

We are planning to re-record a large portion of the asynchronous materials for this section

Homework in the second part of the course is more guided than the first.

Programming | Homework Examples - Unit 10



Reading and Writing Data

In this assignment you will be reading and writing data. In this folder are 3 included data files ending in `csv`, `json` and `pkl`.

- data.csv
- data.json
- data.pkl

These are different file formats that exist. You can run the following **on the command line** to see what is in each file:

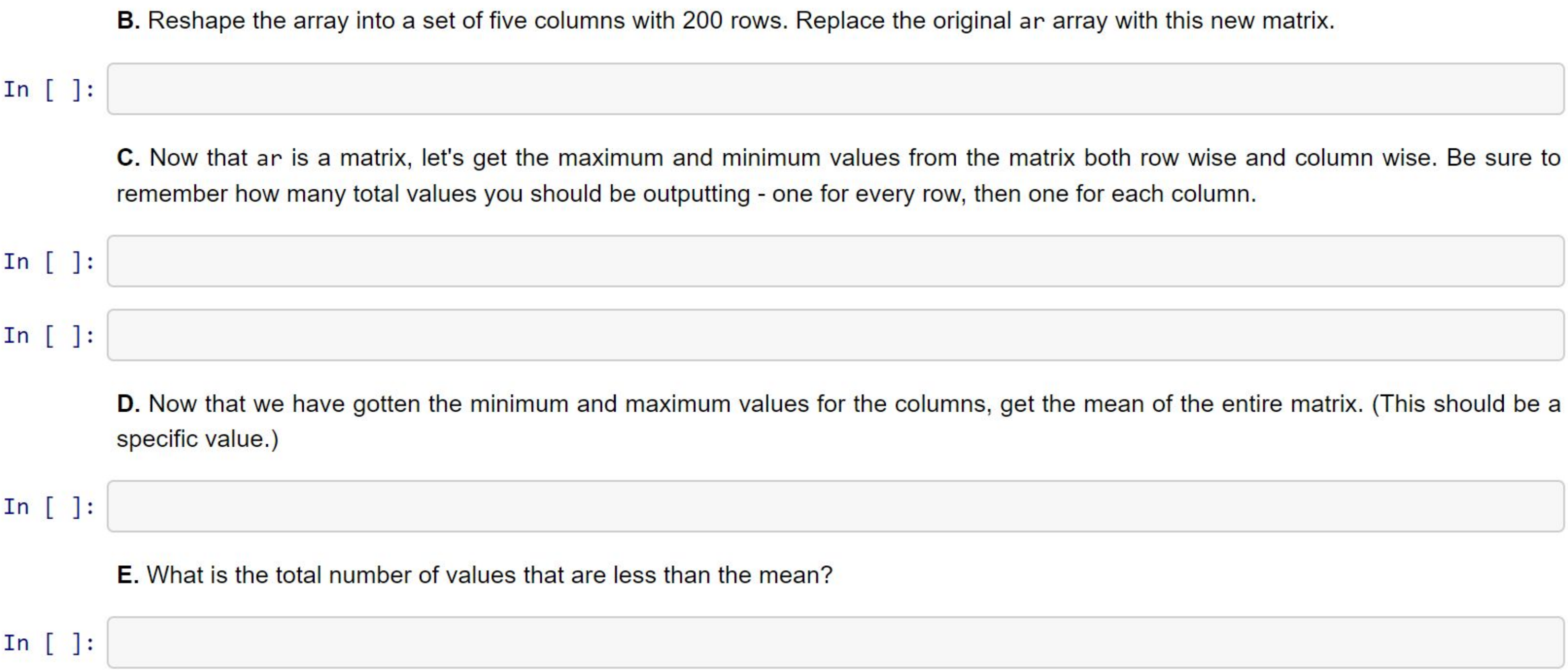
```
head data.csv # or pkl # or json
```

You'll see that there is some method to the madness but that each file has its peculiarities. Each file contains a portion of the total dataset that consists of 100 records, so you will need to read in all of the files and combine them into some standard format with which you are comfortable. Aim for something standard where each "row" is the same format.

After you've standardized all of the data, report the following bits of information by writing them to a csv file labelled `question_1.csv`, `question_2.csv` etc. In addition, show all your work in an iPython notebook.

1. What are the unique countries in the dataset?
2. What are the unique email domains in the dataset?
3. What are the first names of everyone that does not have a P.O. Box address?
4. What are the names of the first 5 people when you sort the data by Country?
5. What are the names of the first 5 people when you sort the data by phone number?

Programming | Homework Examples - Unit 11



B. Reshape the array into a set of five columns with 200 rows. Replace the original `ar` array with this new matrix.

In [ ]:

C. Now that `ar` is a matrix, let's get the maximum and minimum values from the matrix both row wise and column wise. Be sure to remember how many total values you should be outputting - one for every row, then one for each column.

In [ ]:

In [ ]:

D. Now that we have gotten the minimum and maximum values for the columns, get the mean of the entire matrix. (This should be a specific value.)

In [ ]:

E. What is the total number of values that are less than the mean?

In [ ]:

Programming | Homework Examples - Unit 12

To be shown in “Activity” demo

Project 2 | Proposal

With your group (2-3 people) come up with a 1 - 2 page proposal about the questions that you intend to ask of the data. This should include:

- Initial plots, figures or tables.
- References to column names and the analysis that they may provide.
- Additional datasets that you plan on including in your analysis like the weather data. This means links, columns that you'll join on, etc.
- What you plan to cover in the final report and how you plan on organizing it.

Project 2 | The Report

The report will be 8+ pages (including appropriately sized figures) and will be a report on what you found out from the data. This should focus on telling stories and explaining the narrative of the exploration and challenges associated with that. The report should not include any code - all code should be included in a sub-folder in either plain python files or in jupyter notebooks.

For the report, any graph, table, or figure should be annotated with why it is included. This is really to enforce just slapping graphs in your report that have no meaning.

Project 2 | Groups

Discuss Project Groups

One option is to plan based on your availability in this tracker:

https://docs.google.com/spreadsheets/d/1vgC9U0nKJS814oVi2UAqllBaRz6ItjUF7d1EghOuAkk/edit#gid=782266616

- Please add your project repro location

Project 2 | Forming Hypotheses

Which comes first? The question(s), or the data?

1. Data First - e.g., if your company collects data in course of business
2. Question First - e.g., if you have a strategic or research question to answer.

For this project, we suggest an iterative approach.

Project 2 | Forming Hypotheses

For this project, we suggest an iterative approach.

1. Look through potential data sources
2. Discuss questions that could be interesting
3. Confirm that the data source can answer the question
    a. If “yes”, develop your proposal
    b. If “no”, ask if there are similar questions you can answer with your data. OR find a new dataset.

Note: You may decide to combine multiple data sets for your project. For example, merging weather data onto traffic data to get additional insights. You can brainstorm this kind of combination as you look at data.

Course Overview | Agenda

Review Goals

Course Basics

The Student Experience

The Curriculum

Discussion

Activity

Student Showcase

Wrap Up

Discussion | Individualized Grading

We have become known for providing students with individualized feedback on their work, but as we grow that is no longer sustainable.

Discuss what we have done before, and NBGrader as an option to partially automate

Discussion | Flexible Exam Schedules

Our exams must each be completed within 24 hours, however, students are allowed to pick what day within a week they spend

We have worked with 2U to set up a method of accomplishing this

Course Overview | Agenda

Review Goals

Course Basics

The Student Experience

The Curriculum

Discussion

Activity

Student Showcase

Wrap Up

Activity | Let’s Code in Pandas!

Activity | Let’s Code in Pandas!

You are about to embark on an actual homework assignment… to analyze primary 2016 presidential campaign funding in California!

It is one of the easiest assignments in the course:

- We purposefully walk them through what to do step by step
- When we debrief the assignment, students discuss what they learned about the election
- Students actively apply what they have learned while completing Project 2
- We also have an extended demo where we show them how to plot data

Activity | Let’s Code in Pandas!

Please clone into the following repository:

https://github.com/MIDS-INFO-W18/W200CourseOverview.git

Find the pandas exercise here:

W200CourseOverview/exercises/PandasExr.ipynb

Download the dataset from here:

ftp://ftp.fec.gov/FEC/Presidential_Map/2016/P00000001/P00000001-CA.zip

Course Overview | Agenda

Review Goals

Course Basics

The Student Experience

The Curriculum

Discussion

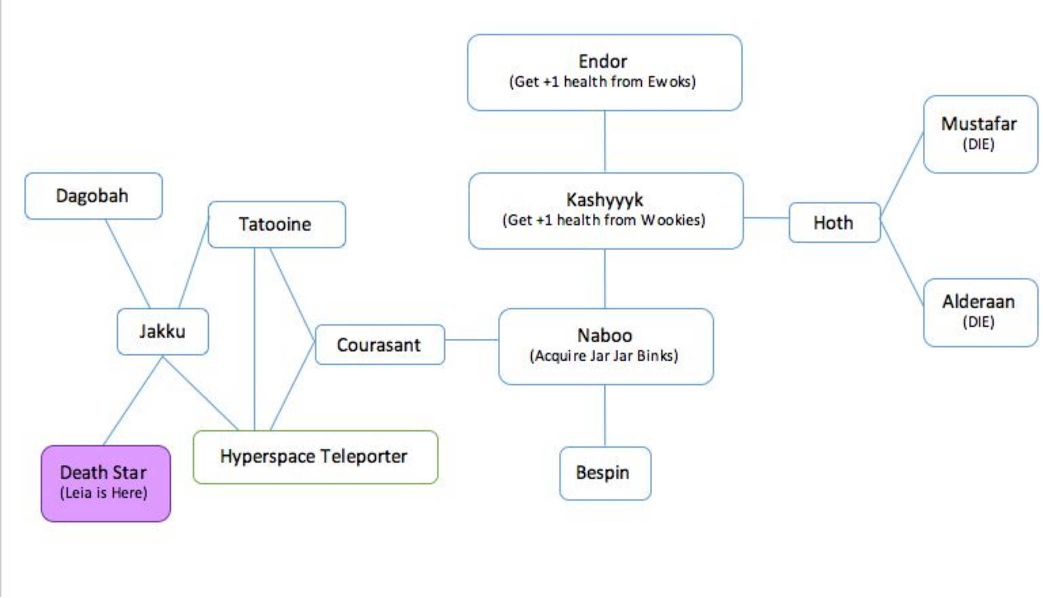
Activity

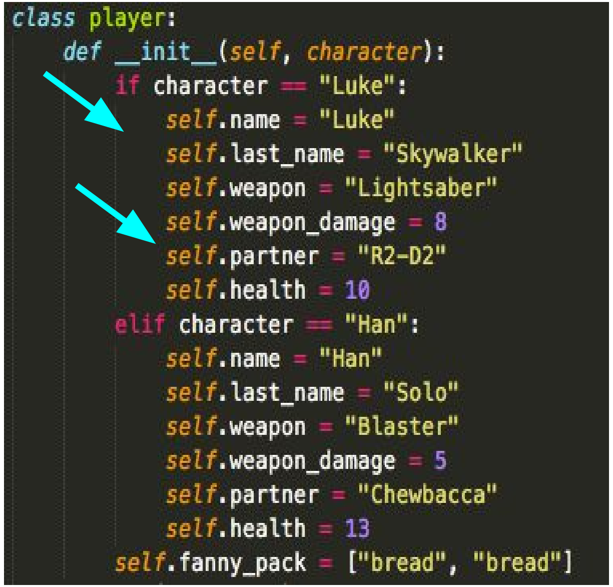
Student Showcase

Conclusions

Student Showcase | Project 1 - Object Oriented

Star Wars by Manish Singh





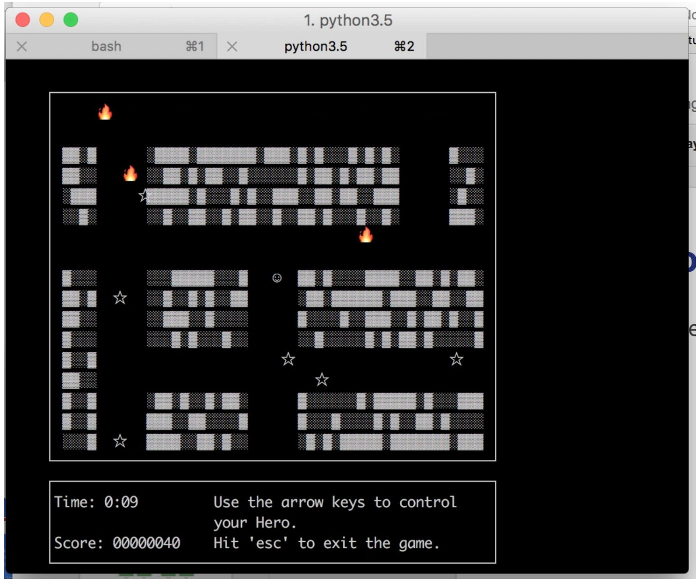
```
class player:
    def __init__(self, character):
        if character == "Luke":
            self.name = "Luke"
            self.last_name = "Skywalker"
            self.weapon = "Lightsaber"
            self.weapon_damage = 8
            self.partner = "R2-D2"
            self.health = 10
        elif character == "Han":
            self.name = "Han"
            self.last_name = "Solo"
            self.weapon = "Blaster"
            self.weapon_damage = 5
            self.partner = "Chewbacca"
            self.health = 13
        self.fanny_pack = ["bread", "bread"]
```

In the repository: W200CourseOverview/project_1/Star_Wars_Manish_Singh/

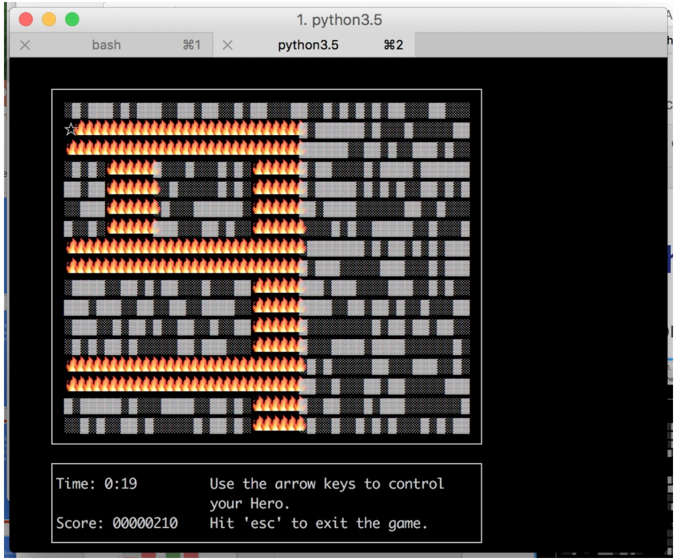
Student Showcase | Project 1 - Object Oriented

Maze Walker by Collin Reinking

Demo



```
Time: 0:09          Use the arrow keys to control
                    your Hero.
Score: 00000040     Hit 'esc' to exit the game.
```

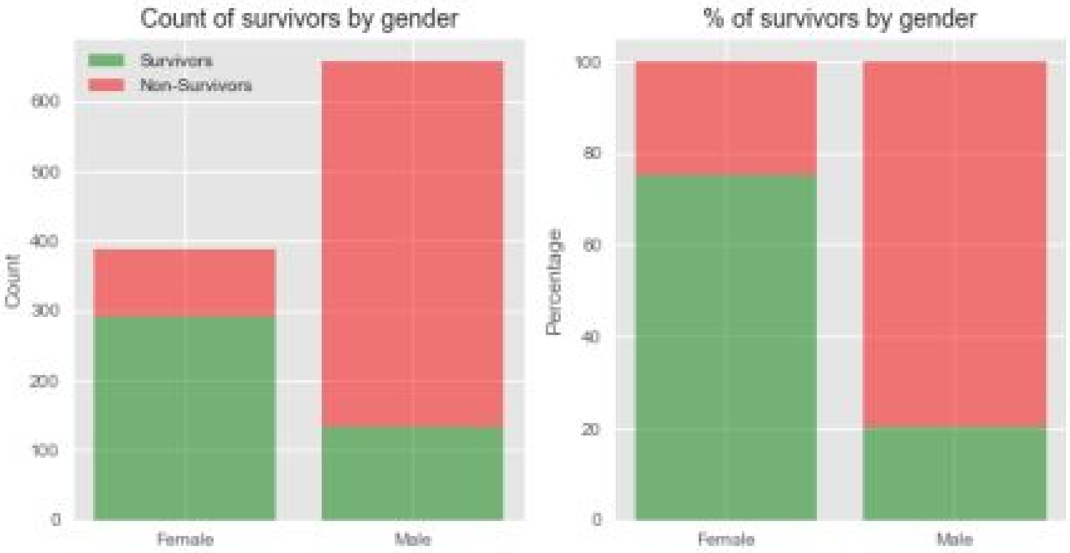


```
Time: 0:19          Use the arrow keys to control
                    your Hero.
Score: 00000210     Hit 'esc' to exit the game.
```

In the repository: W200CourseOverview/project_1/MazeWalker_CollinReinking/

Student Showcase | Project 2 - Pandas

Titanic survival by Danish Iqbal, Matthew Holmes, Kalvin Kao

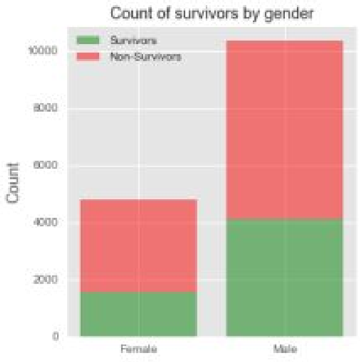


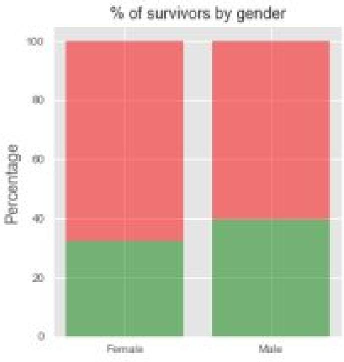
Women and children had higher survival

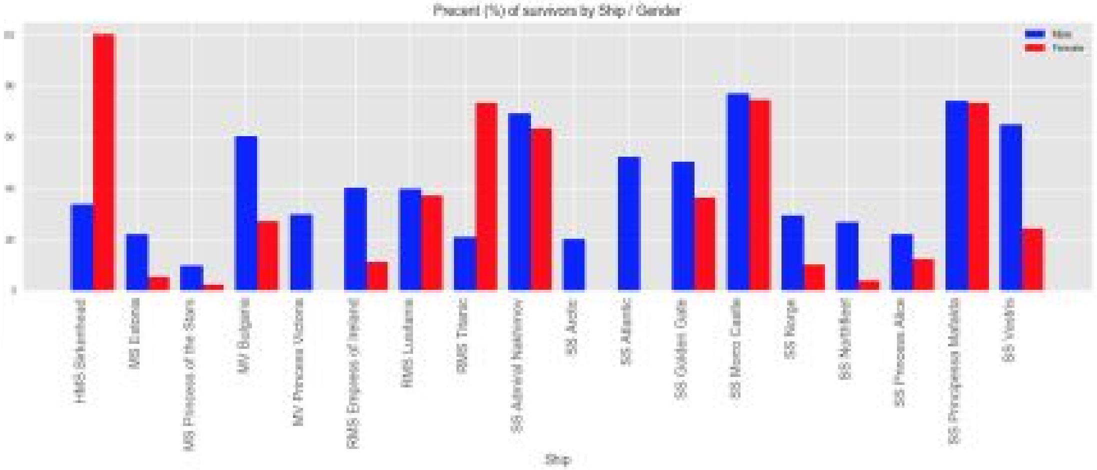
In the repository: W200CourseOverview/project_2/TitanicSurvival

Student Showcase | Project 2 - Pandas

Titanic survival by Danish Iqbal, Matthew Holmes, Kalvin Kao



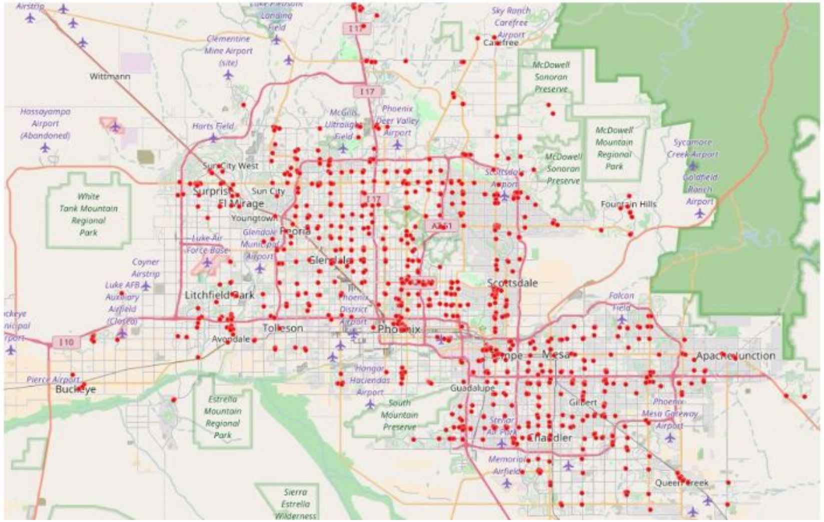




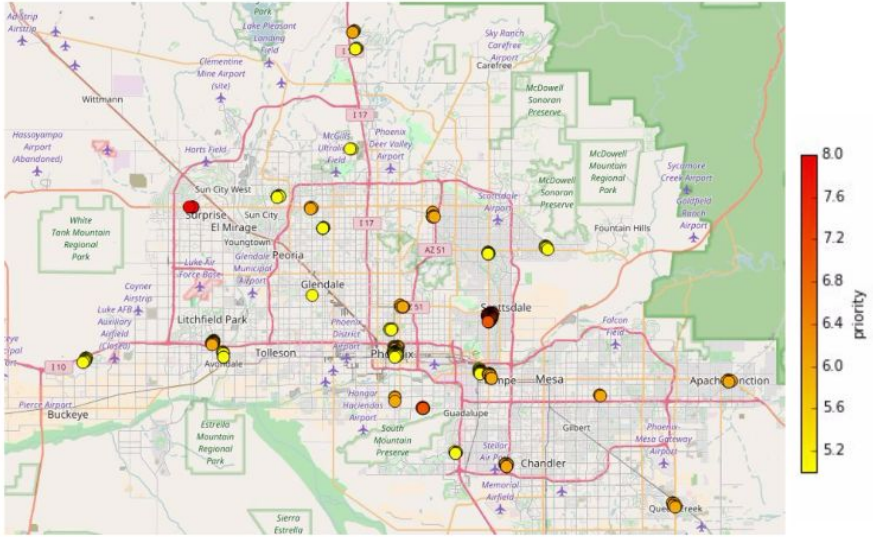
Analysis of 18 other shipwrecks suggest that “women and children first” less general than they expected

Student Showcase | Project 2 - Pandas

Pizza opportunities by Tom Seddon and Mike Amodeo



Existing shops in Phoenix

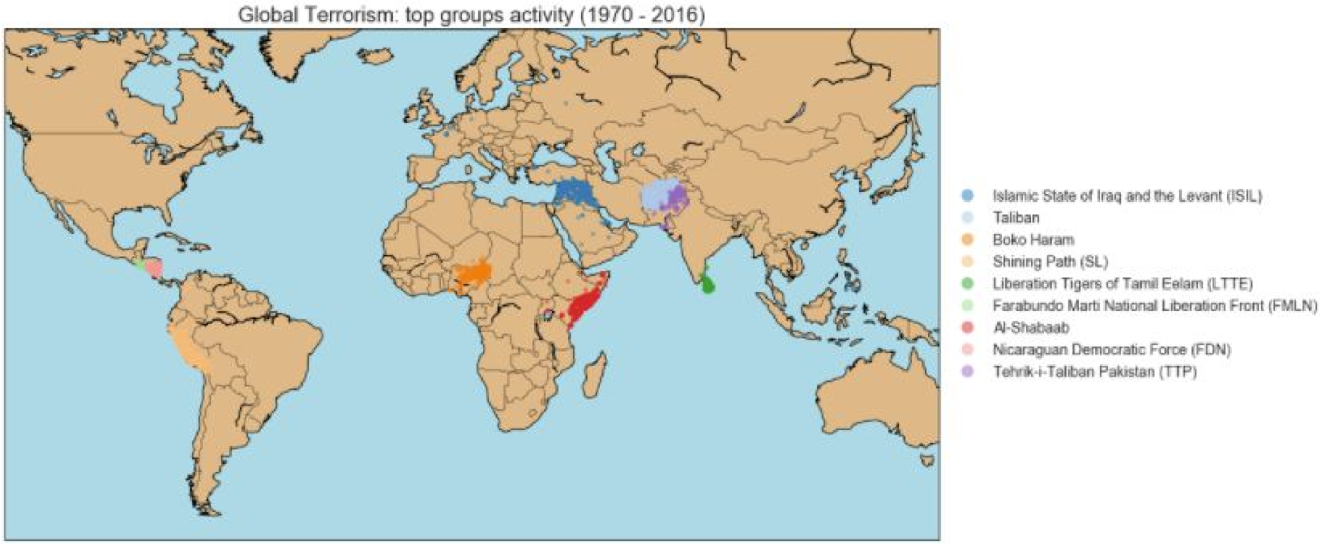


sites of opportunity

In the repository: W200CourseOverview/project_2/PizzaBuisiness

Student Showcase | Project 2 - Pandas

GLOBAL TERRORISM 1970-2016 by V. Nuretdinova, M. Al Zein

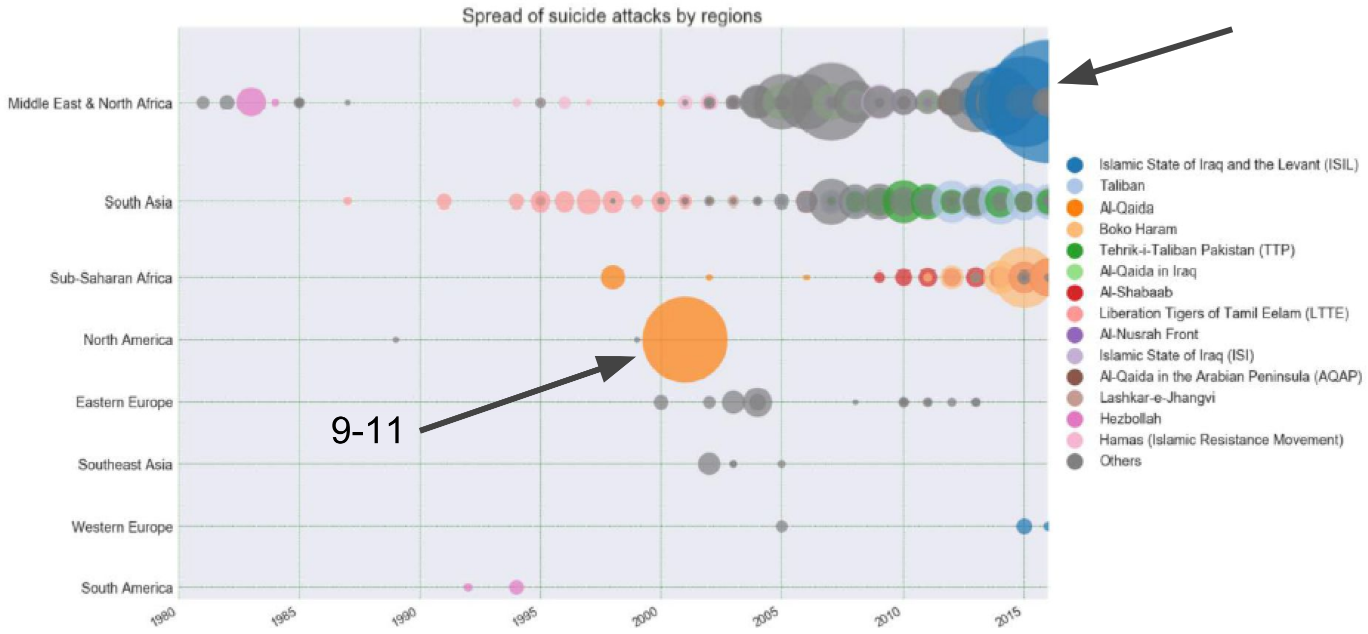

Global Terrorism: top groups activity (1970 - 2016)
● Islamic State of Iraq and the Levant (ISIL)
● Taliban
● Boko Haram
● Shining Path (SL)
● Liberation Tigers of Tamil Eelam (LTTE)
● Farabundo Marti National Liberation Front (FMLN)
● Al-Shabaab
● Nicaraguan Democratic Force (FDN)
● Tehrik-i-Taliban Pakistan (TTP)

Terrorist organizations tend to have a regional focus.

In the repository: W200CourseOverview/project_2/GlobalTerrorism

Student Showcase | Project 2 - Pandas

GLOBAL TERRORISM 1970-2016 by V. Nuretdinova, M. Al Zein



9-11

We can see historical pulses of terrorism by region.

Course Overview | Agenda

Review Goals

Course Basics

The Student Experience

The Curriculum

Discussion

Activity

Student Showcase

Wrap Up

Wrap Up | Thanks for Coming!

- We have a solid course to train students in python
- Upcoming improvements
  - Code updates
  - Development of Pandas section with the addition of more data visualization
  - Address the grading issue
    - Grading - TA
    - Automated grading and distribution (NBgrader)

Wrap Up | Thanks for Coming!

Any questions?

…

Who’s going next week?