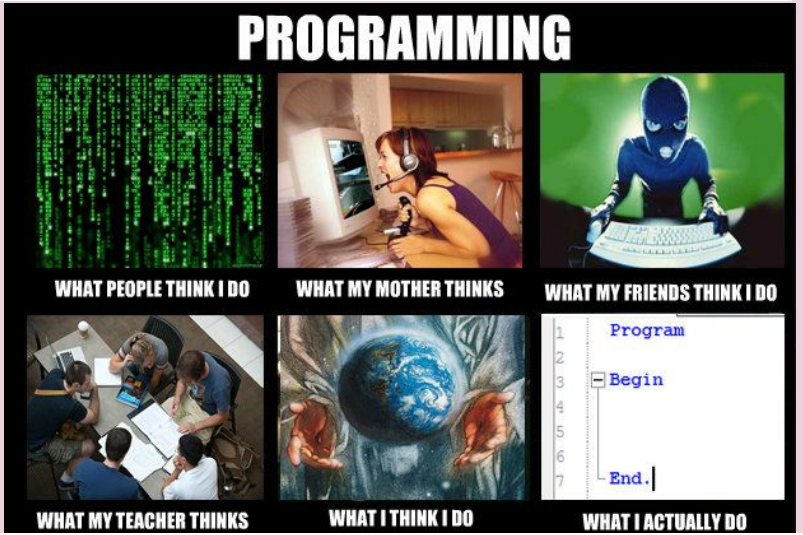
Day 9: Programming and Society, Final Project Work Sessions

July 20th 2023



Outline for today + tomorrow

Today

- review sheep problem from last night
- overview of past topics: what have we learned?
- how to pair code
- final project details
- group discussion about programming and society
  - where has code affected your life?
  - how is code affecting other people's lives?
  - what can we do about it?
- final project work session

office hours today! come code!

Tomorrow

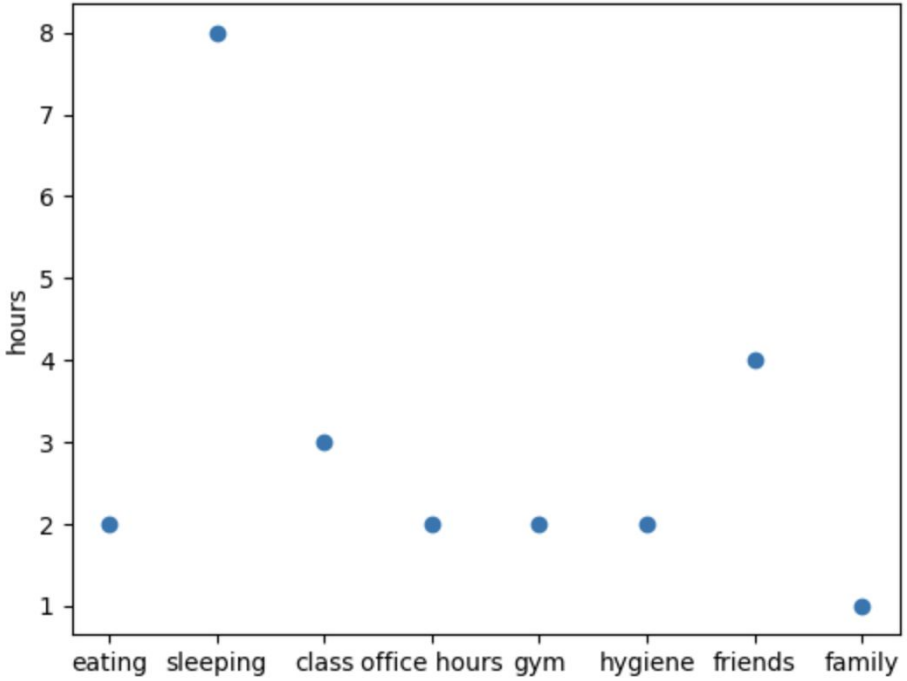
- presentations (class)
- presentations (instructors)
- open question / answer time: college? CS? other things?
- what next!
- course evaluation time

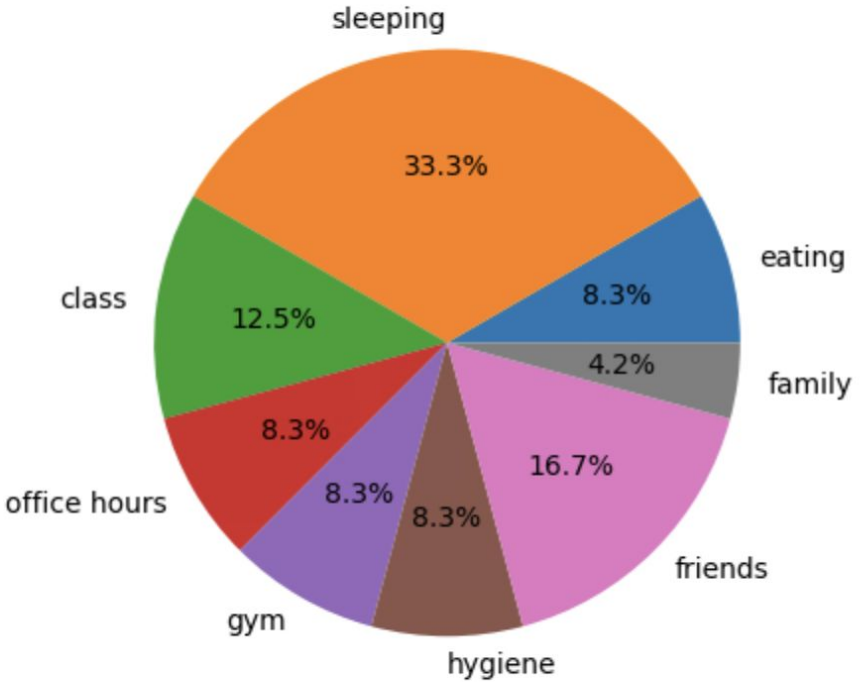
note! friday homework will be turning in your project code, group assessment, and final class reflection. we will only be hosting **short** office hours from 1-3:30pm

Past Topics Review: What all have we learned? What can you use to do your final project (and future coding?)

- how do computers store variables? how specific do you need to be to communicate with a computer?
- boolean logic: combining true/false statements and solving logic problems
- environment diagrams and how to debug code, commenting/documenting
- if/else statements: code performs different operations under different conditions
- looping: when to use for/while statements to make code repeat itself
- data types: ints, floats, strings, dictionaries, lists, vectors
- functions, scope, and writing general, repeatable code
- input(), print(), return
- common packages (numpy, pandas, matplotlib) and how to import / find information using online platforms
- plotting and manipulating data
- modulo math (everyone's favorite)

Effective Data Visualization





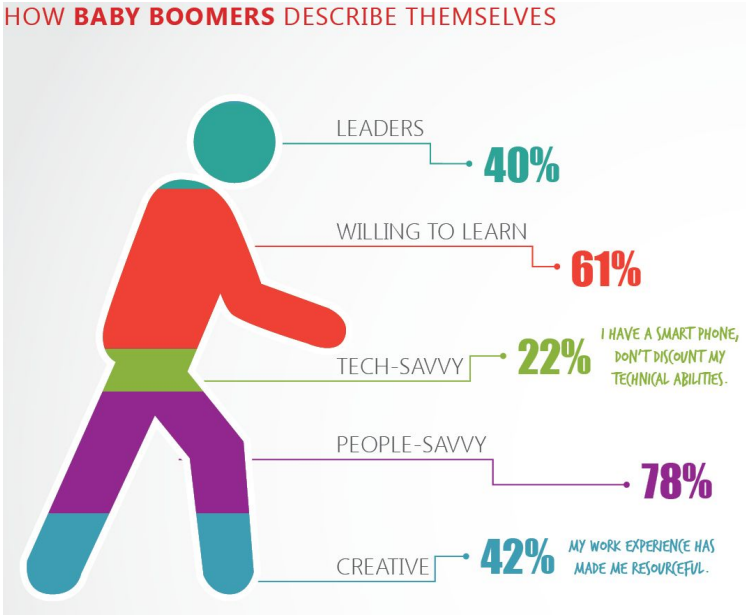
Effective Data Visualization

new fun feature of many data collection services: why is this an effective data visualization?

why can some data visualizations be misleading? interaction between plotting and human brain!

small tip for matplotlib: **figsize**

fig,ax = plt.subplots(figsize=(10,7))



How to Pair Code

Pair Programming

Pair programming is a way to program collaboratively with a partner. It's a great approach when you're learning how to program, plus it's used by many companies in the tech industry.

In pair programming, partners are working together *at the same time*. One partner is the "driver," who actually types the code. The other partner is the "navigator," who observes, asks questions, suggests solutions, and thinks about slightly longer-term strategies.

It's very important for partners to switch roles throughout an assignment, either every 10-20 minutes or alternating each problem.

A note on partnerships…

My partner and I didn't work well together...

We sometimes see feedback like:

> My partner didn't do any work, or didn't do their share of the work, or didn't communicate or meet with me, etc. What can I do?

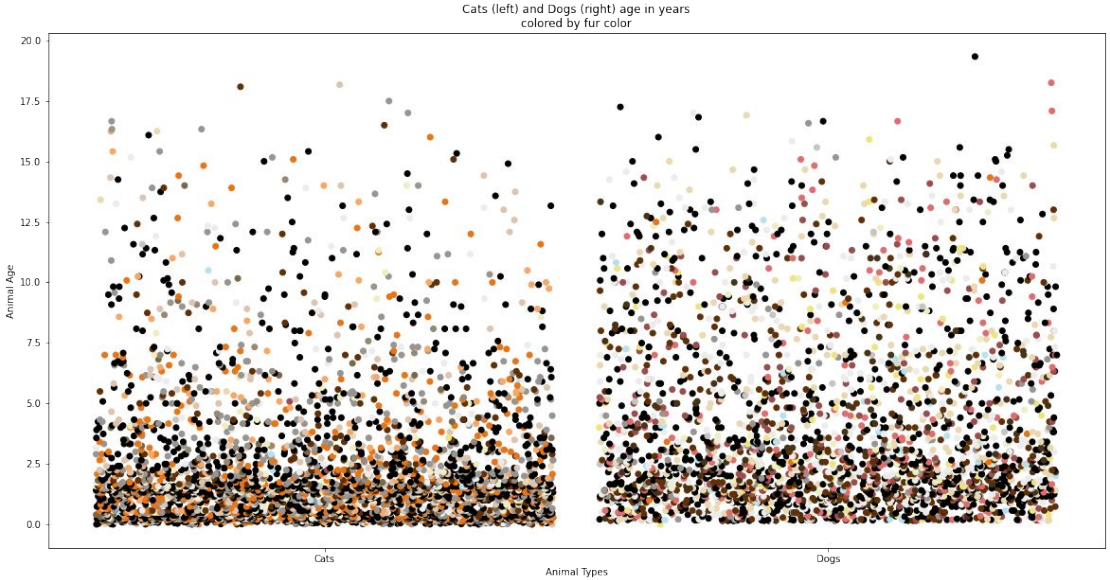
Or:

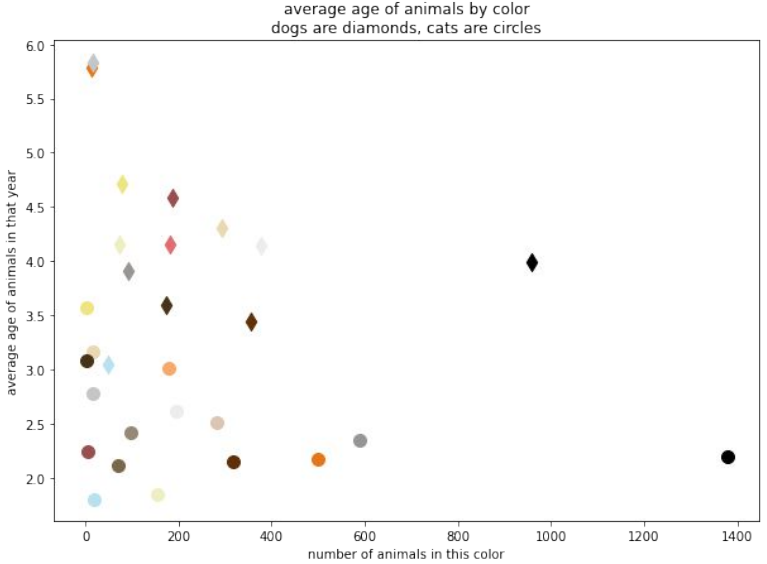
> My partner did too much! They hogged the keyboard, or they did the whole assignment without waiting for me, or they didn't communicate with me, etc. I feel that I didn't get a real chance to help in solving the assignment. What can I do?

Final Project Details

For the final project for this class, you will build on the homeworks you have completed this week on your chosen dataset. Your goal is to make **effective** visualizations of relationships you find in your data. For your project, you should make as many plots as you have people (so if you are in a 3 person group you should make three plots, and a 2 person group would make two plots). You can decide how you code your plots, but you should **comment appropriately –** a good rule of thumb is at least one comment per 5 lines!

Final Project Example





Final Project Details

For the final project for this class, you will build on the homeworks you have completed this week on your chosen dataset. Your goal is to make **effective** visualizations of relationships you find in your data. For your project, you should make as many plots as you have people (so if you are in a 3 person group you should make three plots, and a 2 person group would make two plots). You can decide how you code your plots, but you should **comment appropriately –** a good rule of thumb is at least one comment per 5 lines!

You can use the type of plot of your choosing – if there is a type you want to make we haven't learned, we can give you a crash course. Plots should contain proper labels for axes and titles, as well as any subsets of data. **At least one of your plots should contain a legend** – we have not learned this in class yet! Use your documentation searching skills to figure out how to add this. **Additionally, your plots should use at least one non-standard marker and non-standard color. At least one plot should present only a subset of the dataset.**

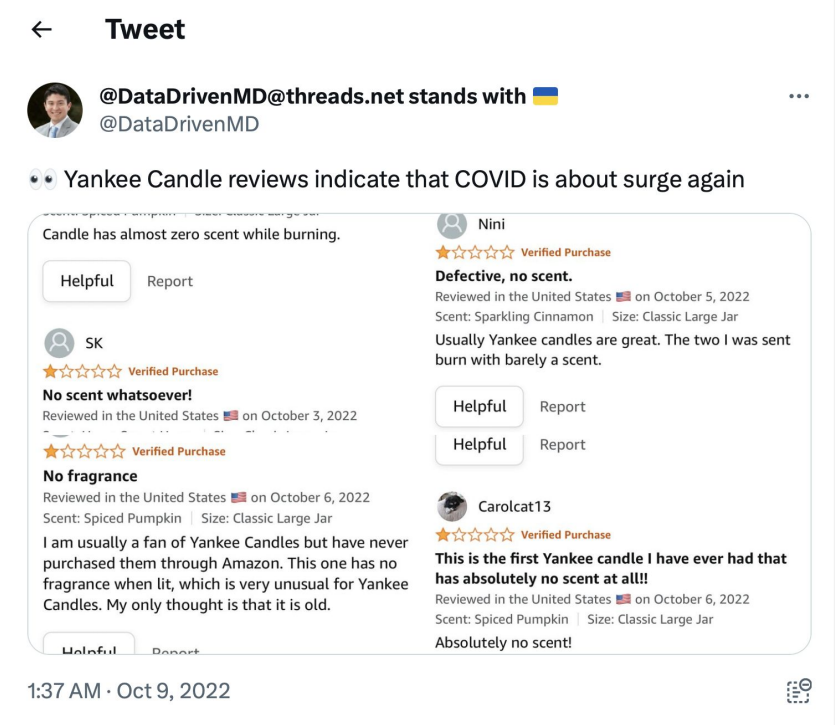
You will present your plots tomorrow in class. You should present what was in your dataset (even the info you don't use) and explain what you decided to explore. Then, you should walk us through your plots and how you chose to visualize the information. If you hit any struggles, you can explain what didn't work too!

You will also submit an (anonymous) review of what each group member contributed to the project, including but not limited to ideas for what data to work with, transforming data into form you want (new columns, subsets, etc), help debugging code, and working on how to visualize data.

Rubric

Category	Points
Correct number of plots	3
Plots visualize data effectively (labels are clear)	5
One or more plots has a legend	5
One or more plots uses non-standard markers	1
One or more plots uses non-standard colors	1
One or more plots uses a subset of the data	3
In class presentation describes dataset as a whole	2
In class presentation describes what you are plotting	3
Final code is clear: 1 or more comments approximately every 5 lines	3
Group work description is completed	1
Code history and group members indicate appropriate group contribution	3

Programming and Society



less serious examples of data and society

very serious examples of data affecting society

how might datasets perpetuate inequality? what types of code assumptions can be limiting for people?

Biases at many stages: analyzing but also collecting!

where should we put traffic cameras? where there are more people speeding! how do we know where people are speeding? look at data we have on arrests / traffic stops!

why can this logic introduce biases?

Chat GPT / Art AI

```python
import pandas as pd

dfrt = pd.read_csv('use_for_rt_analysis.csv')
sc = []  # start with an empty list
subjects = dfrt['subid'].unique()
tasks = dfrt['taskid'].unique()
dtypes = dfrt['dtypes'].unique()
for s in subjects:
  subdf = dfrt.loc[dfrt['subid'] == s].copy()
  for t in tasks:
    subtaskdf = subdf.loc[subdf['taskid'] == t].copy()
    for d in dtypes:
      # select data in that category
      # you can chain logical statements with & in loc
      sdata = subtaskdf.loc[subtaskdf['dtype'] == d]
      # I have 3 features I want to look at, calles s1 s2 s3
      for f in ['s1', 's2', 's3']:
        ttype1_rt_mean = sdata.loc[sdata['same_' + f] == False]['rt'].mean()
        ttype2_rt_mean = sdata.loc[sdata['same_' + f] == True]['rt'].mean()
        cost = ttype1_rt_mean - ttype2_rt_mean
        summary_dict = {'subid':s,'taskid':t,'dtype':d,'feat':f,'cost':cost}
        # add this dictionary to a list
        sc.append(sc)
# you can give pandas a list of dictionaries and it will make it a dataframe!
sc_df = pd.DataFrame(sc)
def rename_lambda_func(x):
  return 's' + str(x)
sc_df['subid'] = sc_df['subid'].apply(rename_lambda_func)
```

Interesting Cases:

- hallucinating court cases
- ignoring lines of code ->
- write an algorithm to decide who gets a life jacket on the titanic

Have you ever used chat gpt? What did you think about it?

How do you think chatgpt should be used in the classroom? Does it matter what subject?

Work Session

- before calling us over with questions, practice working together to solve bugs!
  - identify: what line is it on? what part of the line is broken?
  - is it one of the **common bugs** from our list?
  - describe in words to another person in your group what you want to do
  - remember, the computer is not smart :) are you giving explicit instructions?
- we are here to help you especially if you have questions about how to do something we didn't learn explicitly in class!! we want to help you visualize your data however you see best fit
  - if you are stuck on what you might want to plot, we can help with that too – but discuss what your homework answers were!
  - a good way to start is picking two columns and asking 'how are these related? do I expect sub groups in them to be related differently?'

reminder: tomorrow we will only have office hours from 1-3:30pm! today regular hours

Homework Problem 1: Response about one article

Choose an article from the following links (also posted as pdfs on our class webpage) and write a short (approximately 200 word) paragraph reflecting on what you learned and/or what the article made you think about. You can use the following prompts for inspiration if you are not sure what to write:

What is the article about? Is this new information or something you have encountered before? Did it change the way you think about any of your data or how you interact with code? Is there anything you would like to learn more about after reading this article?

money lending

bike shares in dc

gay marriage

gender binary

big data

predictive policing

articles talk both about actual coding as well as data generated!

Homework Problem 2: Final Project

For the final project for this class, you will build on the homeworks you have completed this week on your chosen dataset. Your goal is to make **effective** visualizations of relationships you find in your data. For your project, you should make as many plots as you have people (so if you are in a 3 person group you should make three plots, and a 2 person group would make two plots). You can decide how you code your plots, but you should **comment appropriately –** a good rule of thumb is at least one comment per 5 lines!

You can use the type of plot of your choosing – if there is a type you want to make we haven't learned, we can give you a crash course. Plots should contain proper labels for axes and titles, as well as any subsets of data. **At least one of your plots should contain a legend** – we have not learned this in class yet! Use your documentation searching skills to figure out how to add this. **Additionally, your plots should use at least one non-standard marker and non-standard color. At least one plot should present only a subset of the dataset.**

You will present your plots tomorrow in class. You should present what was in your dataset (even the info you don't use) and explain what you decided to explore. Then, you should walk us through your plots and how you chose to visualize the information. If you hit any struggles, you can explain what didn't work too!

You will also submit an (anonymous) review of what each group member contributed to the project, including but not limited to ideas for what data to work with, transforming data into form you want (new columns, subsets, etc), help debugging code, and working on how to visualize data.