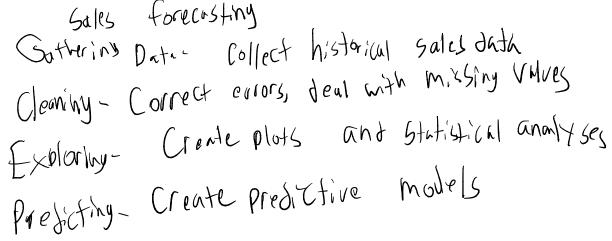
1. Today we discussed the data science process broadly as consisting of four steps: Gathering data, cleaning that data, exploring the data, and predicting unknowns. Imagine a data science problem and describe what these steps would look like for that problem in a few sentences each.



2. Read the program code below and answer the following questions.

```
import matplotlib.pyplot as plt imports for 4

xs = [70, 72, 72, 75, 80, 82, 90, 91, 89, 86, 82]

ys = [5, 5, 6, 8, 7, 8, 9, 12, 14, 11, 10, 8]

plt.scatter(xs, ys) (rent Scatter(xs, ys) (rent Scatter(xs, ys)) (abe( AKi))

plt.ylabel('temperature') (abe( AKi))

plt.ylabel('lemonade sold')
```

(a) Provide a brief explanation of what each line of code is doing. If you aren't sure, try running the program with that line commented out to see what happens

(c) What relationship between the two variables is suggested by the plot this code produces?

- 3. For each of the following data scenarios, state the most appropriate choice of visualization method and briefly justify your answer. Pick from the options of a scatter plot, a bar chart, a line chart, and a histogram.
  - (a) Visualizing a collection of numeric grades earned in one specific course.

- (b) Plotting votes for different catering options for a party.

  Bot What to Viscolitic different catering options for a party.
- (c) Charting a water reservoir's temperature measurements taken each day over a year.

(d) Visualizing mechanical component age, size and whether each part has failed yet or not.

4. pyplot is a large library with many, many features we aren't able to cover in this class. An important skill is to learn how to read code documentation, so you can continue to teach yourself after this course and your student career have ended. For this question you will want to consult the documentation page on pyplot's bar() function, found here (https://matplotlib.org/stable/api/\_as\_gen/matplotlib.pyplot.bar.html). Starting from the code below, modify this code to show the color bars as their actual colors. E.g. the bar for red should actually be red, the blue bar blue, etc. Read the documentation, experiment, and figure out how to do this. The answer should not take a large amount of new code.

```
import matplotlib.pyplot as plt

categories = ["Red", "Orange", "Yellow", "Green", "Blue", "Indigo", "Violet"]

votes = [20, 12, 14, 25, 29, 4, 9]

plt.bar(categories, votes)

plt.title("What is your favorite color?")

plt.xlabel('Colors')

plt.ylabel('Votes')

plt.show()
```

## 5. Consider the following problem:

You are given two **sorted** lists L1, L2 of arbitrary lengths. Design an algorithm that merges these two sorted lists into one larger **sorted** list. Your solution must have a worst-case complexity of  $\mathcal{O}(n)$ .

(a) Comparison-based sorting takes  $\mathcal{O}(n \log(n))$ , so naively combining the two lists and sorting the result is too slow. Consider how you can use the fact that the two lists are *already* sorted to merge them. Come up with an idea (try some examples) and describe the key idea in 1-2 sentences.

(b) Using pseudocode, implement your algorithm below. Write a function "merge" that takes two arguments L1 and L2, and returns the merged, sorted list.