

CS 134:

Plotting with matplotlib

Announcements & Logistics

- **Lab 5** is due Friday at noon
 - No TA hours on Thursday evening
 - Come see/email an instructor if necessary
 - See calendar for office hours this week
- **Midterm:** Thu Mar 17th
 - Attend one slot: **6 - 7:30pm** or **8 - 9:30pm** in **Wachenheim B I I**
 - **Wachenheim 002** at 6pm for reduced distractions/extra time
- **Practice midterm** on Glow
- Please fill out the **CSI 34 TA feedback form** by Friday

Do You Have Any Questions?

Last Time

- Wrapped up dictionaries
- Investigated **sorting** with dictionaries
- Discussed a new unordered data structure: **sets**
- Reviewed all data structures so far and when to use each

Today's Plan

- Learn about plotting with matplotlib
- Gain more practice using dictionaries, sets, tuples, and file reading

An Overview of Python Data Structures (so far!)

Python Data Structures at a Glance

	Lists	Tuples	Dictionaries	Sets
Order	Yes	Yes	No	No
Mutability	Yes	No	Yes (keys are immutable)	Yes (items are immutable)
Iterable	Yes	Yes	Yes	Yes
Comprehensions	Yes	Yes (need to enclose in tuple)	Yes	Yes
Methods	<code>.append()</code> , <code>.extend()</code> , <code>.count()</code> , <code>.index()</code> , etc	<code>.count()</code> , <code>.index()</code> ,	<code>.get()</code> , <code>.pop()</code> , etc	<code>.add()</code> , <code>.remove()</code> , etc

Python Data Structures at a Glance

	Lists	Tuples	Dictionaries	Sets
Order	Yes	Yes	No	No
Mutability	Yes	No	Yes (keys are immutable)	Yes (items are immutable)
Iterable	Yes	Yes	Yes	Yes
Comprehensions	Yes	Yes (need to enclose in tuple)	Yes	Yes
	<code>.append(),</code> <code>.extend(),</code> <code>.insert(),</code>	<code>.count(),</code> <code>.index(),</code>	<code>.get(),</code> <code>.pop(),</code>	<code>.add(),</code> <code>.remove(),</code> etc
Which to use when?				

Does Order Matter?

- Examples where **order** in data is important:

- Ranked ballots
- Queues
- Words in a sentence
- Tables/Matrices



- Tuples or lists?

- Do we need to **add/remove items dynamically**?
 - If yes, use **lists** (they are mutable!)
- If data stays same (no changes), use **tuples** (more space efficient)
- Even though you can concatenate items to tuples, it is not efficient, as it requires “copying over all the data” and creating a new tuple



Unordered Collections



- When storing a collection of data with ***no implicit ordering***:
 - Use **dictionaries** or **sets**
 - Dictionaries are more appropriate when there is a **key, value pair**
 - Better performance in general as compared to ordered structures
- Suppose we want to store student data in this course and quickly look up info for a given unix ID. Which data structure should we use?
 - Info may contain student name, class year, section, etc
 - Can store a **dictionary of dictionaries** (just like lists of lists!)

```
hpDict = { 'hp23': {'name': 'Harry James Potter',  
                 'house': 'Gryffindor', 'patronus': 'Stag'},  
          'hg3': {'name': 'Hermione Jean Granger',  
                 'house': 'Gryffindor', 'patronus': 'Otter'},  
          'll4': {'name': 'Luna Lovegood',  
                 'house': 'Ravenclaw', 'patronus': 'Hare'}}
```

Moving on...

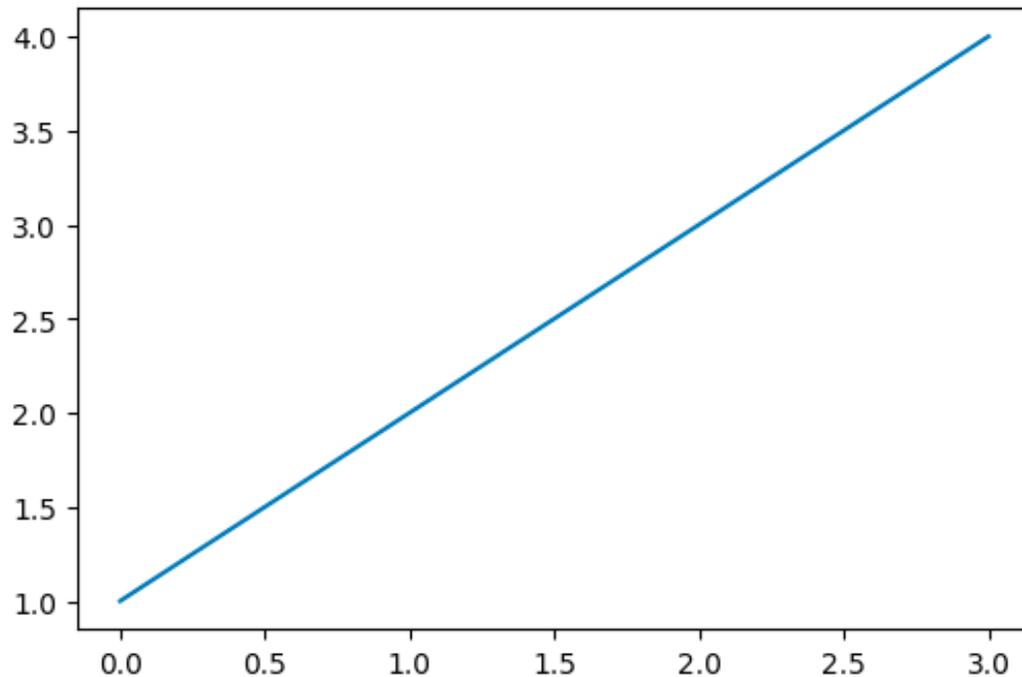
Plotting with `matplotlib`

- Suppose we want to a way to visualize our data (not just print it to the terminal)
- A plot is a graphical technique for representing a data set, usually as a graph showing the relationship between two or more variables
- We'll be using Python's `matplotlib` library to make plots/graphs
- The best way to learn how to plot different types of graphs is to read the documentation and see examples
- **Resources**
 - **matplotlib examples:** <http://matplotlib.org/examples>
 - **pyplot documentation:** http://matplotlib.org/api/pyplot_summary.html
 - **cool plots:** <https://matplotlib.org/gallery.html>

Plotting Basics: Plot function

```
import matplotlib.pyplot as plt  
plt.plot([1, 2, 3, 4])  
plt.show()
```

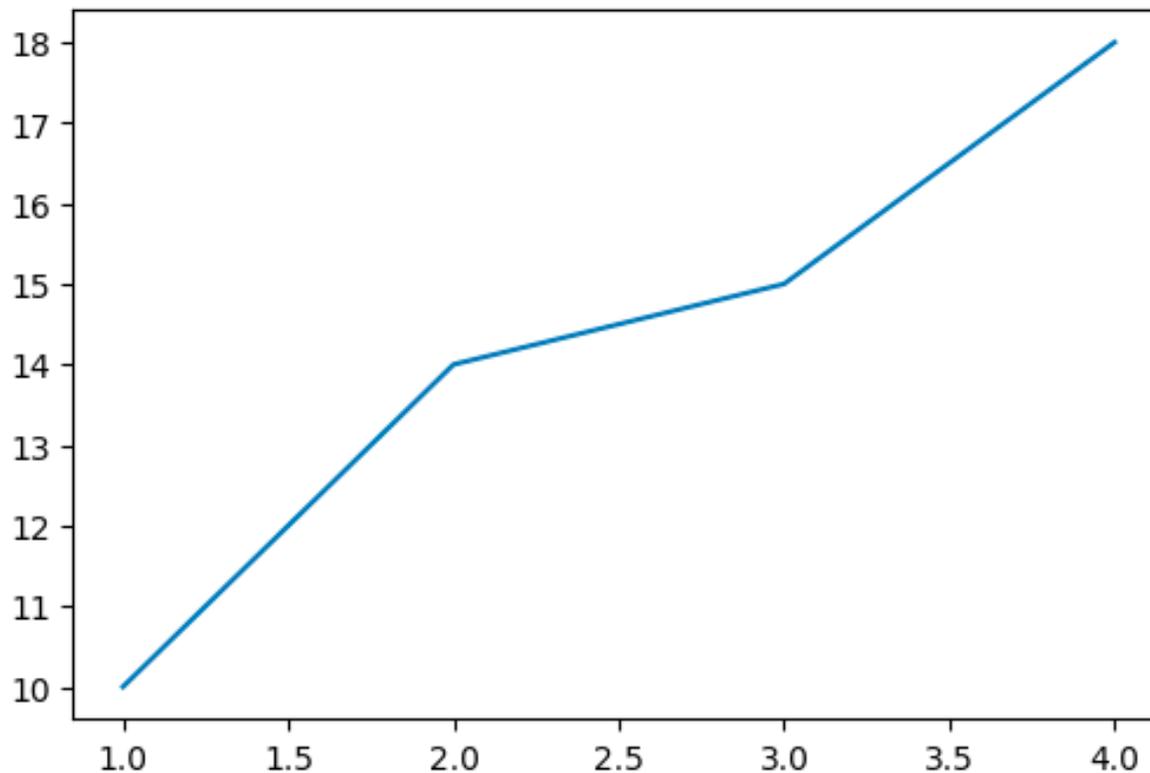
If only one list is provided, Python assumes it is as the points on the **y axis** (x values start at 0)



Plotting Basics: Plot function

```
import matplotlib.pyplot as plt  
plt.plot([1, 2, 3, 4], [10, 14, 15, 18])  
plt.show()
```

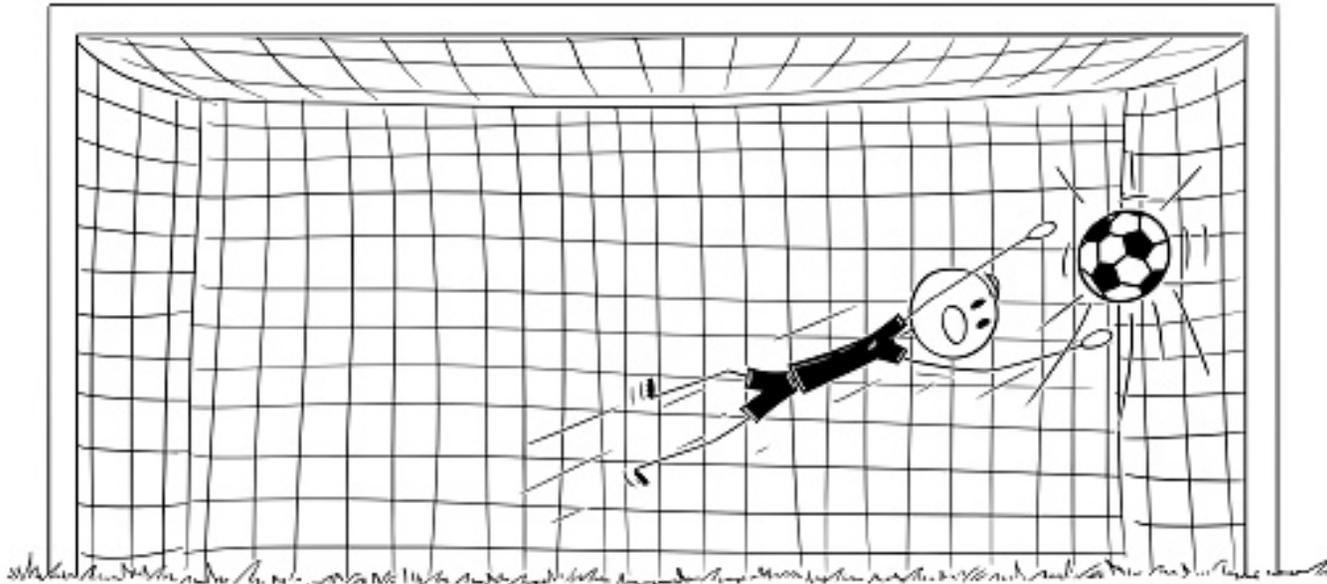
Equivalent to saying plot the points
(1, 10), (2, 14), (3, 15), (4, 18)



Exercise: Jupyter notebook

You are a talent scout for an English football (soccer) club. The club you work for has a good defense, but a weak offense. So, you've been tasked with identifying a star striker to help score more goals!

So you decide to identify candidates in a data-driven manner.



What we're aiming to produce

- We will plot bar charts showing the most frequent goal scorers in various years, and use them to determine who to try and recruit to our team

