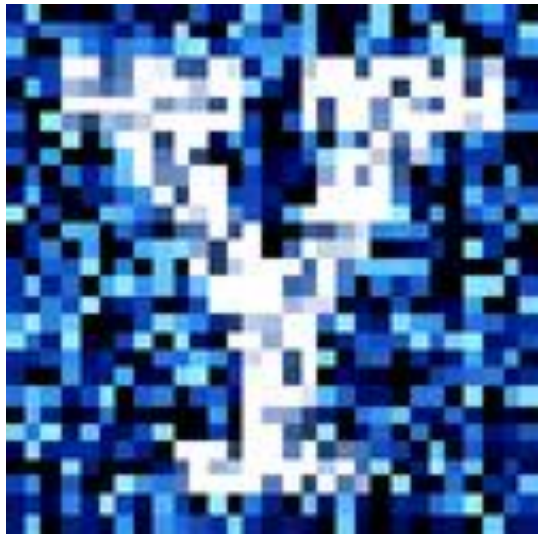
YData: Introduction to Data Science



Lecture 25: Odds and ends

Overview

Review of:

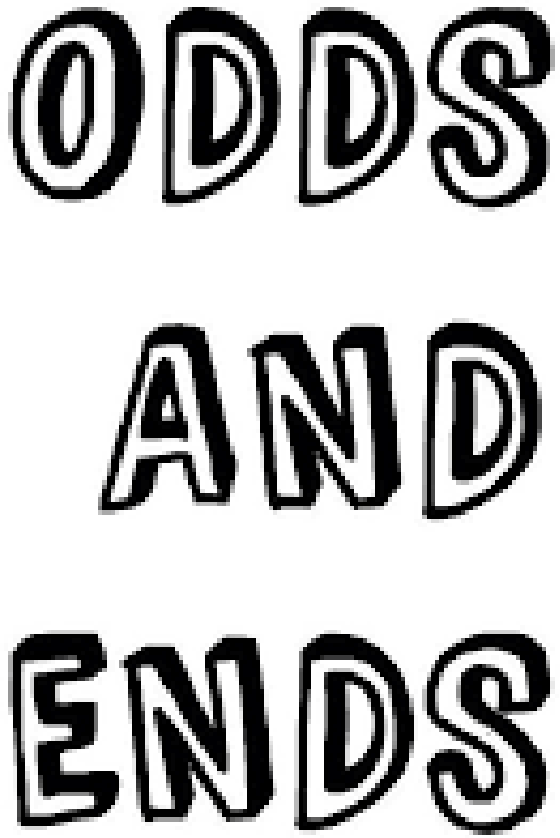
- Clustering
- Objected oriented programming

Miscellaneous topics

- Widgets for interactive analyses
- Creating a webpage on GitHub
- Installing Jupyter on your own computer

If there is time

- Ethics in Data Science



Project timeline

~~Sunday, November 17th~~

- ~~Projects are due on Gradescope at 11pm~~
- ~~Email a pdf of your project to your peer reviewers~~
  - ~~A list of whose paper you will review is posted on Canvas~~
  - ~~Fill out the draft reflection on Canvas~~

~~Sunday, November 24th~~

- ~~Jupyter notebook files with your reviews need to be sent to the authors~~
- ~~A template for doing your review has been posted~~

Sunday, December 8th

- Project is due on Gradescope
  - Add peer reviews to the Appendix of your project

~~Homework 9 has been posted~~

- ~~It is due December 1st~~

Final exam review session

- Monday December 9th at 3pm
- In this room

Review of Clustering

How many clusters?

Six Clusters

Two Clusters

Four Clusters

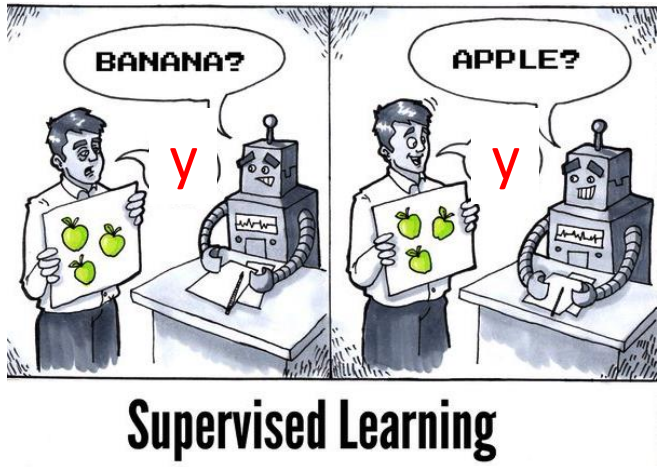
Supervised learning and unsupervised learning

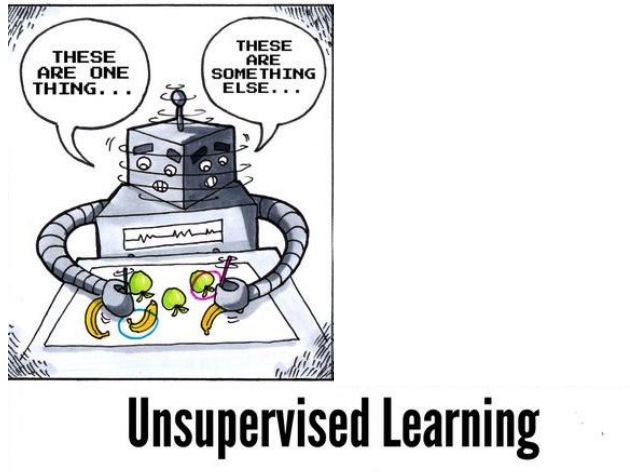
In **unsupervised learning**, we have features X, but **no** response variable y

- Unsupervised learning can be useful in order to find structure in the data and to visualize patterns,
- but there is no real ground truth response variable y

1. **Clustering** where we try to group similar data points together

2. **Dimensionality reduction** where we try to find a smaller set of features that captures most of the variability in the data
   - Principal component analysis (PCA)





Clustering

Clustering divides n data points x_i's into subgroups

- Data points in the same group are similar/homogeneous
- Data points in different groups are different from each other

p

n

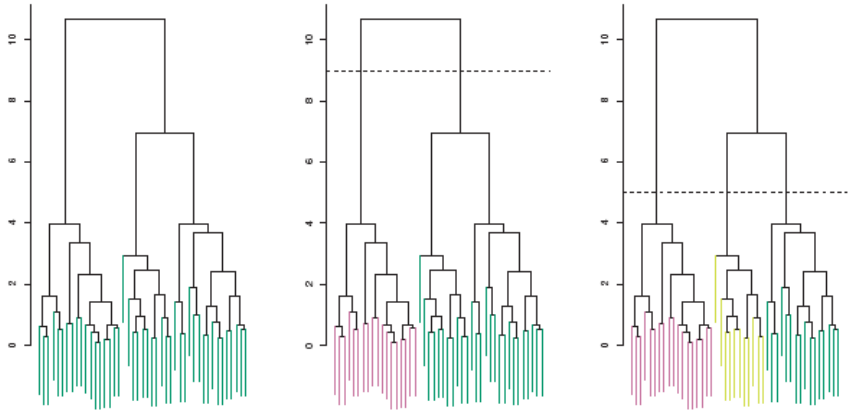
$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

Flat clustering: k-means

- Specify k clusters in advance is each point is assigned to one cluster

Hierarchical clustering

- Tree of nested clusters is created and we "cut" the tree to get a particular number of clusters



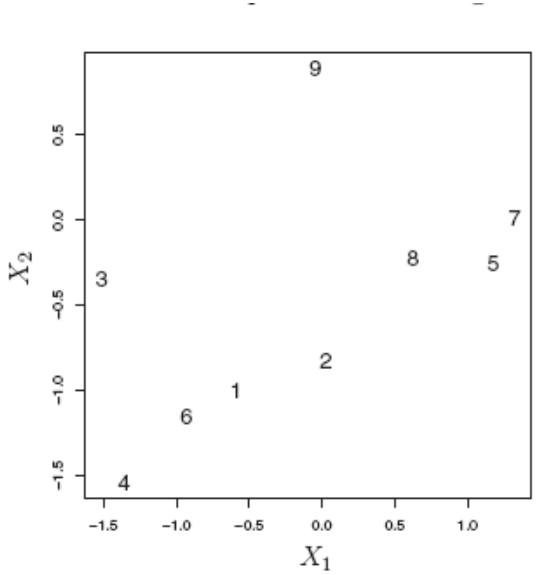
K-means clustering

1. Randomly assign points to clusters C_k
2. Calculate cluster centers as means of points in each cluster
3. Assign points to the closest cluster center
4. Recalculate cluster center as the mean of points in each cluster
5. Repeat steps 3 and 4 until convergence

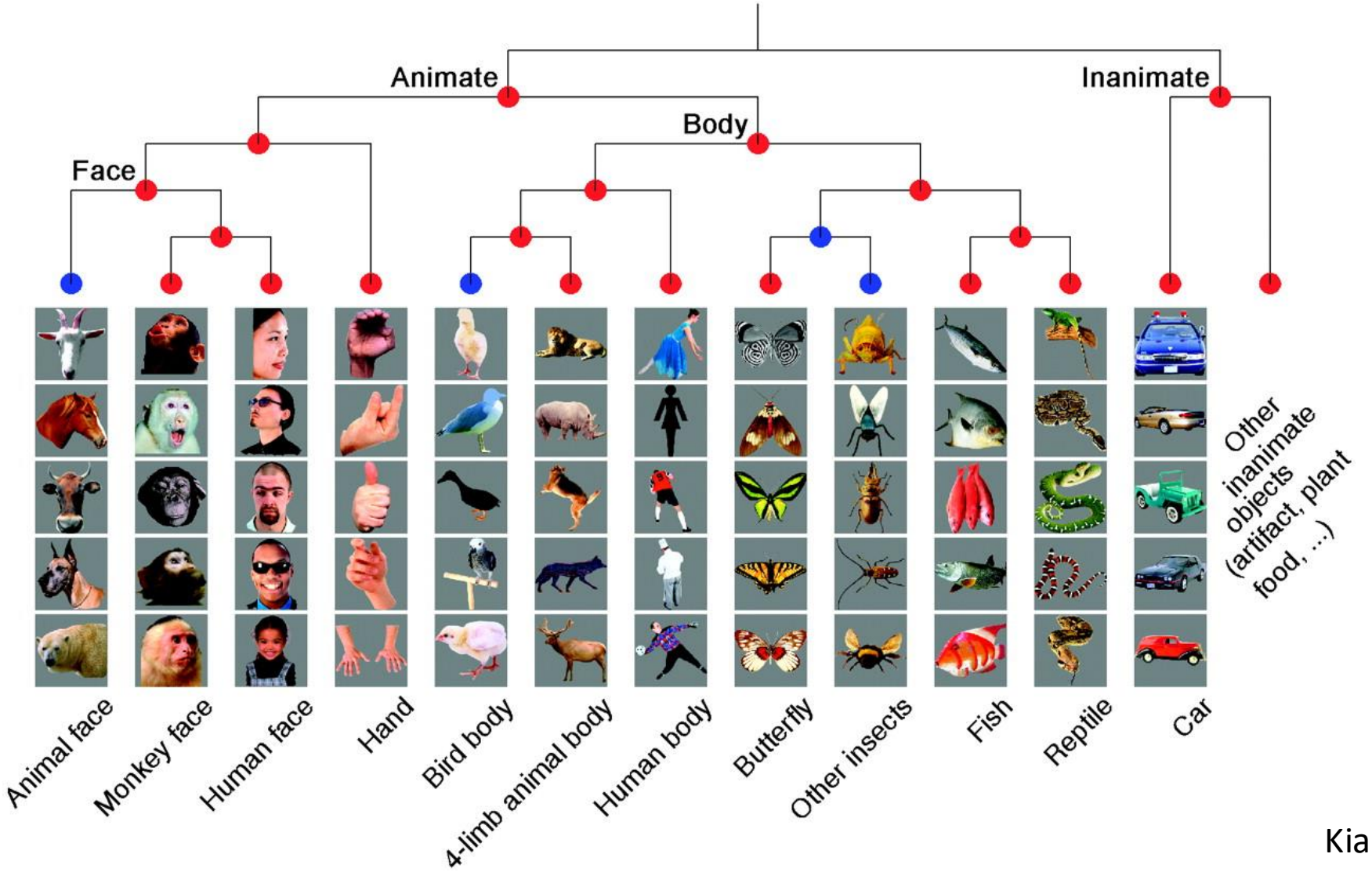
Hierarchical clustering

We can create a hierarchical clustering of the data using simple bottom-up agglomerative algorithm:

1. Choosing a (dis)similarity measure
   - E.g., The Euclidean distance
2. Initializing the clustering by treating each point as its own cluster
3. Successively merging the pair of clusters that are most similar
   - i.e., calculate the similarity between all pairs of clusters and merging the pair that is most similar
4. Stopping when all points have been merged into a single cluster



Hierarchical clustering example



Kiani et al, 2007

Review: Object oriented programming

Object-oriented programming

Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects", which can contain:

1. **data**: in the form of fields
    - often known as attributes or properties

2. **methods**: functions that operate on the data

Using object-oriented programming makes it easier to design larger software systems

- E.g., useful for writing packages, such as the ones we have used in this class

Object-oriented programming

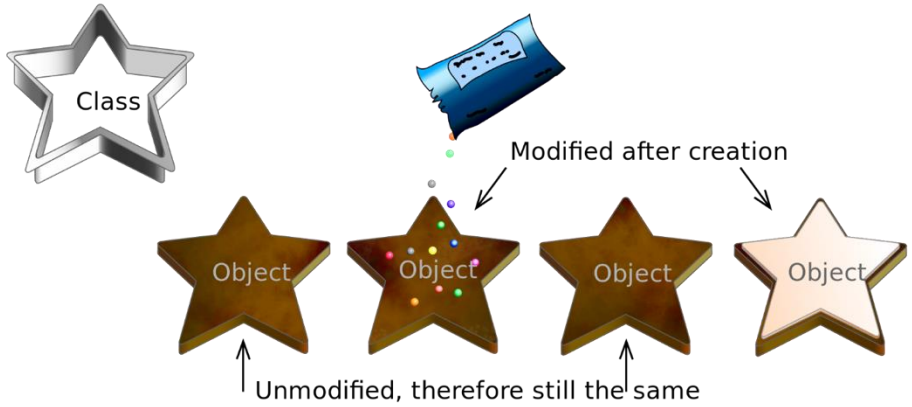
A **class** defines what the object is

- i.e., classes have code that lists the types of data stored and the methods

An **object** is a realization of classes (often called an instance of a class)

- Stores actual data that can be manipulated with methods

```
knn1 = KNeighborsClassifier(n_neighbors = 1)
knn5 = KNeighborsClassifier(n_neighbors = 5)
```



Useful because it allows us to have many object instances

- E.g., A KNN classifier trained on different data, different values of k, etc.

Object-oriented programming

A **constructor** defines what should happen when an object is created

- In Python the constructor is defined with the **__init__(self)** method

```
class StorageObj:

    # Constructor
    def __init__(self, a_num):

        self.n = a_num
        self.b = False
        self.s = "Hi"
```

```
# create an instance of an object
my_obj = StorageObj(3)

# get the value stored in n
my_obj.n
my_obj.s
```

Object-oriented programming

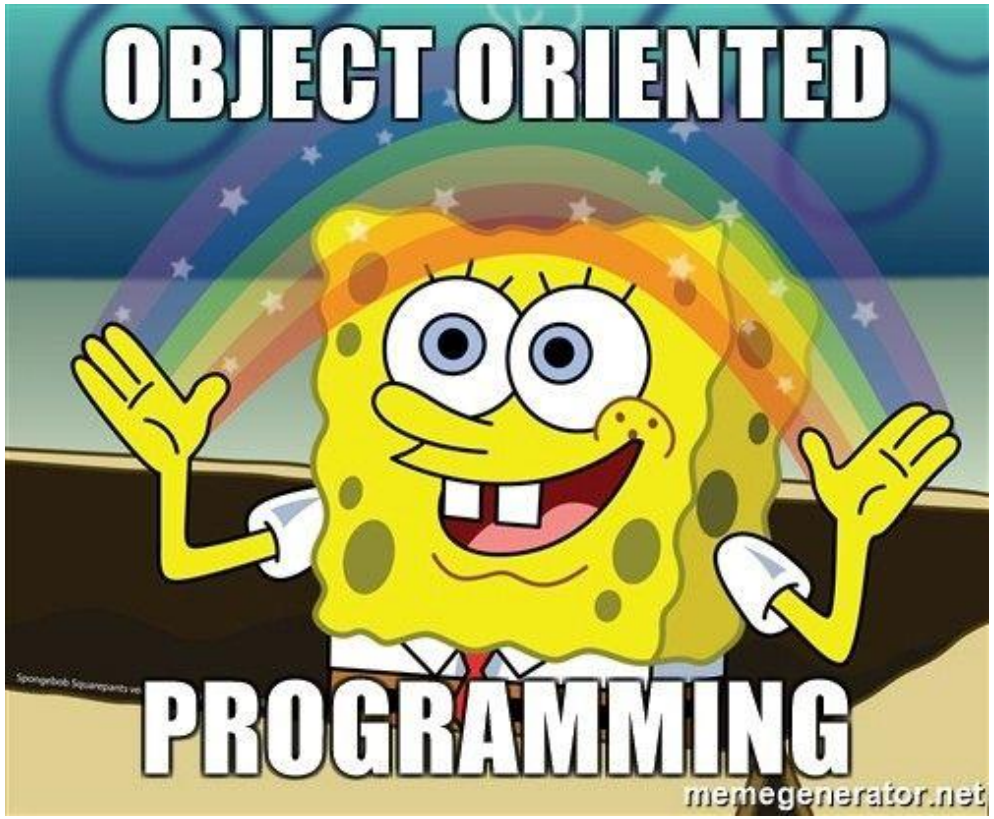
Methods are functions that are applied to the data stored in the object

For example, the .fit() method in an KNN object would save training and test data

```
class StorageObj:

    …   # constructor, etc. above

    def mult_num(self, val = 7):

        return self.n * val
```

```
# create an instance of an object
my_obj = StorageObj(3)

# call the .mult_num() method
my_obj.mult_num()

my_obj.n = 10
my_obj.mult_num(5)
```

Questions?



Let’s quickly review this in Jupyter!

Jupyter widgets

Jupyter widgets

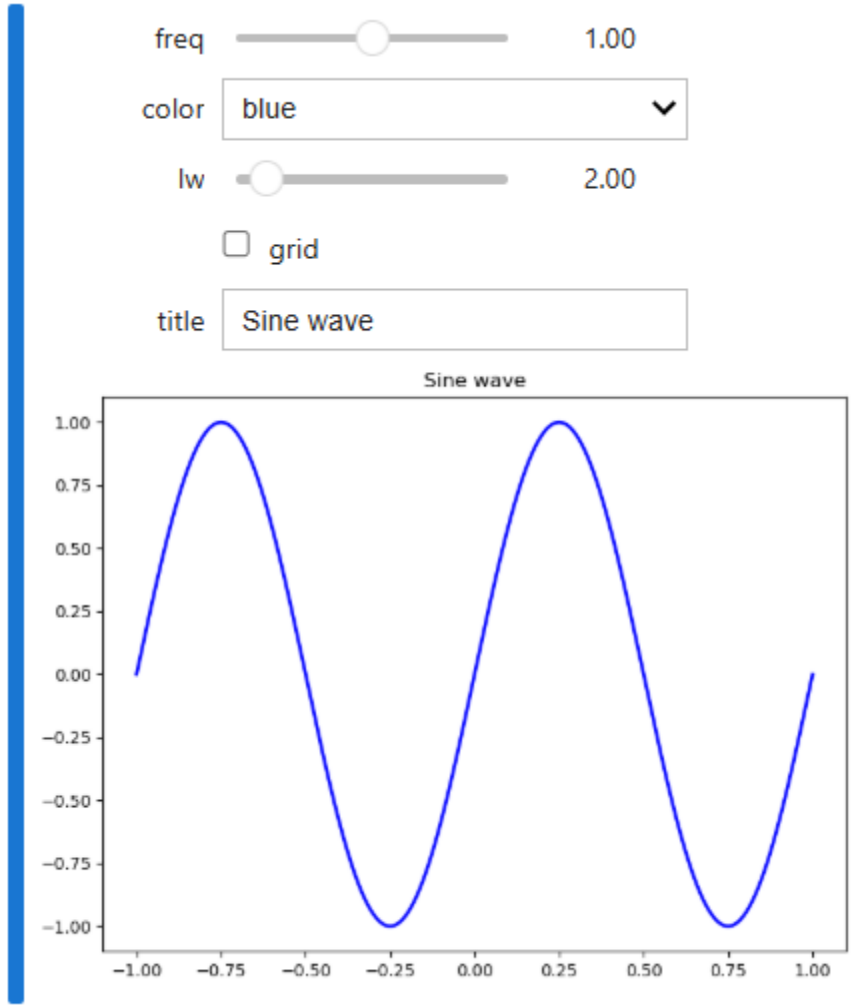
Jupter widgets allow you to add buttons, sliders, etc. to a Jupyter notebook so that you can create interactive visualizations

We can add widgets by importing:

```
import ipywidgets as widgets
```

We can then create a widget using:

```
bu = widgets.Button(description="Click")
```



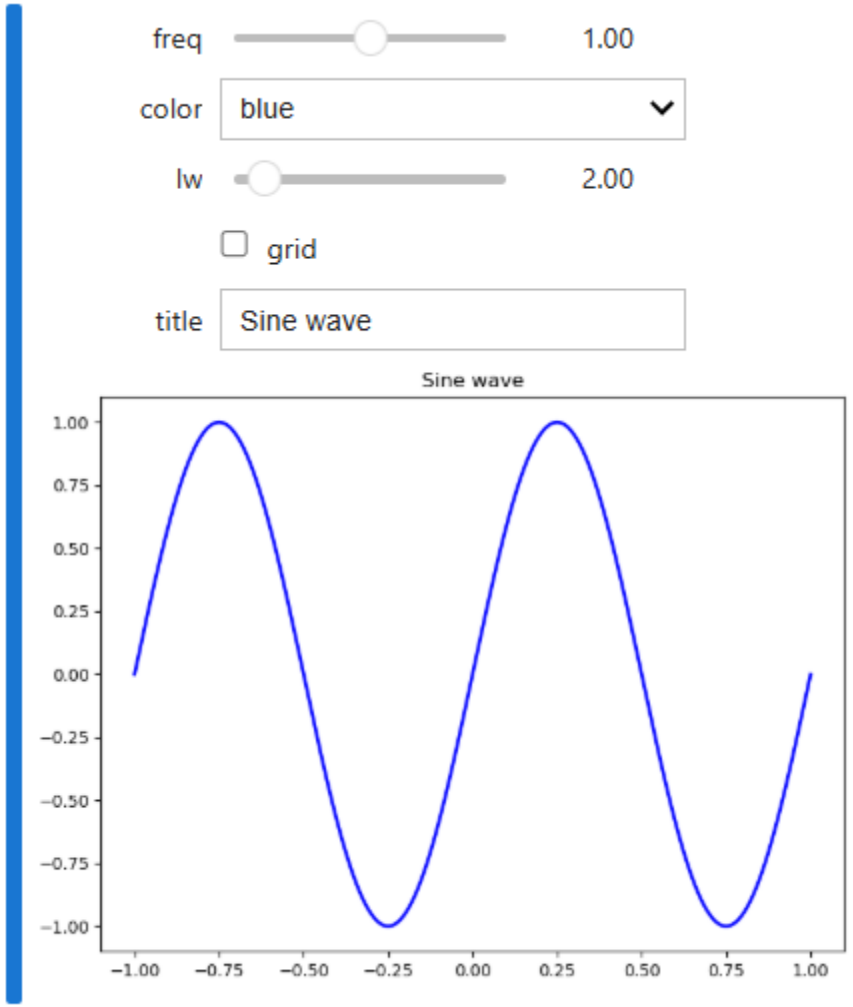
Jupyter widgets

There are several ways to connect widgets to figures

- One way is through widgets.interact()

```
def bandwidth_widget(bw = 1):
        sns.kdeplot(cars.horsepower,
                    lw=3,
                    bw_adjust=bw)

widgets.interact(bandwidth_widget,
                 bw = (.1, 3));
```

Adds a slider widget because this is a range of continuous values



Let's try it in Jupyter!

Hosting a webpage on GitHub

Hosting a webpage on GitHub

You've visiting many webpages in your life

- E.g., https://canvas.yale.edu/

Let's now discuss how we can create our own webpage and host it for free using GitHub.com

Webpages

Web pages are written in Hypertext Markup Language (HTML)

- Typically webpages end with the extension .html
  - Have you seen any files like this related to this class?

Webpage consists of text and tags

- Text is the content displayed on the webpage
- Tags allow one to insert links, images, and other meta-data

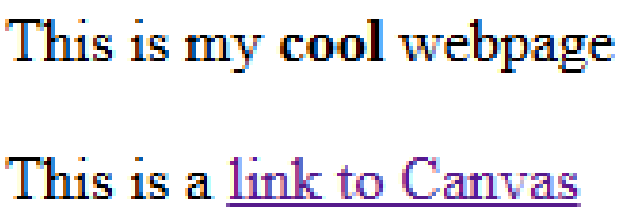
Tags have the form <> </>

- For example, we can make text bold using

The word bold is bold

Example of a basic web page

```
<html>

<title> My cool page </title>

<body>

    This is my <b>cool</b> webpage

    <br><br>

     Here is <a href="https://canvas.yale.edu/">Canvas </a>

</body>

</html>
```



Let’s create this simple webpage

Hosting webpages on GitHub pages

Webpages we've created

Your homework and your class project were saved as .html documents before you printed them to pdfs

GitHub.com is a service that allows people to share code

- Also allows one to host webpages for free!
- So if you save your project as a .html document, you can share it on the web!

Let's go through the steps now to host a webpage on GitHub

Installing Python/Juypyter on your own computer

Installing Juypyter on your own computer

Throughout this semester we have been using Jupyter on the YCRC cluster

Once the semester is over, you will lose access to this cluster

- So please be sure to backup any work you want to save!

If you want to keep using Jupyter, you will need to either:

- Use Google Colab
- Install Jupyter on your own computer

Environments and Anaconda

In order use packages on your own computer (e.g. pandas, matplotlib, etc) you first need to install

Environments allow one to switch between different versions of packages

- This is useful because packages are often updated

One of the most popular environment managers **conda**

- i.e., it allows you to install packages and switch between environments that have different packages installed

To download anaconda please go to:

https://www.anaconda.com/download/

Creating a conda environment

Once anaconda is installed, we can create an environment that has the data science packages we need.

One of the easiest ways to do this is to use a .yml file that has a list of the packages we would like to install

- E.g., ydata123.yml

Let’s try this now…