Announcements

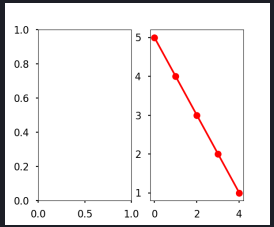
- Homework
  - Homework 10 posted!
  - Should already be able to do Problem 1 after today, and Problem 2 by Wednesday
- My quest to get caught up on grading continues, expect at least mostly up-to-date grade reports soon
- Information on semester-ending project coming end of the week
- Polling: `rembold-class.ddns.net`

Review Question

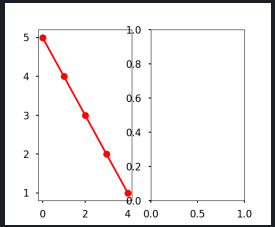
```
import matplotlib.pyplot as plt
f = plt.figure()
axa = f.add_subplot(122)
axb = f.add_subplot(121)
axa.plot([5,4,3,2,1], 'ro-')
plt.show()
```

Which plot below would result from the above code?

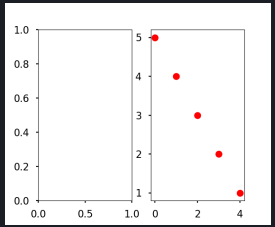
A



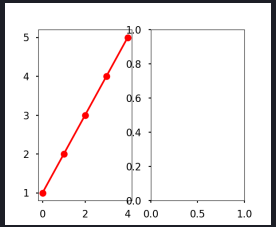
B



C



D



Extra Figure and Axes Features

- Clear communication is important with visualization
- Should always include meaningful and descriptive axes label and figure titles.
  - Axes labels controlled with `.set_xlabel()` and `.set_ylabel()`
  - Axes title controlled with `.set_title`
  - Figure title controlled with `.suptitle`

Extra Figure and Axes Features

- Clear communication is important with visualization
- Should always include meaningful and descriptive axes label and figure titles.
  - Axes labels controlled with `.set_xlabel()` and `.set_ylabel()`
  - Axes title controlled with `.set_title`
  - Figure title controlled with `.suptitle`
- Adjust tick spacing or labels:
  - Customize autoscale limits: `.set_xlim()` or `.set_ylim()`
  - Exactly where ticks appear: `.set_xticks()` or `.set_yticks()`
  - What labels they have: `.set_xticklabels()` or `.set_yticklabels()`

Extra Figure and Axes Features

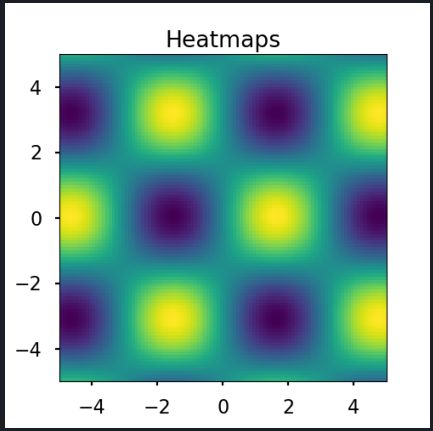
- Clear communication is important with visualization
- Should always include meaningful and descriptive axes label and figure titles.
  - Axes labels controlled with `.set_xlabel()` and `.set_ylabel()`
  - Axes title controlled with `.set_title`
  - Figure title controlled with `.suptitle`
- Adjust tick spacing or labels:
  - Customize autoscale limits: `.set_xlim()` or `.set_ylim()`
  - Exactly where ticks appear: `.set_xticks()` or `.set_yticks()`
  - What labels they have: `.set_xticklabels()` or `.set_yticklabels()`
- Overall plot style:
  - See available styles with `plt.style.available`
  - Use with `plt.style.use('name of style')`

Shared Axes

- Commonly might want to shared an axis in a plot
  - Overlaying plots on overlapping axes
  - Side by side plots with linked axes
- Easy method for overlapping:
  - Use `.twinx()` or `.twiny()`
  - Will automatically add the new axis to the opposite side of the plot as the current axis
- For more manual control:
  - Use `sharex` and `sharey` options when creating the axes object

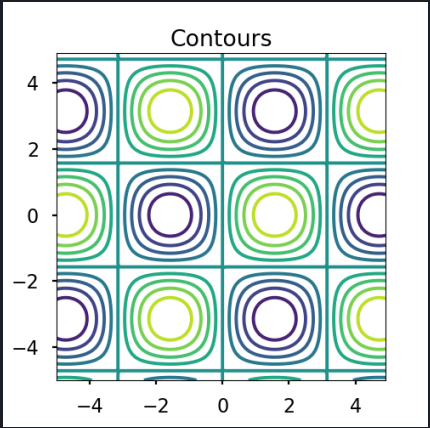
Types of Plots

- Heatmaps: `.imshow()`
- Contours: `.contour()`
- Histograms: `.hist()`
- Bar plots: `.bar()` or `.barh()`
- Pie Charts: `.pie()`



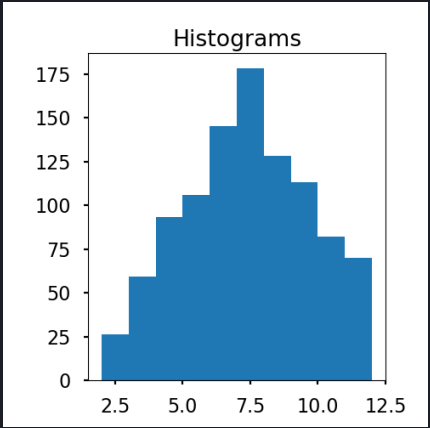
Types of Plots

- Heatmaps: `.imshow()`
- Contours: `.contour()`
- Histograms: `.hist()`
- Bar plots: `.bar()` or `.barh()`
- Pie Charts: `.pie()`



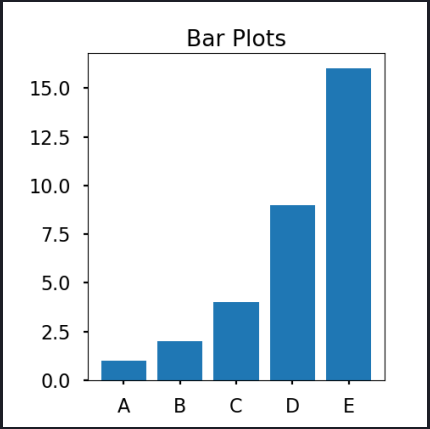
Types of Plots

- Heatmaps: `.imshow()`
- Contours: `.contour()`
- Histograms: `.hist()`
- Bar plots: `.bar()` or `.barh()`
- Pie Charts: `.pie()`



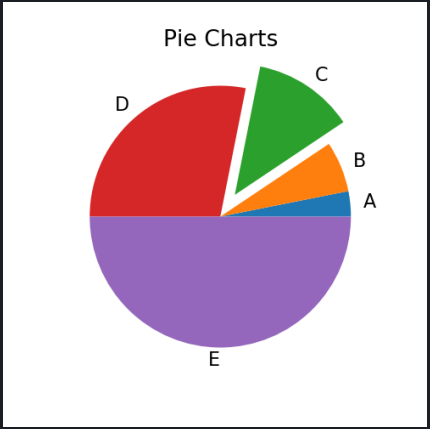
Types of Plots

- Heatmaps: `.imshow()`
- Contours: `.contour()`
- Histograms: `.hist()`
- Bar plots: `.bar()` or `.barh()`
- Pie Charts: `.pie()`



Types of Plots

- Heatmaps: `.imshow()`
- Contours: `.contour()`
- Histograms: `.hist()`
- Bar plots: `.bar()` or `.barh()`
- Pie Charts: `.pie()`



A Classy Arcade

- We've been using `arcade` throughout the semester, but not to its full potential
- Like Matplotlib, Arcade is comprised of multiple classes that define specific objects
- We can inherit from those objects and then make small changes to get large amounts of functionality with relatively little effort!

Window to the Future

- The primary class we can inherit from and use is `arcade.Window`
  - The same class that was being formed when you used to call `arcade.open_window()`
- The window class has a huge amount of predefined methods that we can override to provide almost any sort of flexibility
  - `on_draw` for basic drawing
  - `on_update` for animation
  - Methods to get location and input from the mouse
  - Methods to get keyboard input
  - Methods to control what happens if the window is resized
  - etc