

Lecture 19: Visualizations

CS 51P

November 13, 2023

Last Week: Dictionaries

- a data structure that stores key:value pairs

```
d = {'apple': .99, 'banana': .19, 'cantalope': 2.99}
```

- restriction:
 - key must be immutable

Last Week: Dictionary Operations

adding to a dictionary

- `a_dict[key] = value`
- `a_dict.update(b_dict)`

removing from a dictionary

- `del (a_dict[key])`
- `a_dict.pop(key)`
 - returns `a_dict[key]`

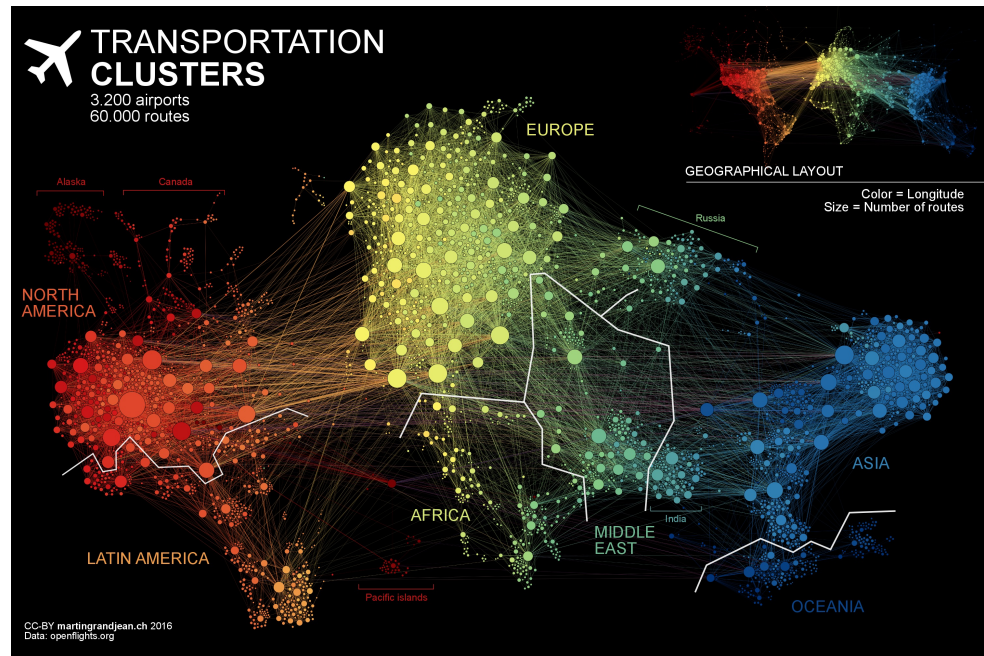
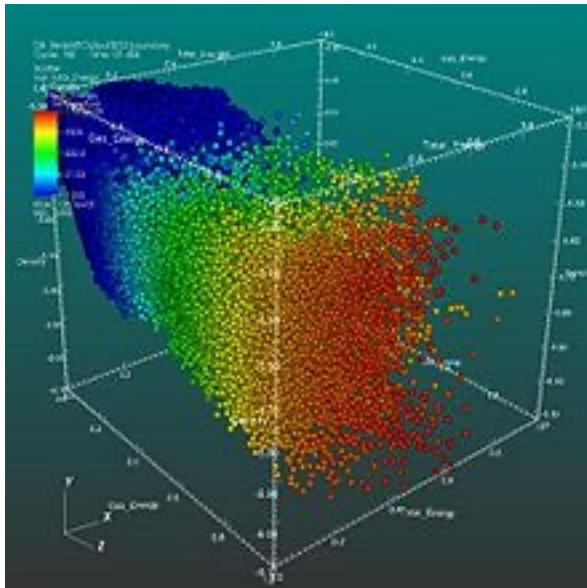
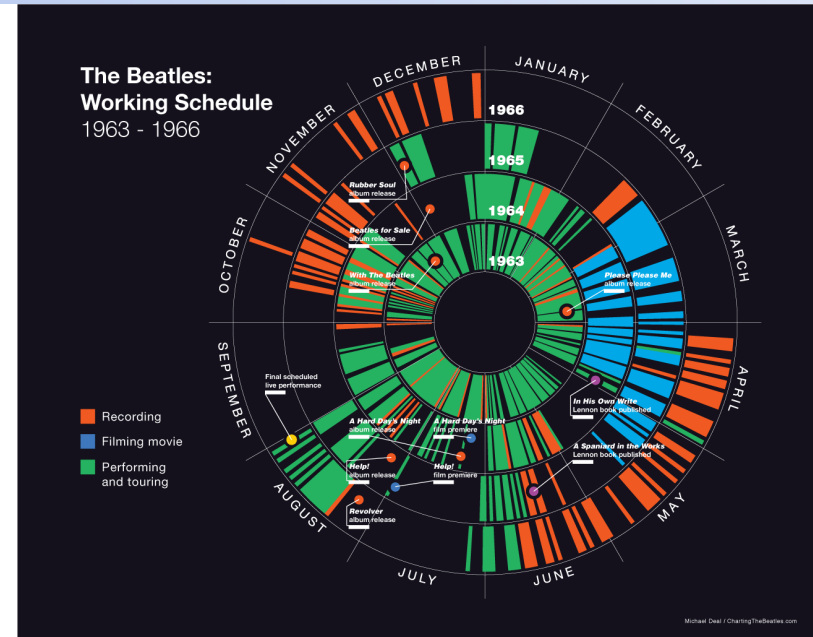
other

- `len(a_dict)`
- `a_dict.keys()`
 - returns list
- `a_dict.values()`
 - returns list
- `a_dict.items()`
 - returns list of tuples
- `b_dict = a_dict.copy()`
 - shallow copy!

Last Week: Lists vs. dictionaries

- Both data structures.
- Why would you choose one over the other?
 - a data structure is something that holds a collection of data and that supports certain operations for working with that data
- Lists: sequential access
- Dictionaries: fast lookup

Data Visualization

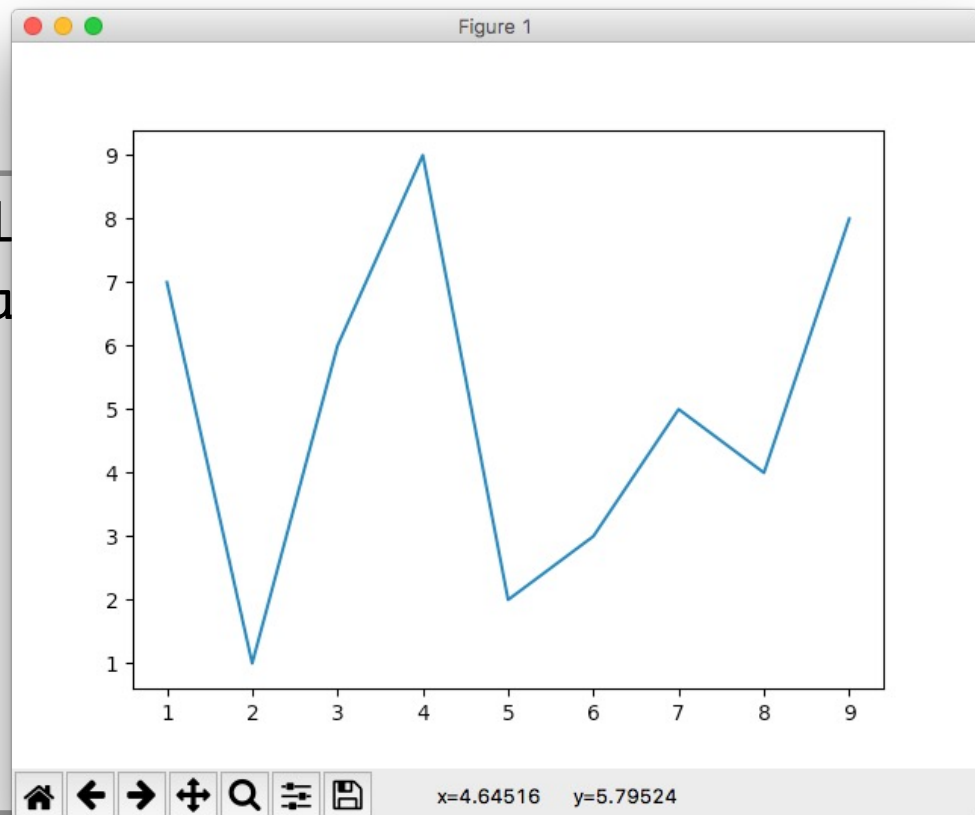


Data Visualization

- matplotlib
 - <https://matplotlib.org/index.html>
 - based on Matlab

```
import matplotlib.pyplot as plt
from random import shuffle

x = list(range(1,10))
y = x.copy()
shuffle(y)
plt.plot(x, y)
plt.show()
```



Exercise

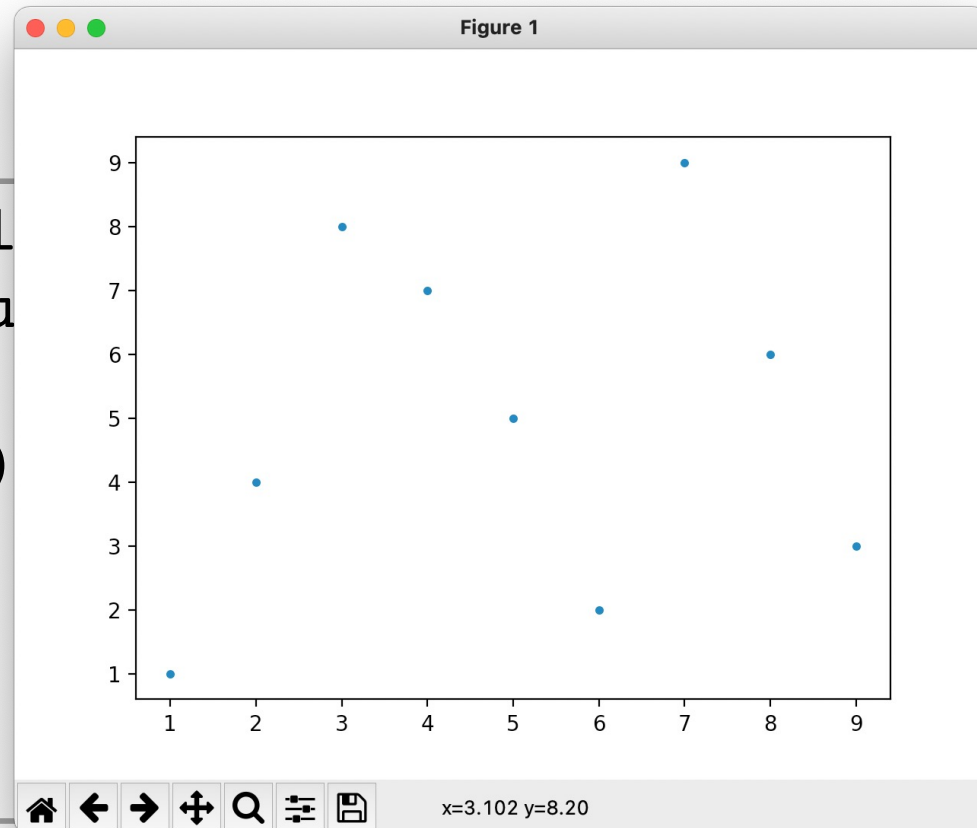
- Generate a line graph showing the powers of 2 from 1, 2, 4, 8, ... 1024

Scatter Plots

- matplotlib
 - <https://matplotlib.org/index.html>
 - based on Matlab

```
import matplotlib.pyplot as plt
from random import shuffle

x = list(range(1, 10))
y = x.copy()
shuffle(y)
plt.plot(x, y, '.')
plt.show()
```



Customizing

```
import matplotlib.pyplot  
from random import shuffle
```

```
x = list(range(1, 10))
```

```
y = x.copy()
```

```
shuffle(y)
```

```
plt.plot(x, y, 'rx' , label='random' ,
```

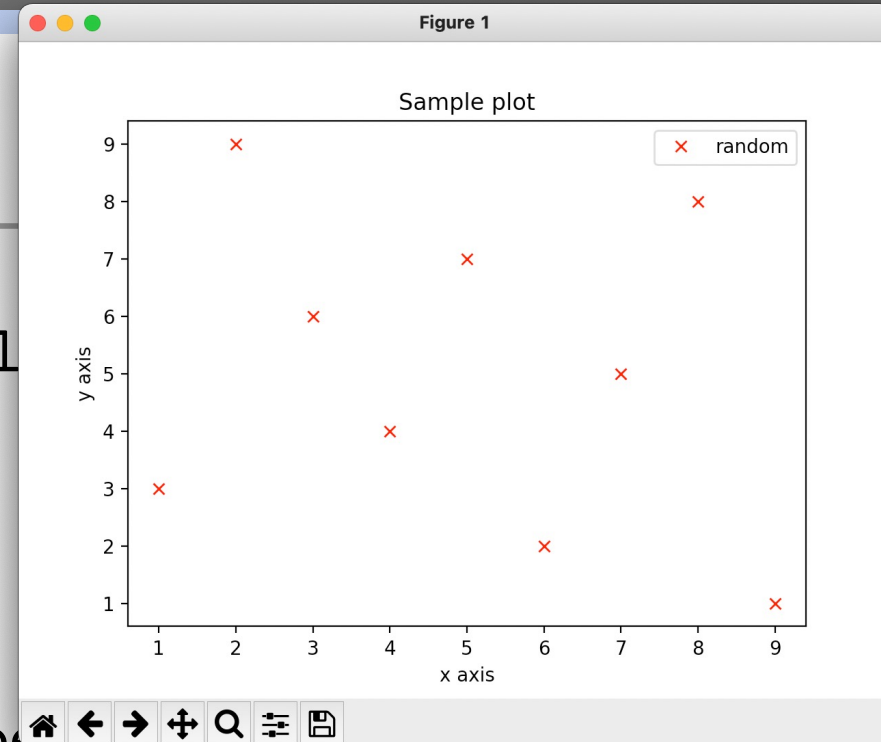
```
plt.xlabel("x axis")
```

```
plt.ylabel("y axis")
```

```
plt.title("Sample plot")
```

```
plt.legend()
```

```
plt.show()
```



Exercise

- Modify your plot of the powers of 2 from 1, 2, 4, 8, ... 1024 to include axis labels, a title, and a legend.

Multiple Datasets

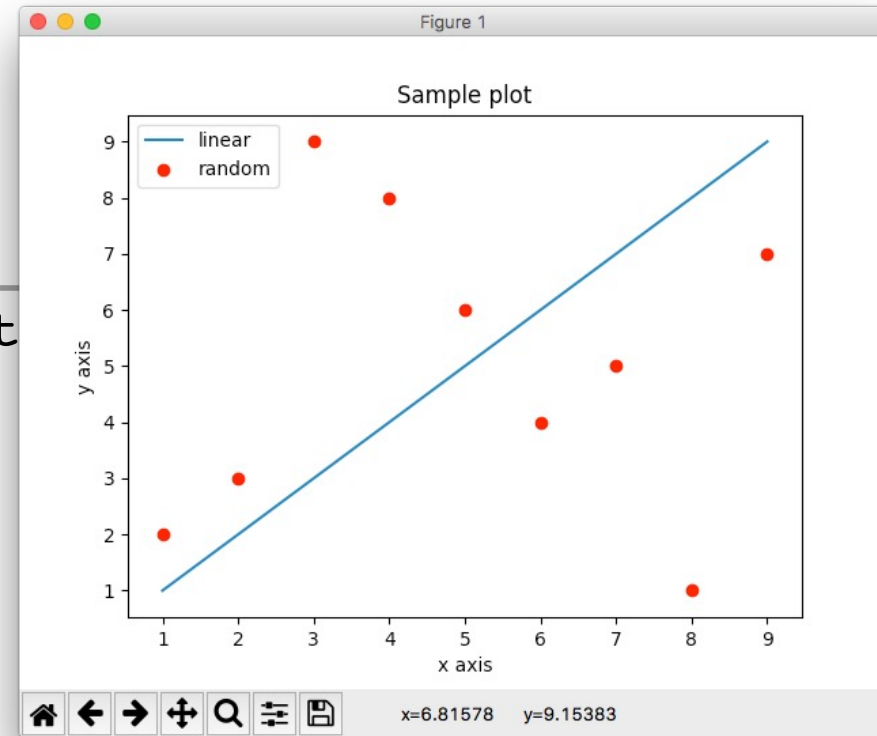
```
import matplotlib.pyplot as plt
from random import shuffle

x = list(range(1,10))
y = x.copy()
shuffle(y)

plt.plot(x, x, label="linear")
plt.plot(x, y, 'r.' label="random")

plt.xlabel("x axis")
plt.ylabel("y axis")
plt.title("Sample plot")
plt.legend()

plt.show()
```



Bar Graphs

```
import matplotlib.pyplot as plt
from random import shuffle
```

```
x = list(range(1, 5))
```

```
y = x.copy()
```

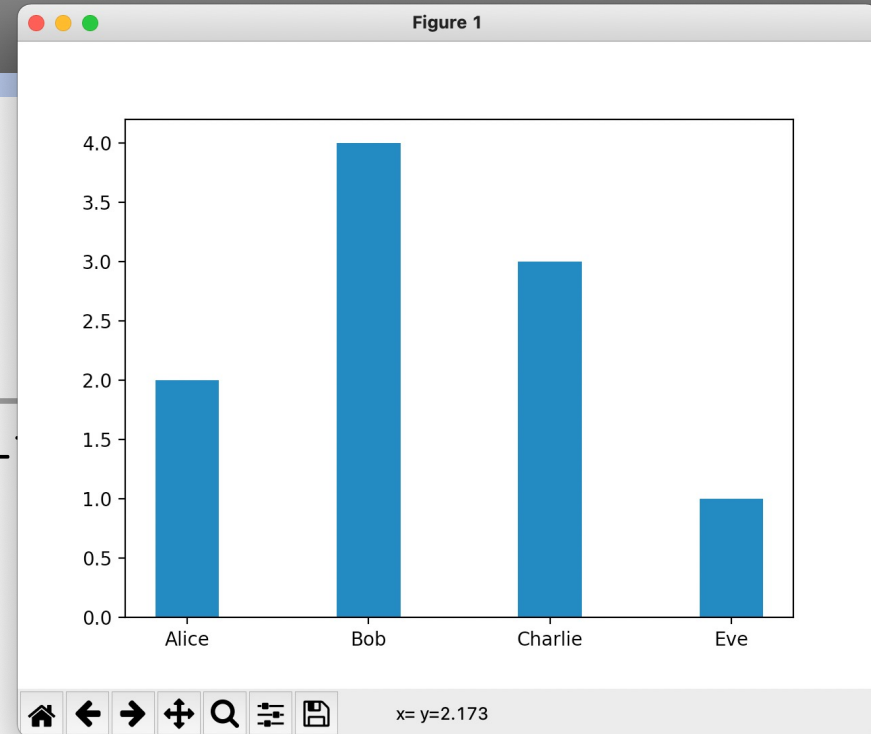
```
shuffle(y)
```

```
bar_width = .35
```

```
plt.bar(x, y, bar_width)
```

```
plt.xticks(x, ('Alice', 'Bob', 'Charlie', 'Eve'))
```

```
plt.show()
```



Bar Graphs w/ Multiple

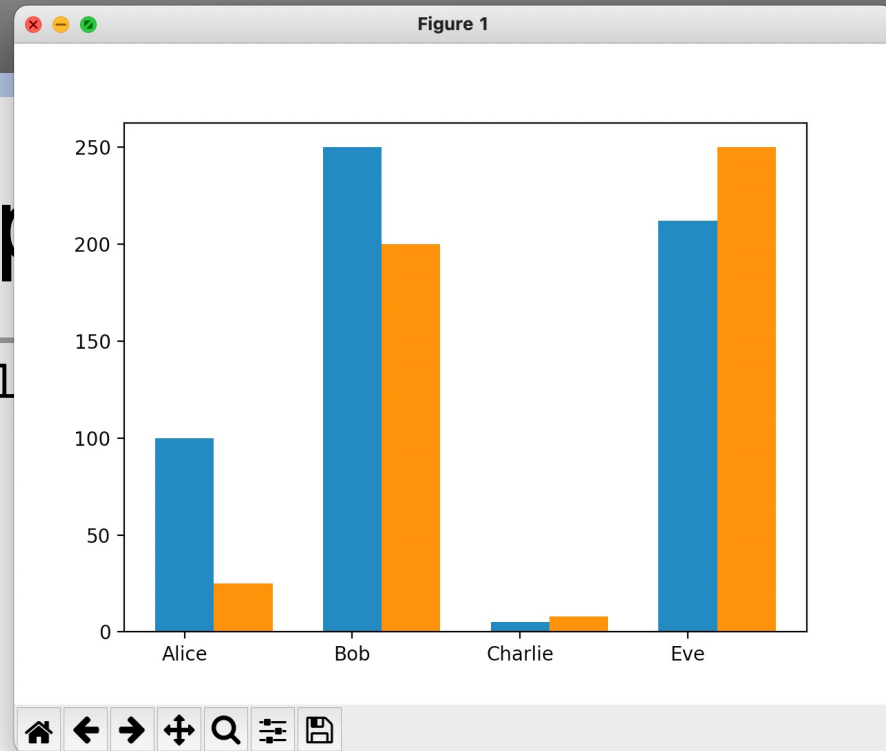
```
import matplotlib.pyplot as plt
from random import shuffle
```

```
x = list(range(0, 4))
y1 = [100, 250, 5, 212]
bar_width = .35
```

```
x2 = [x[0] + bar_width, x[1] + bar_width, x[2] +
bar_width, x[3] + bar_width]
y2 = [25, 200, 8, 250]
```

```
plt.bar(x, y1, bar_width)
plt.bar(x2, y2, bar_width)
plt.xticks(x, ('Alice', 'Bob', 'Charlie', 'Eve'))

plt.show()
```

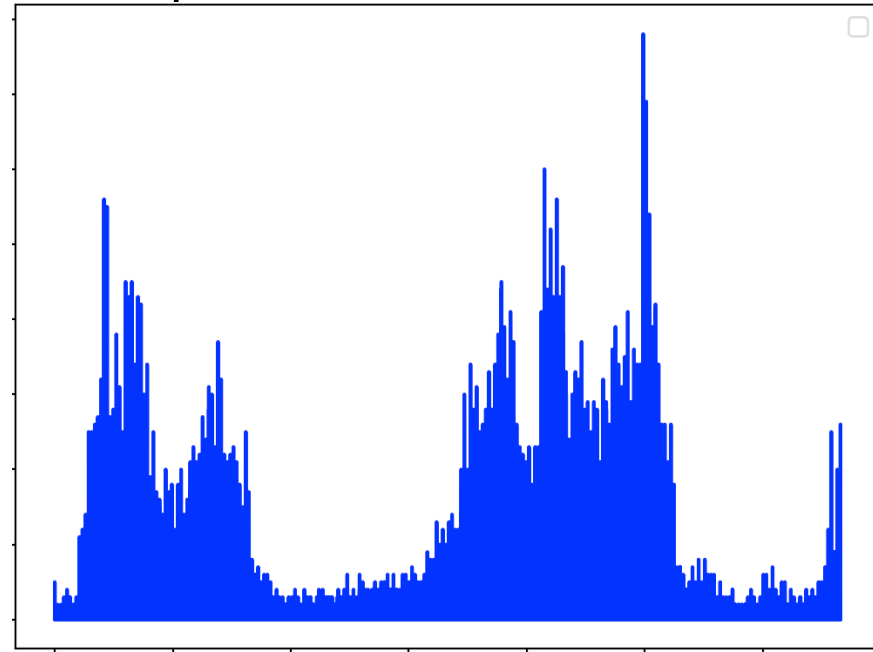


Exercise

- Generate a bar graph of the powers of 2 from 1, 2, 4, 8, ... 1024

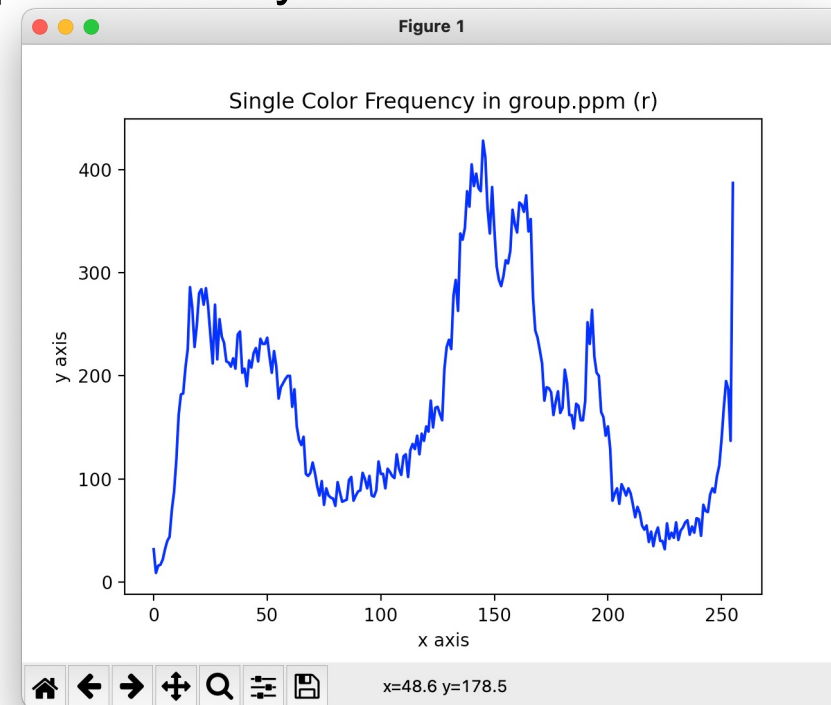
Exercise

- Write a function `plot_ppm` that takes a filename as input and produces an image histogram (a plot of the number of pixels for each rgb value).
- Hint: you will want to use a bar graph
- Hint: r g b values in a ppm file are represented by three numbers (e.g., **250** **50** **100**). You can represent this pixel as a number **250***(255*255) + **50***(255) + **100**



Exercise

- Write a function `plot_ppm_color` that takes two arguments, a filename and a color option ("r", "g", or "b"), as input and produces a graph of the frequencies of that color (a plot of the number of pixels for each value for that color)
 - Hint: r g b values in a ppm file are represented by three numbers (e.g., **250** **50** **100**).



Exercise: Visualizing Text Files

