

Lecture 19: Plotting

CS 51P

November 9, 2022



Tom Yeh
he/him/his

Previously: Dictionaries

- a data structure that stores key:value pairs

```
price_list = {'apple': .99, 'banana': .19, 'melon': 2.99}
```

- restriction:
 - key must be immutable

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.get(key)`
 - returns associated value
- `a_dict.keys()`
 - returns list
- `a_dict.values()`
 - returns list
- `a_dict.items()`
 - returns list of tuples
- `b_dict = a_dict.copy()`
 - shallow copy!

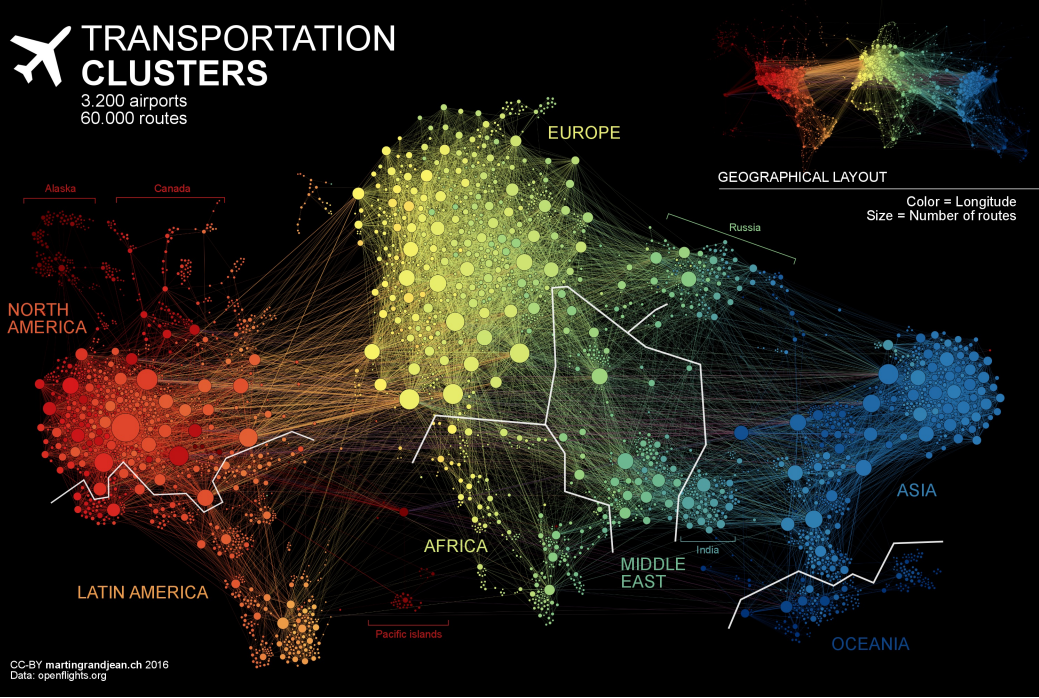
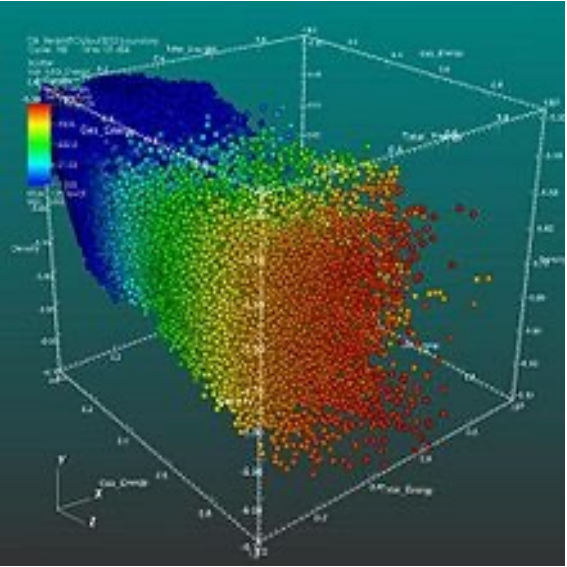
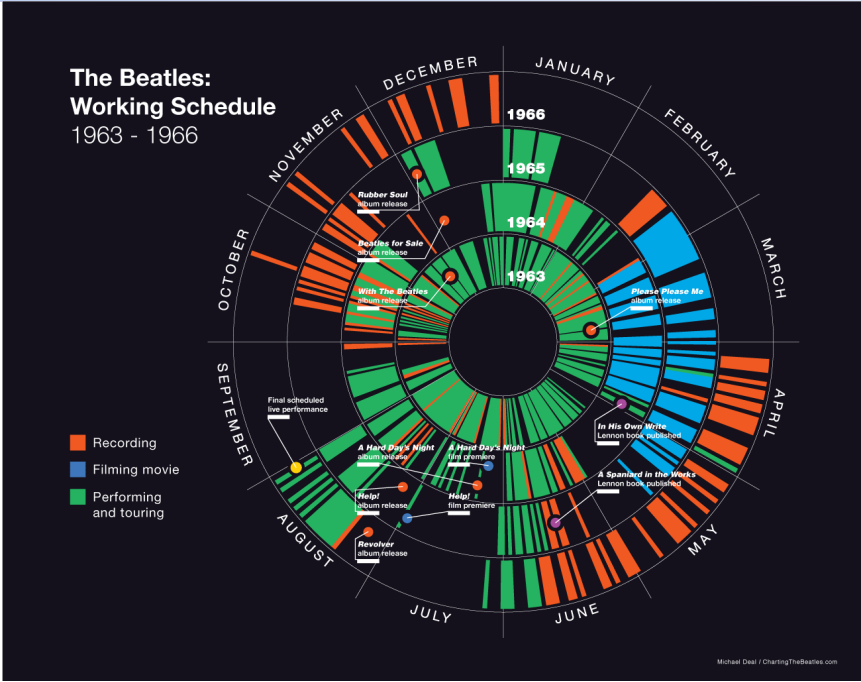
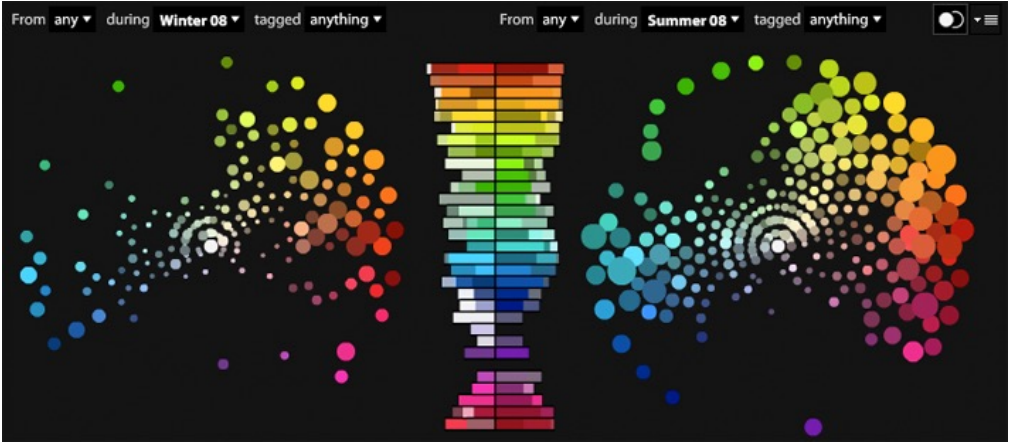
```
def mystery(my_dict):
    d = {}
    for key in my_dict.keys():
        if my_dict[key] in d:
            d[my_dict[key]].append(key)
        else:
            d[my_dict[key]] = [key]
    return d

def main():
    d = {"a":1, "b":2, "c":1, "d":0, "e":2}
    print(mystery(d))

main()
```

What does this code print?

Data Visualization

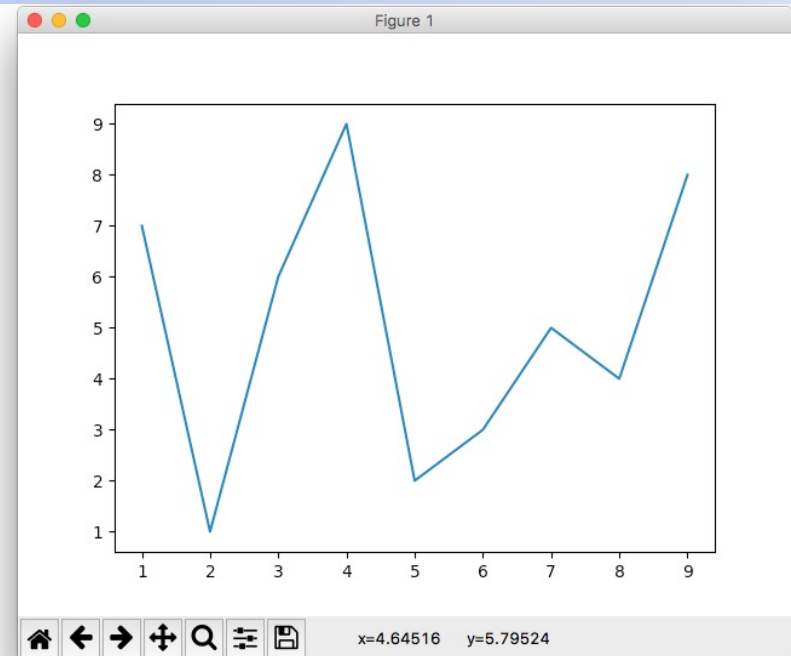


Data Visualization



Data Visualization

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



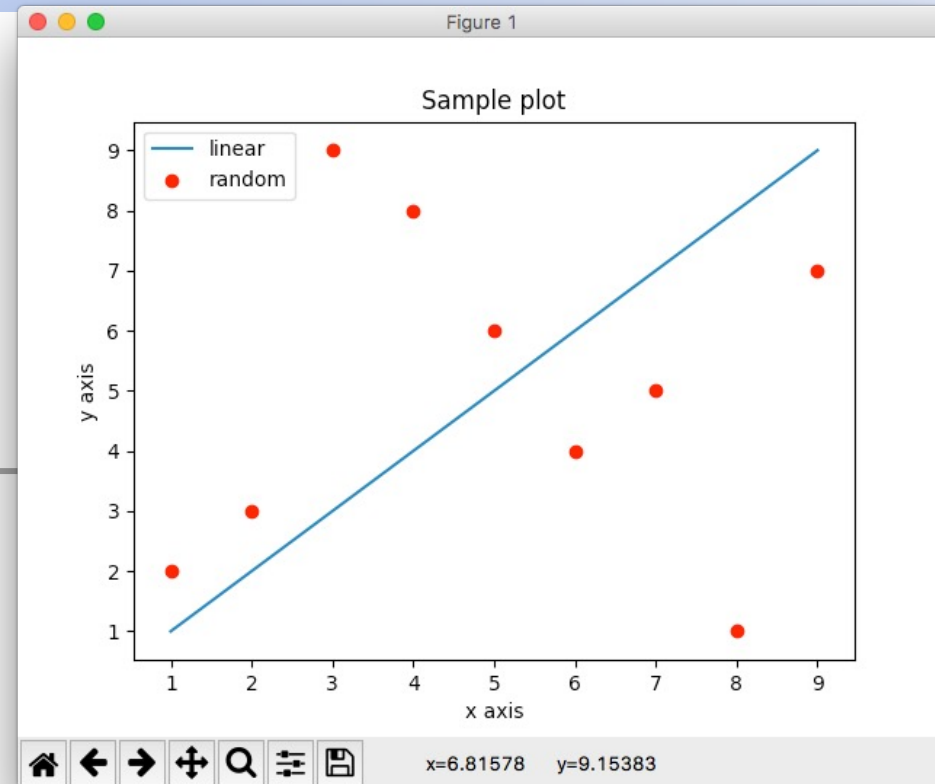
```
import matplotlib.pyplot as plt
from random import shuffle
```

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

Customizing

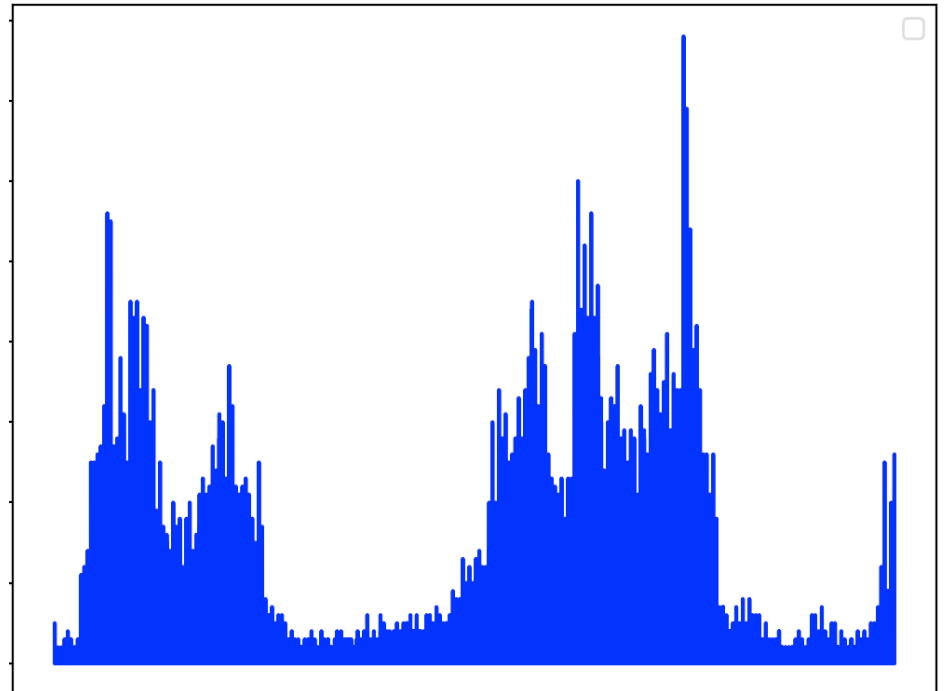
```
import matplotlib.pyplot as plt
from random import shuffle
```

```
x = list(range(1,10))
y = x.copy()
shuffle(y)
plt.plot(x, x, label="linear")
p = plt.plot(x, y, 'r.' label="random")
plt.xlabel("x axis")
plt.ylabel("y axis")
plt.title("Sample plot")
plt.legend()
plt.show()
```



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: 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 $250*(255*255) + 50*(255) + 100$



Dictionary Operations

File Functions

- `f = fopen(fname, mode)`
 - open file named `fname` in mode
- `f.close`
 - close `f`
- `f.readlines()`
 - returns list

String Functions

- `s.split()`
- `s.strip(string.punctuation)`
 - removes punctuation

Dictionary Functions

- `a_dict[key] = value`
 - set, update
- `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!

Example: Word Count

- Write a function that processes a file and returns a dictionary for handling repeated queries of the form "How many times does the word _____ appear?"

```
import string
word_counts = {}
f = fopen("file.txt", "r")
text = f.readlines()
for line in text:
    words = line.split()
    for w in words:
        w2 = w.strip(string.punctuation)
        if w2 in word_counts:
            word_counts[w2] = word_counts[w2] + 1
        else:
            word_counts[w2] = 1
```