

Python for Probability: Part 3

CS 109 SPRING 2020

Slides by Julie Wang
Spring 2020

Contents

Plotting

- Regular graphs
- Bar graphs

Python Data Structures

- Dictionaries
- Tuples

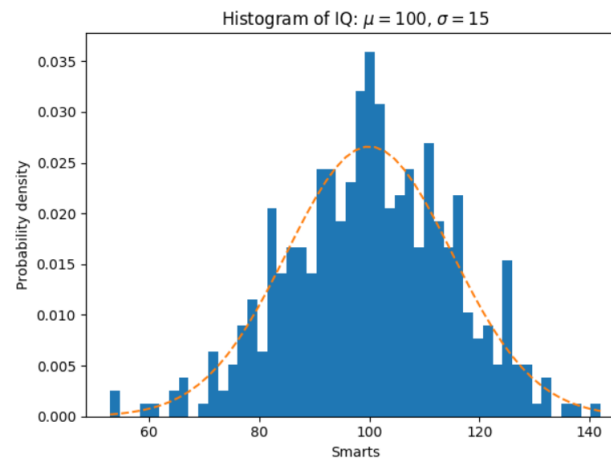
Better Math with Numpy

- Mean, variance, median
- Efficient array operations

Questions/Ask about any topic

Plotting

MATPLOTLIB



Making plots in python

Install Matplotlib (command line)

```
pip3 install matplotlib
```

In your .py file, import the package

```
import matplotlib.pyplot as plt
```

If using Jupyter notebook, run the following to get inline plots

```
%matplotlib inline
```

Given a list or array of data, python plots away happily:

```
x = np.arange(0, 3 * np.pi, 0.1)
```

```
y_cos = np.cos(x)
```

```
y_sin = np.sin(x)
```

```
# Plot the points using matplotlib
```

```
plt.plot(x, y_sin)
```

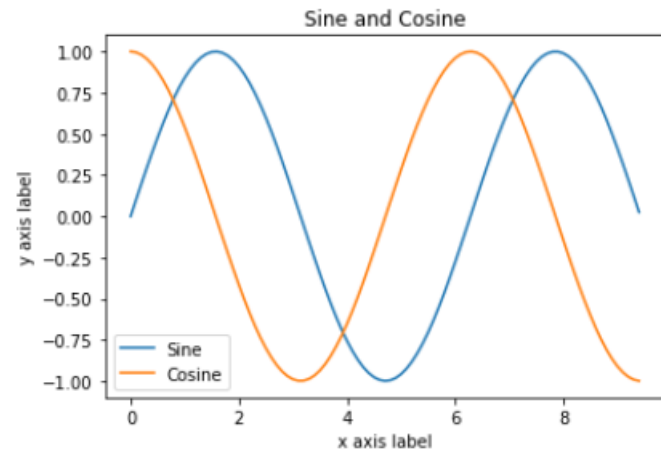
```
plt.plot(x, y_cos)
```

```
plt.xlabel('x axis label')
```

```
plt.ylabel('y axis label')
```

```
plt.title('Sine and Cosine')
```

```
plt.legend(['Sine', 'Cosine'])
```



More plots

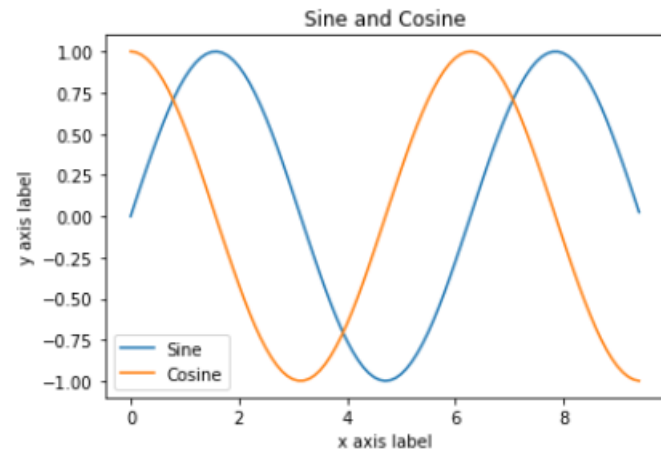
Histograms ([full reference](#))

```
x = np.random.rand(50)
plt.hist(x)
plt.show()
```

```
#To make bins, give a sequence of ints
#bins [0, .25), [.25, .5), [.5, .75), [.75, 1]
plt.hist(x, [0, .25, .5, .75, 1])
```

Saving a figure

```
plt.savefig('my_plot.png')
```



Data Structures

BEYOND LISTS



Structures you've seen already

Lists

```
a = [1, 2, 3, 4] #make a list with 1, 2, 3, 4
a += [5] # appends 5 to end of list
b = [0] * 100 #makes a list of size 100 with 0's
c = [x for x in range(42, 45)] #makes [42, 43, 44]
len(b) # gets length of b
```

Numpy Arrays

```
import numpy as np
a = np.zeros((3,4)) # makes array of zeros, 3x4
a.shape # prints shape as tuple
b = np.random.rand(3,4) # makes random array, 3x4
a = np.zeros(4) #makes vector of length 4
```

More data structures

Tuple – Immutable Sequence

```
tup1 = ('probability', 'is', 'awesome', 44)
tup2 = (42,) # must include comma at end if one entry
tup2[2] # accesses 42
len(tup1) # length of tuple
```

Dictionary: Associates a key with a value. Key must be immutable

```
my_dict = {} #Makes an empty dictionary
my_dict[1] = 'apple'
my_dict[(0, 0)] = 'origin'
my_dict["orange"] = 6
my_dict.keys() # returns all keys as iterable obj
my_dict.values() # returns all values as iterable obj
[x for x in my_dict.keys()] # puts all keys into a list
```


Better Numpy Math

ARRAY BROADCASTING



Useful Functions

Mean, Median, Variance, sum

```
import numpy as np
a = np.arange(1, 15)
np.mean(a)
np.median(a)
np.var(a)
np.sum(a)
```

Sample from various distributions

```
# Sample 100 times from a normal, mean 2, std dev 3
a = np.random.normal(loc=2, scale=3, size=100)
# Sample 2x3 array from a uniform, [0, 10)
b = np.random.uniform(0, 10, (2,3))
```

Numpy Axes and Dimensions

Size of an array can be multidimensional

```
a = np.random.rand(2,3) # 2x3 2d array
```

```
b = np.random.rand(2,3,4) #2x3x4 3d array
```

```
c = np.random.rand(2,3,4,5) #2x3x4x5 4d array
```

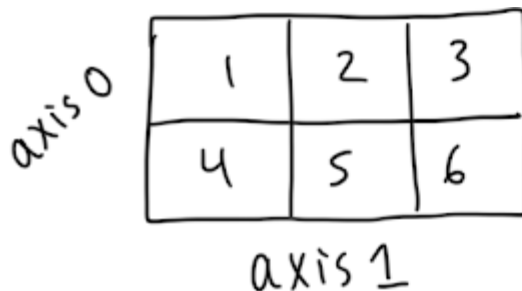
Each axis has a label, use this when taking stats

```
# Take mean over rows, axis 0
```

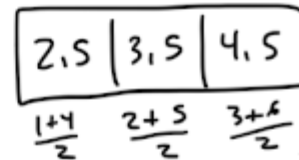
```
np.mean(a, axis=0)
```

```
# Take mean over columns, axis 1
```

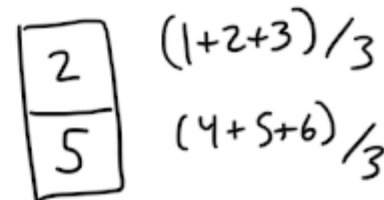
```
np.mean(a, axis=1)
```



$\text{np.mean}(a, \text{axis}=0)$



$\text{np.mean}(a, \text{axis}=1)$



Numpy: Operations on Arrays

Numpy automatically broadcasts operations to size of the array

```
a = np.random.rand(2,3) # 2x3 2d array
a * 5 + 4 #makes an array of size 2x3 with 4's
```

Elementwise math works the way you expect

```
a = np.array([[1, 2, 3],[4, 5, 6]])
b = np.array([[10, 20, 30],[40, 50, 60]])
c = a + b # c is [[11, 22, 33], [44, 55, 66]]
c = a * b # c is [[10, 40, 90],[160, 250, 360]]
```

Dot products and matrix operations done with “@”, transpose with .T

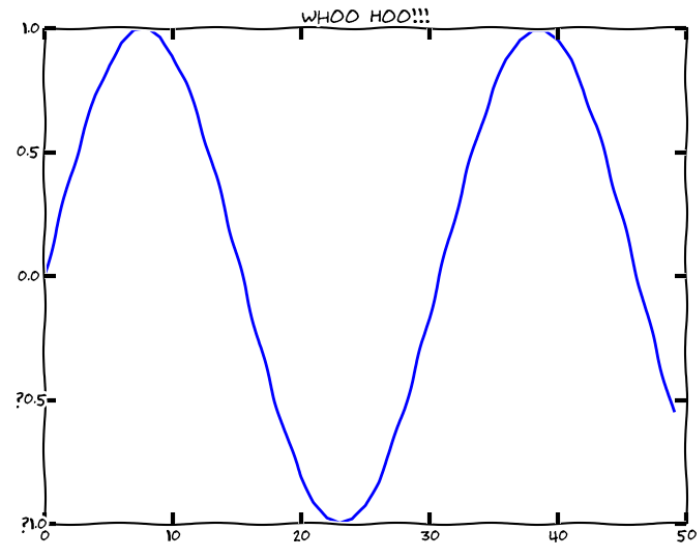
```
a @ b.T
array([[140, 320],
       [320, 770]])
```

To get an outer product of two vectors, use np.outer, otherwise will just get dot product!

```
np.outer(np.array([1,2]), np.array([3,4]))
array([[3, 4],
       [6, 8]])
```

Questions?

ASK ME ANYTHING



xkcd matplotlib plotter

```
from matplotlib import pyplot as plt
import numpy as np

plt.xkcd()

fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
plt.xticks([])
plt.yticks([])
ax.set_ylim([-30, 10])

data = np.ones(100)
data[70:] -= np.arange(30)

plt.annotate(
    'THE DAY I REALIZED\nI COULD COOK\nBACON\nWHENEVER I WANTED',
    xy=(70, 1), arrowprops=dict(arrowstyle='->'), xytext=(15, -10))

plt.plot(data)

plt.xlabel('time')
plt.ylabel('my overall health')
plt.show()
```

