

CS101: Intro to Computing

Fall 2015

Lecture 19

Administrivia

- Homework 12 due on today
- Homework 13 coming Wednesday
- Midterm 2: November 16th

REVIEW

```
x=np.zeros((3,3))  
for i in range(3):  
    for j in range(3):  
        x[i][j]=i*j+j
```

A

0	0	0
1	2	3
2	4	6

B

0	0	0
0	2	4
0	4	8

C

0	1	2
0	2	4
0	3	6

```
x=np.zeros((3,3))  
for i in range(3):  
    x[i][i]=1  
    for j in range(3):  
        if i<j:  
            x[i][j]=2
```

A

1	2	2
0	1	2
0	0	1

B

1	0	0
2	1	0
2	2	1

C

2	2	1
2	1	0
1	0	0

```
x=np.zeros((3,3))  
for i in range(3):  
    x[i][0]=i  
    for j in range(1,3):  
        x[i][j]=x[i][j-1]+1
```

A

2	1	0
3	2	1
4	3	2

B

0	0	0
1	1	1
2	2	2

C

0	1	2
1	2	3
2	3	4

EXCEPTIONS

Exceptions

- Represent computation reaching an exceptional (unexpected or unusual) state
- Exceptions are “thrown” when we reach the state

```
print x
```

- If exception is not *caught* (or *handled*) Python will print a *trace*
 - list of lines of code that were

Handling Exceptions

- Exceptions can be caught using the *try/except* structure

```
try:
```

```
    a=[1,2]
```

```
    print a[2]
```

```
except:
```

```
    print "Oh no!"
```

Throwing Exceptions

- Exceptions can be thrown with the “raise” structure

```
raise Exception("Oh no!")
```

```
x=[ 4 , 2 , 6 ]  
x=x.sort( )  
try:  
    print len(x)  
except:  
    print type(x)
```

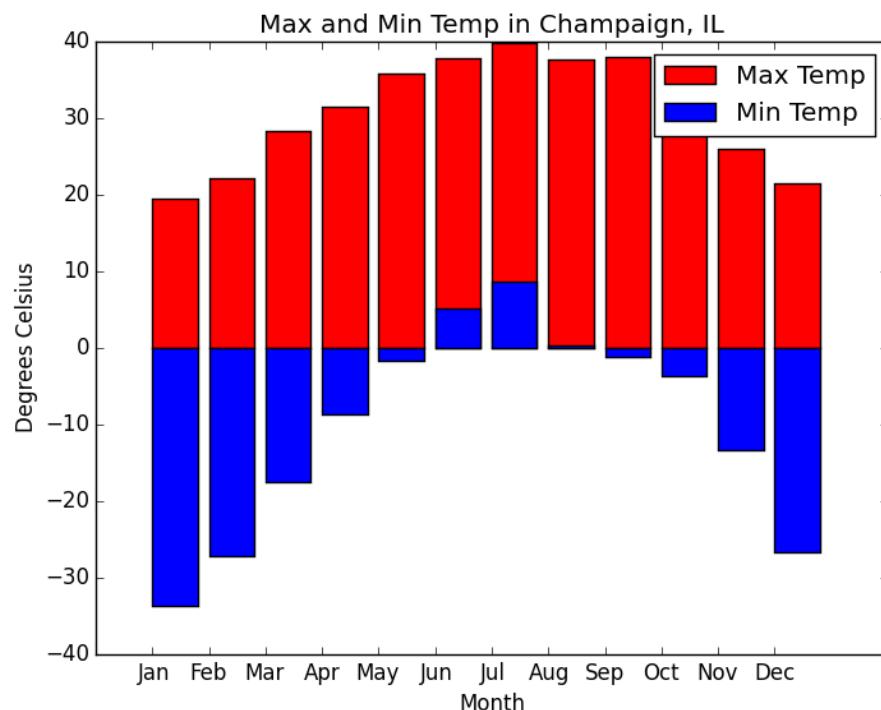
- a) TypeError
- b) 3
- c) list
- d) NoneType

MORE PLOTTING

Bar Graph

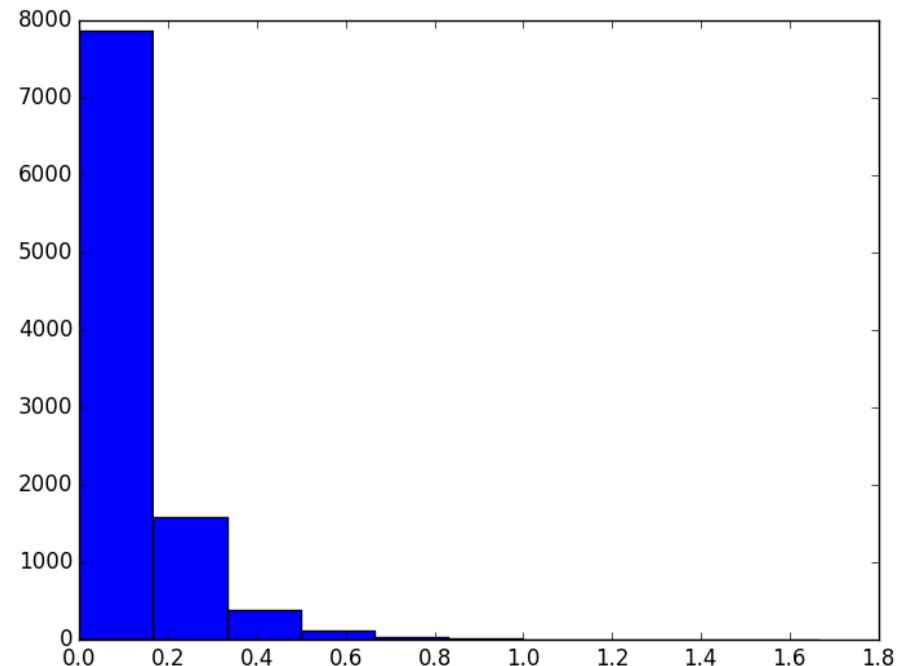
- Similar to plot, but bars are displayed
- Used to display proportions for easy comparison

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



Histogram

- Plots frequency data to display the distribution of data
- x axis displays the values in “bins”
- y axis displays frequency of values in each “bin”
- Visual estimate of PDF



Histogram

```
x=arange(1,100)  
plt.hist(x)
```

PSEUDORANDOM NUMBERS

Pseudorandom numbers

- PRNG - a program that generates *seemingly* random numbers
- Starting from an initial input (seed) PRNG generates “unpredictable” numbers
- Next input is generated from previous input
- numpy implements “Mersenne Twister” PRNG

Rand

- Creates an array sampled from **uniform distribution** on interval [0,1)
- Arguments are dimensions of array

```
x=np.random.rand(10000)
```

```
plt.hist(x)
```

```
plt.show()
```

Randn

- Creates an array sampled from **standard normal distribution**
 - Mean 0 and variance 1
- Arguments are dimensions of array

```
x=np.random.randn(10000)
```

```
plt.hist(x)
```

```
plt.show()
```

Other distributions

- Lognormal
- Poisson
- Binomial
- Rayleigh
- Gamma
- Geometric
- Chi-square
- Dirichlet
- Laplace
- Beta
- Gamma
- Gumbel
- Hypergeometric
- Cauchy
- Weibull
- And more!

Choice

- Generates a random sample from a 1-D array

```
x=np.arange(1,7)  
c=np.random.choice(x)
```



Choice

- Can generate samples without replacement

```
x=np.arange(1,53)  
c=np.random.choice(  
    x, size=5,  
    replace=False)
```



Shuffle

- Randomly reposition elements of a 1-D array
- Creates a random permutation of the array elements

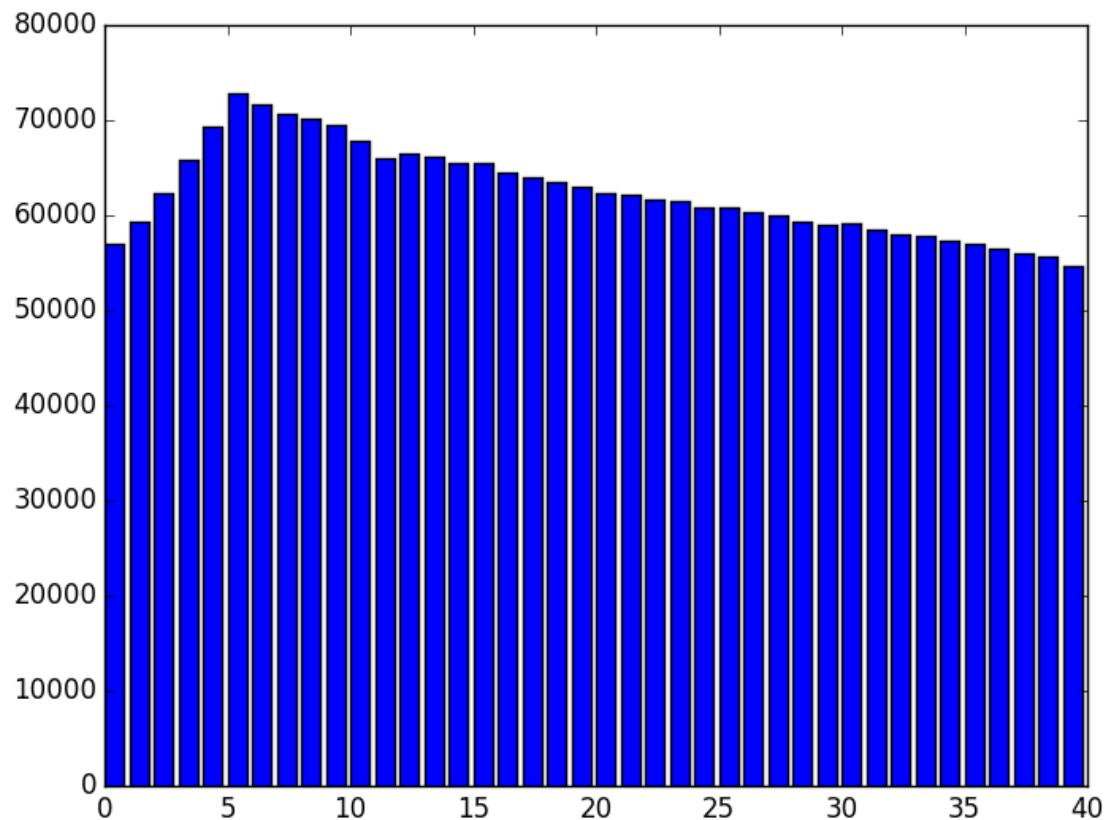
```
x=np.arange(1,53)  
np.random.shuffle(x)
```



Example



Example



Descriptive Statistics

- Can compute basic descriptive statistics like:
 - Min, max, percentile
 - Mean, median, standard deviation

```
x=np.random.rand(1000)  
print np.median(x)  
print np.mean(x)
```

Example

- A random walk in 2 dimensions
- Start in middle
- Take one step in a random direction: north, south, east, or west

Matshow

- Pyplot function to display a 2d array as an image