

Randomness & Classes

CS 177 - Recitation 10

Announcements

- Project 3 is posted (due 11/09 at 23:59 PM)
- Read files (balance.txt, transaction.txt) fetch a web-page, parse and process data

Outline

- Randomness
- Introduction to Classes

Randomness

random.random()

random.random() -> x in the interval [0, 1).

```
>>> import random
>>> random.random()
0.37801095286380204
>>> random.random()
0.5521781818878924
>>> random.random()
0.5496262107831614
>>> random.random()
0.07677937890343389
>>> random.random()
0.38419527490023464
>>> random.random()
0.04221118292371551
```

random.randint()

Return random integer in range [a, b], including both end points.

```
>>> import random
>>> random.randint(5,10)
7
>>> random.randint(5,10)
9
>>> random.randint(5,10)
10
>>> random.randint(5,10)
8
>>> random.randint(5,10)
5
```

random.randrange()

Choose a random item from range(start, stop[, step]).

This fixes the problem with randint() which includes the endpoint; in Python this is usually not what you want.

```
>>> import random
>>> random.randrange(5, 10)
8
>>> random.randrange(5, 10)
5
>>> random.randrange(5, 10)
9
>>> random.randrange(5, 10)
5
>>> random.randrange(5, 10)
6
```

random.shuffle()

Shuffle list x in place, and return None.

```
>>> import random
>>> a = [1,2,3,4,5]
>>> random.shuffle(a)
>>> a
[2, 1, 3, 5, 4]
```


bar graph using matplotlib

```
import matplotlib.pyplot as plt

pos = [1,2,3,4,5]
data = [10,16,20,30,40]

bar1 = plt.bar(pos, data, width=1, \
               color='yellow', align='center')

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

bar graph using matplotlib

```
import matplotlib.pyplot as plt

pos = [1,2,3,4,5]
data = [10,16,20,30,40]

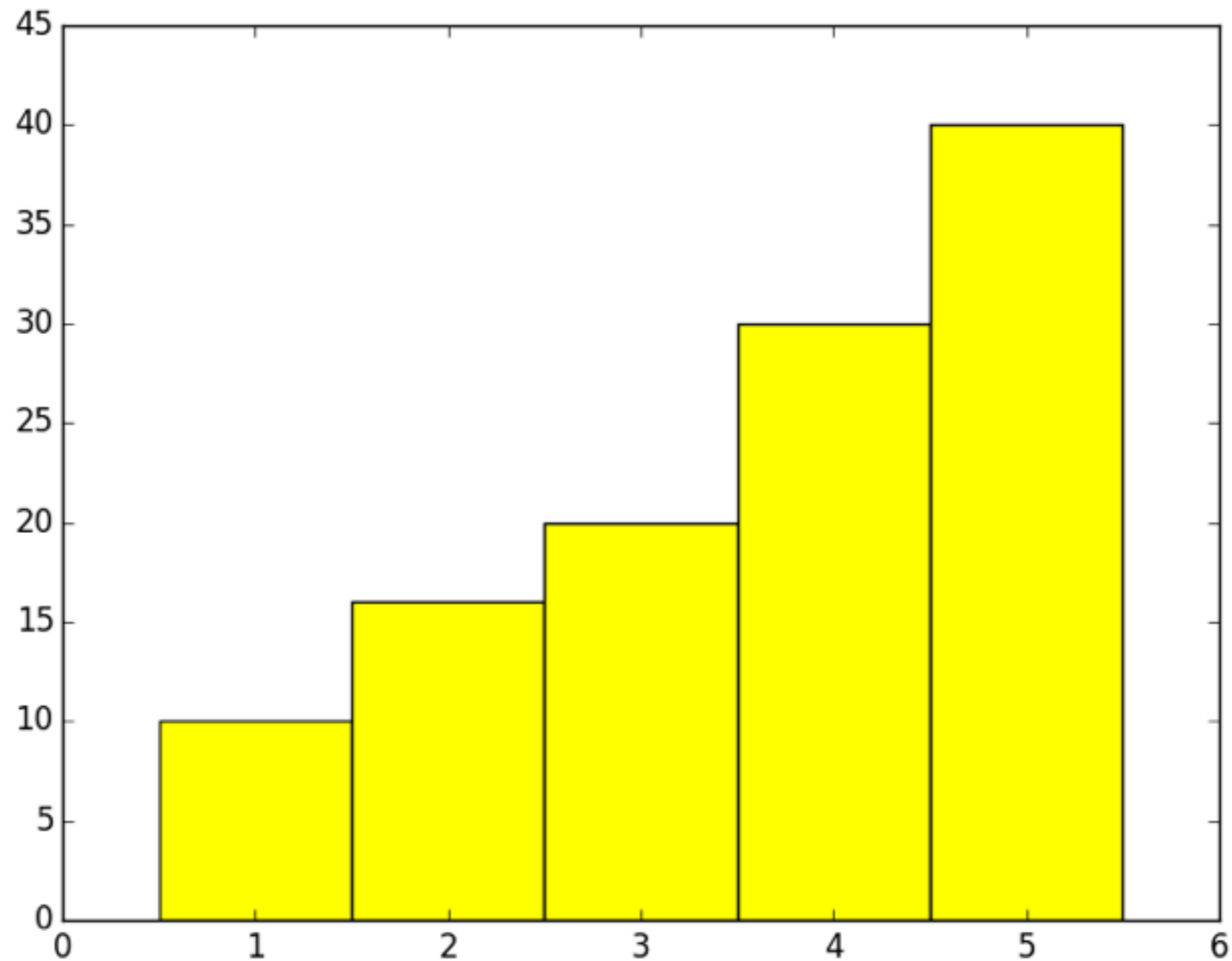
bar1 = plt.bar(pos, data, width=1, \
               color='yellow', align='center')

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

to continue in
new line



bar graph using matplotlib



generate random numbers

```
# Pseudocode  
do 100000 times:  
    a = pick a random number between [1, 10]  
plot the histogram of a's
```

generate random numbers

```
# Pseudocode  
do 100000 times:  
    a = pick a random number between [1, 10]  
plot the histogram of a's
```

How to do this in Python?

generate random numbers

```
# Pseudocode  
do 100000 times:  
    a = pick a random number between [1, 10]  
plot the histogram of a's
```

How to do this in Python?

No, please...
Somebody actually tell me
how to do this in Python.

generate random numbers

```
import random
import matplotlib.pyplot as plt

pos = range(1,11)
data = [0] * 10

for i in range(100000):
    ...?
    ...?

bar1 = plt.bar(pos, data, width=1, \
               color='yellow', align='center')

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

generate random numbers

```
import random
import matplotlib.pyplot as plt

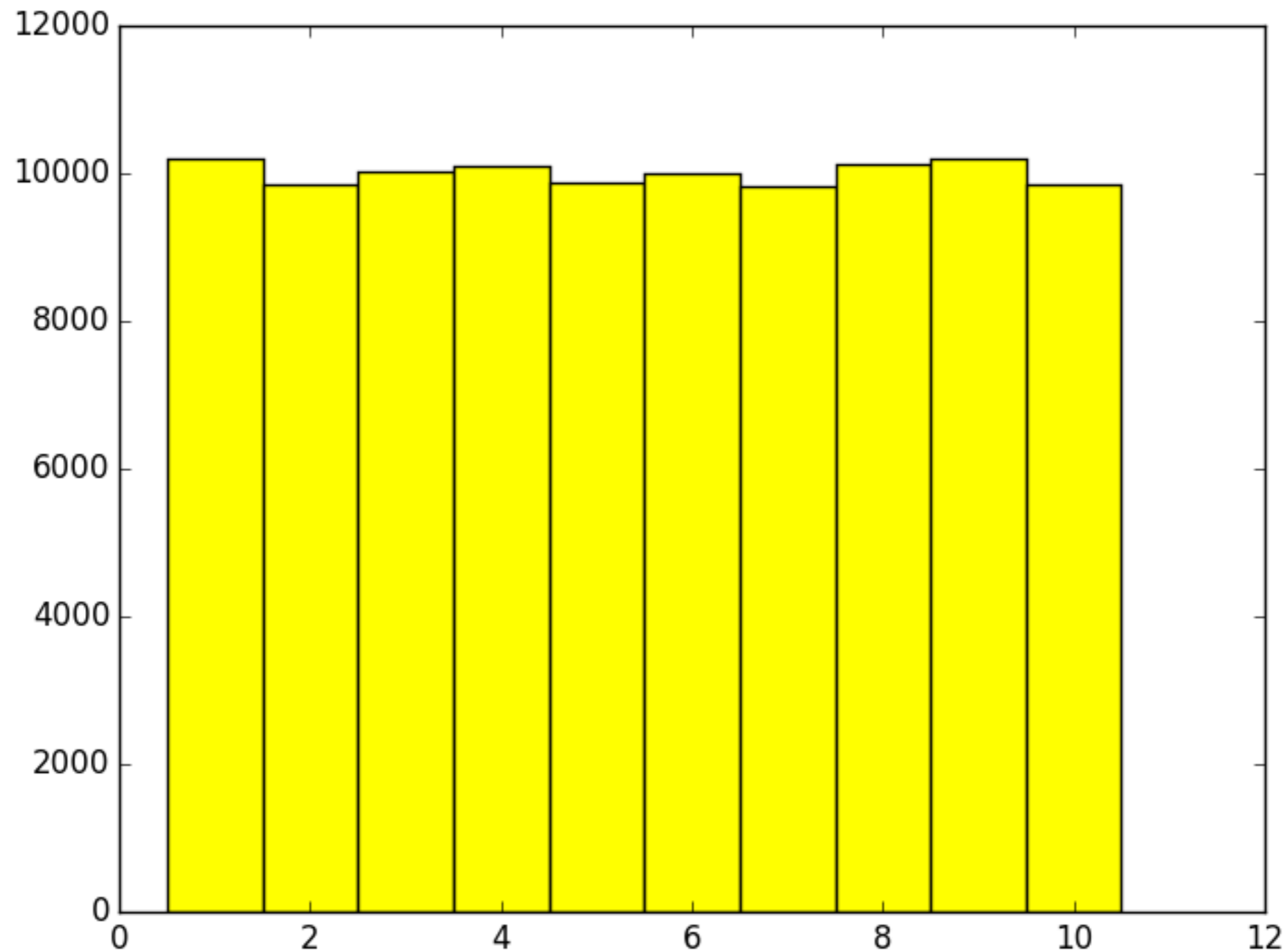
pos = range(1,11)
data = [0] * 10

for i in range(100000):
    a = random.randrange(1,11)
    data[a-1] += 1

bar1 = plt.bar(pos, data, width=1, \
    color='yellow', align='center')

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


generate random numbers



generate random numbers

```
do 100000 times:  
    a = pick a random number between [1, 10]  
    b = pick a random number between [1, 10]  
    c = a + b  
plot the histogram of c's
```

How to do this in Python?

generate random numbers

```
import random
import matplotlib.pyplot as plt

pos = ?
data = ?

for i in range(100000):
    a = random.randrange(1,11)
    b = random.randrange(1,11)
    data[ ? ] += 1

bar1 = plt.bar(pos, data, width=1, \
    color='yellow', align='center')

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

generate random numbers

```
import random
import matplotlib.pyplot as plt

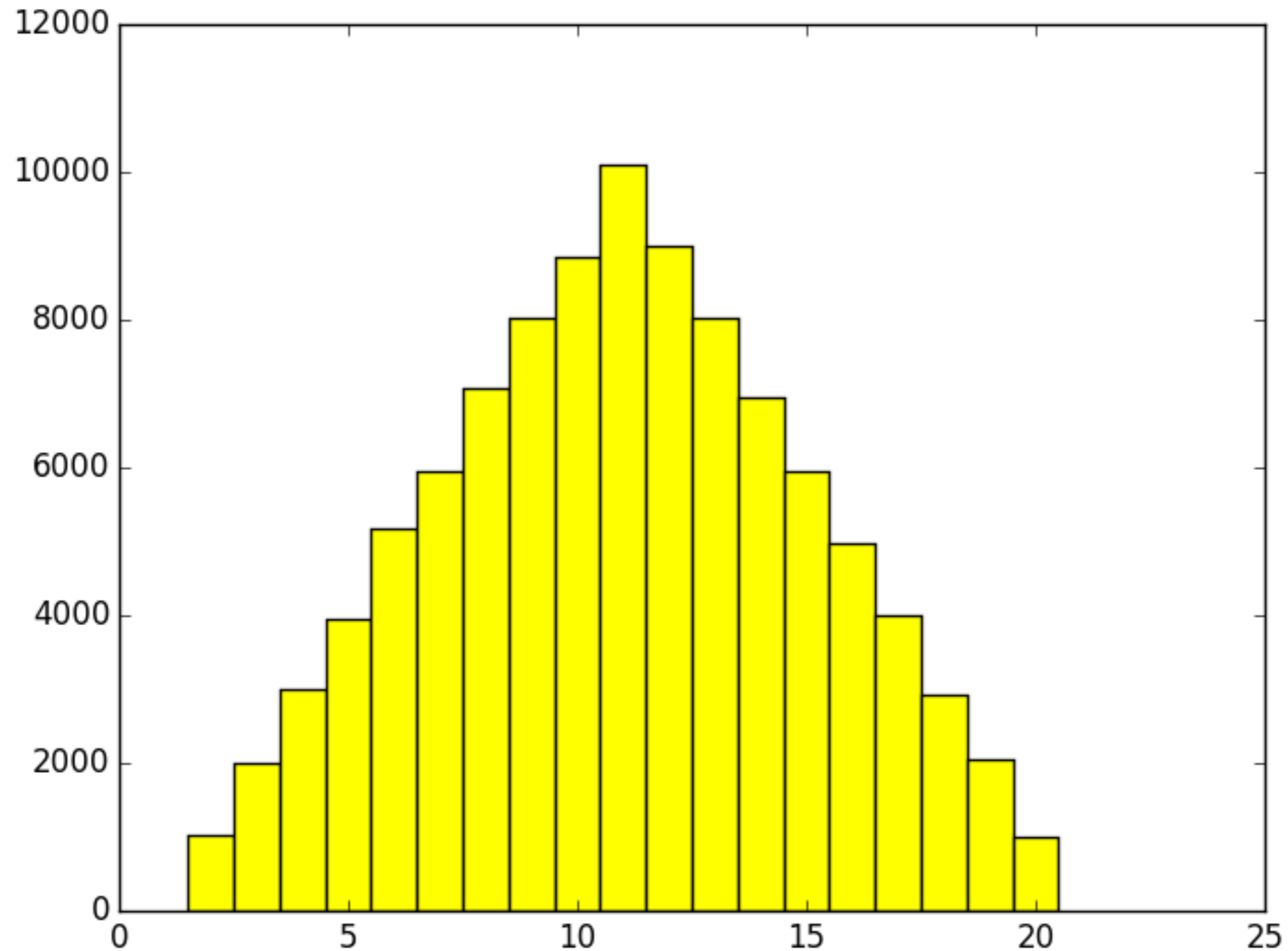
pos = range(2,21)
data = [0] * 19

for i in range(100000):
    a = random.randrange(1,11)
    b = random.randrange(1,11)
    data[a+b-2] += 1

bar1 = plt.bar(pos, data, width=1, \
    color='yellow', align='center')

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

generate random numbers



generate random numbers

```
do 100000 times:  
    a = pick a random number between [1, 10]  
    b = pick a random number between [1, 10]  
    c = pick a random number between [1, 10]  
    d = pick a random number between [1, 10]  
    e = pick a random number between [1, 10]  
    f = a + b + c + d + e  
plot the histogram of f's
```

How to do this in Python?

generate random numbers

```
import random
import matplotlib.pyplot as plt

pos = ?
data = ?

for i in range(100000):
    a = random.randrange(1, 11)
    b = random.randrange(1, 11)
    c = random.randrange(1, 11)
    d = random.randrange(1, 11)
    e = random.randrange(1, 11)
    data[ ? ] += 1

bar1 = plt.bar(pos, data, width=1, \
               color='yellow', align='center')

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

generate random numbers

```
import random
import matplotlib.pyplot as plt

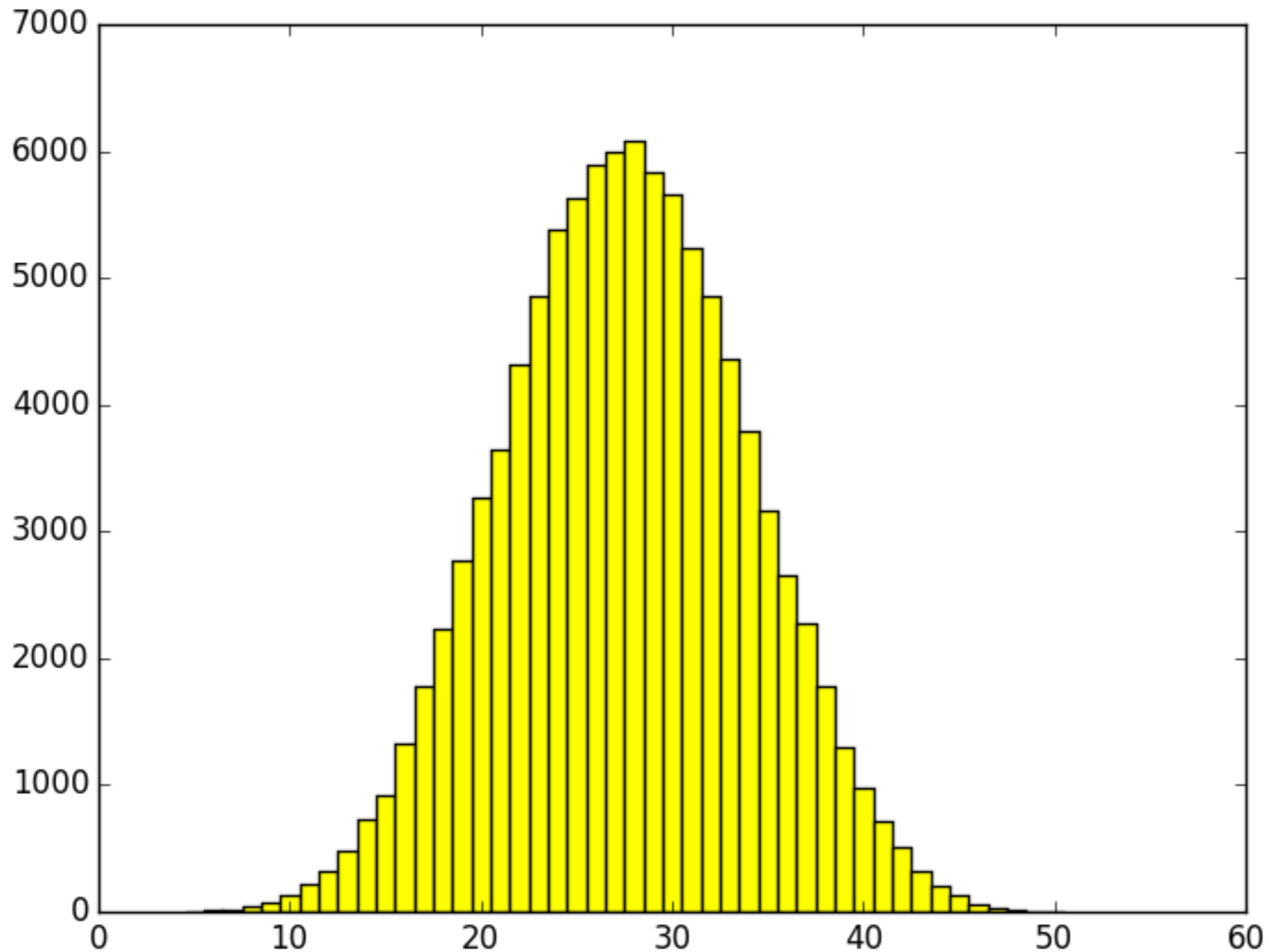
pos = range(5, 51)
data = [0] * 46

for i in range(100000):
    a = random.randrange(1, 11)
    b = random.randrange(1, 11)
    c = random.randrange(1, 11)
    d = random.randrange(1, 11)
    e = random.randrange(1, 11)
    data[a+b+c+d+e-5] += 1

bar1 = plt.bar(pos, data, width=1, \
               color='yellow', align='center')

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


generate random numbers



generate random numbers

- Is this Magic?
- Any Statistics majors in the class?

Introduction to Classes

We know how to define functions

Define a function:

```
def function(value):  
    temp = value * value  
    return temp
```

Call a function:

```
def main():  
    result = function(10)  
    print(result)  
  
main()
```

We also used some other classes and objects

How we used classes in Graphics:

```
myCircle = Circle(Point(0,10), 200)
```

Circle is class
myCircle is object

But we did not define any classes yet.

We will learn how to define classes today.

Define & use a class

How to define
a class:

```
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def printPoint(self):
        print('(' , self.x, ', ' , self.y, ')')
```

How to use
the class:

```
def main():
    p = Point(10, 50)
    p.printPoint()

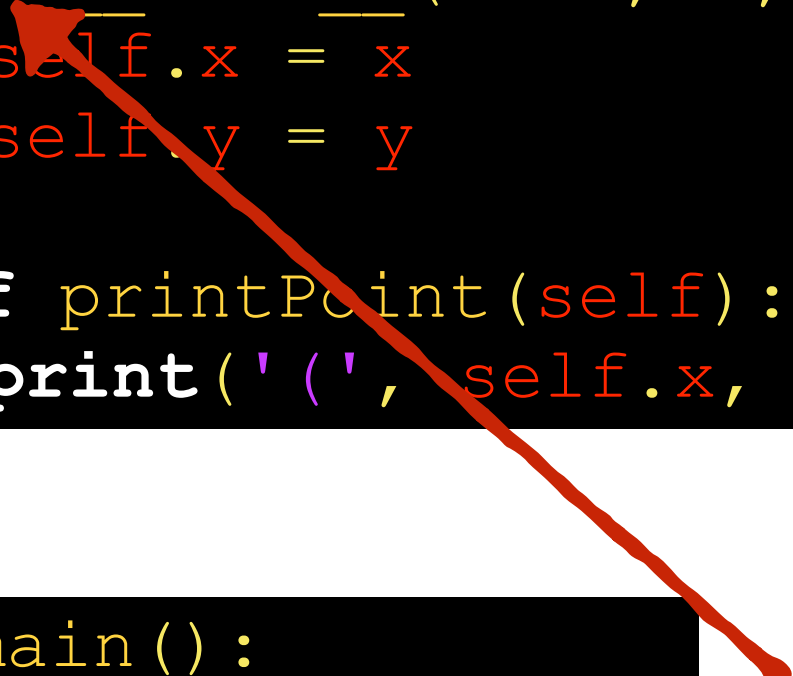
main()
```

Define & use a class

How to define
a class:

```
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def printPoint(self):
        print('(' , self.x, ', ' , self.y, ')')
```



constructor
function

How to use
the class:

```
def main():
    p = Point(10, 50)
    p.printPoint()

main()
```

Define & use a class

How to define
a class:

```
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def printPoint(self):
        print('(' , self.x, ', ' , self.y, ')')
```

constructor
function

How to use
the class:

```
def main():
    p = Point(10, 50)
    p.printPoint()

main()
```

pointer to the
object itself

Define & use a class

other input parameters

How to define a class:

```
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def printPoint(self):
        print('(' , self.x, ', ' , self.y, ')')
```

How to use the class:

```
def main():
    p = Point(10, 50)
    p.printPoint()

main()
```

constructor function

pointer to the object itself

Define & use a class

other input parameters

How to define a class:

```
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def printPoint(self):
        print('(' , self.x, ', ' , self.y, ')')
```

How to use the class:

```
def main():
    p = Point(10, 50)
    p.printPoint()

main()
```

object,
instance

constructor
function

pointer to the
object itself

Define & use a class

How to define
a class:

```
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def printPoint(self):
        print('(' , self.x, ', ' , self.y, ')')
```

How to use
the class:

```
def main():
    p = Point(10, 50)
    p.printPoint()

main()
```

Prints:

```
( 10 , 50 )
```

Questions

Thanks