« Previous (/course/cs357-f15/flow-session/74246/0/)

下一页 » (/course/cs357-f15/flow-session/74246/2/)

跳转至最末 »

1 2 3 (/course/cs357-f15/flow-session/74246/0/) (/course/cs357-f15/flow-session/74246/2/) 4 5 (/course/cs357-f15/flow-session/74246/3/) (/course/cs357-f15/flow-session/74246/4/) 6 7 (/course/cs357-f15/flow-session/74246/5/) 8 (/course/cs357-f15/flow-session/74246/6/) (/course/cs357-f15/flow-session/74246/7/) 9 (/course/cs357-f15/flow-session/74246/8/)

Rolling Dice and Flipping coins

We're going to look at stock prices and when we look at their histories they appear random! There's a lot going into the value of a stock price, but one thing we can do is simulate what happens under some general conditions.

+0.96%
+0.48%
0.00%
-0.48%
10am
11
12
1
2
3
4pm

One easy model is that of dice. Suppose we roll some dice (with values between 1 and 6). And every time we roll, we add or subtract that value to a running total. How should we decide whether to add or subtract? Let's flip a coin!

We're going to need two python modules in this part:

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

`Numpy` gives use some tools to deal with random numbers.

Now the coin flip is going to either be a add (heads or 1) or a subtract (tails or -1), so we can implement this with a `lambda` function as follows:

```
flip = lambda: np.random.choice([1, -1])
```

Try calling `flip()` a bunch of times.

We'll also need a dice roller. How should we do that? use `randint`

```
dice = lambda: np.random.randint(1,6)
```

Try calling `dice()` a bunch of times.