https://campus.datacamp.com/courses/intermediate-python/case-study-hacker-statistics?ex=2

Random Numbers

INTERMEDIATE PYTHON



Data Scientist at DataCamp **Hugo Bowne-Anderson**



К датасамр

https://campus.datacamp.com/courses/intermediate-python/case-study-hacker-statistics?ex=2

К датасамр

INTERMEDIATE PYTHON

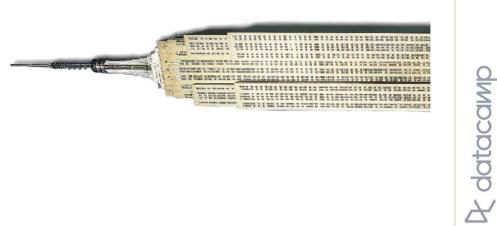
100 X







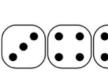


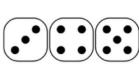


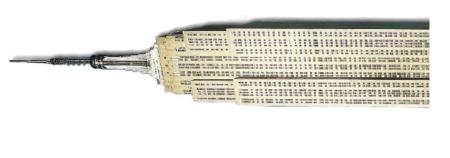
R datacamp









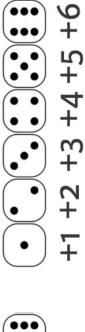














- 100 X

T

- +
- +1 +2 +3 +4 +5 +6

•••

- Can't go below step 0
- 0.1 % chance of falling down the stairs
- Bet: you'll reach step 60

R datacamp

PDF.js viewer

How to solve?

- Analytical
- Simulate the process
- Hacker statistics!



Random generators

import numpy as np
np.random.rand() # Pseu

Pseudo-random numbers

0.9535543896720104

Mathematical formula

np.random.seed(123)

Starting from a seed

np.random.rand()

0.6964691855978616

np.random.rand()

0.28613933495037946



np.random.seed(123)

Random generators

np.random.rand()

0.696469185597861

Same seed: same random numbers!

np.random.rand()

Ensures "reproducibility"

0.28613933495037946



Coin toss

game.py

```
coin = np.random.randint(0,2) # Randomly generate 0 or 1
                                 np.random.seed(123)
import numpy as np
                                                                                                       print(coin)
```

0



Coin toss

game.py

```
coin = np.random.randint(0,2) # Randomly generate 0 or 1
                     np.random.seed(123)
import numpy as np
                                                                                                                                                          print("tails")
                                                                                                               print("heads")
                                                                                      if coin == 0:
                                                               print(coin)
                                                                                                                                        else:
```

ن heads



Let's practice!

Random Walk

INTERMEDIATE PYTHON

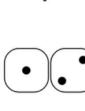


Data Scientist at DataCamp **Hugo Bowne-Anderson**

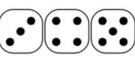


PDF.js viewer

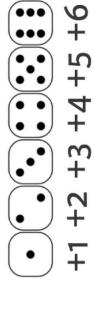
Random Step













Random Walk

100 X











Known in Science

- Path of molecules
- Gambler's financial status



Heads or Tails

headtails.py

```
coin = np.random.randint(0, 2)
                                                                                                                                                    outcomes.append("heads")
                                                                                                                                                                                                       outcomes.append("tails")
                                                                         for x in range(10) :
                         np.random.seed(123)
                                                                                                                            if coin == 0:
import numpy as np
                                                                                                                                                                                                                                    print(outcomes)
                                                 outcomes = []
```

```
'tails', 'heads']
'heads',
                 'heads', 'heads', 'tails',
  'heads',
'tails',
 ['heads',
```



Heads or Tails: Random Walk

headtailsrw.py

```
coin = np.random.randint(0, 2)
                                                                                                                     tails.append(tails[x] + coin)
                                                                       for x in range(10):
                        np.random.seed(123)
import numpy as np
                                                                                                                                                 print(tails)
                                                 tails = [0]
```

[0, 0, 1, 1, 1, 1, 1, 1, 2, 3, 3]



Step to Walk

outcomes

```
'heads', 'heads', 'tails', 'tails', 'heads']
['heads', 'tails', 'heads', 'heads', 'heads'
```

tails

[0, 0, 1, 1, 1, 1, 1, 1, 2, 3, 3]

Let's practice!

Distribution

INTERMEDIATE PYTHON



Hugo Bowne-Anderson Data Scientist at DataCamp



Distribution











Simulate 10,000 times: 10,000 end points

Each random walk has an end point



Distribution!



7

Calculate chances!





•••

Random Walk

headtailsrw.py

```
import numpy as np
np.random.seed(123)
tails = [0]
for x in range(10) :
    coin = np.random.randint(0, 2)
    tails.append(tails[x] + coin)
```



100 runs

distribution.py

```
coin = np.random.randint(0, 2)
                                                                                                                                                                            tails.append(tails[x] + coin)
                                                                                                                                                                                                       final_tails.append(tails[-1])
                                                                                                                        for x in range(10):
                                                                      for x in range(100) :
    tails = [0]
                       np.random.seed(123)
import numpy as np
                                                                                                                                                                                                                                  print(final_tails)
                                                final_tails = []
```

[3, 6, 4, 5, 4, 5, 3, 5, 4, 6, 6, 8, 6, 4, 7, 5, 7, 4, 3, 3, ..., 4]



Histogram, 100 runs

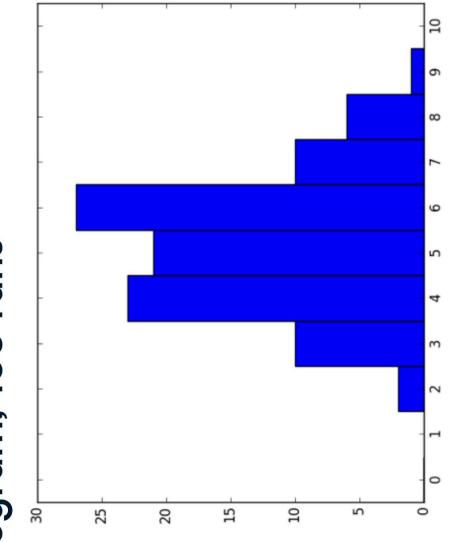
distribution.py

```
coin = np.random.randint(0, 2)
                                                                                                                                                                                                                               tails.append(tails[x] + coin)
                                                                                                                                                                                                                                                            final_tails.append(tails[-1])
                                                                                                                                                                                                                                                                                       plt.hist(final_tails, bins = 10)
                               import matplotlib.pyplot as plt
                                                                                                                                                                      for x in range(10) :
                                                                                                               for x in range(100) :
                                                          np.random.seed(123)
import numpy as np
                                                                                      final_tails = []
                                                                                                                                              tails = [0]
                                                                                                                                                                                                                                                                                                                    plt.show()
```



PDF.js viewer

Histogram, 100 runs





Histogram, 1,000 runs

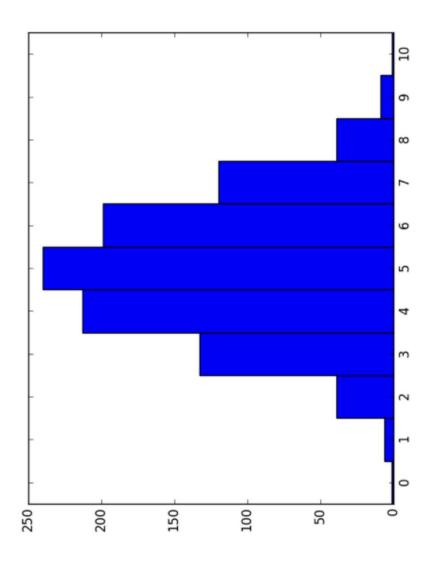
distribution.py

```
coin = np.random.randint(0, 2)
                                                                                                                                                                                                                        tails.append(tails[x] + coin)
                                                                                                                                                                                                                                                  final_tails.append(tails[-1])
                                                                                                                                                                                                                                                                               plt.hist(final_tails, bins = 10)
                            import matplotlib.pyplot as plt
                                                                                                             for x in range(1000) : # <--
                                                                                                                                                               for x in range (10):
                                                      np.random.seed(123)
import numpy as np
                                                                                 final_tails = []
                                                                                                                                       tails = [0]
                                                                                                                                                                                                                                                                                                          plt.show()
```



PDF is viewer

Histogram, 1,000 runs





Histogram, 10,000 runs

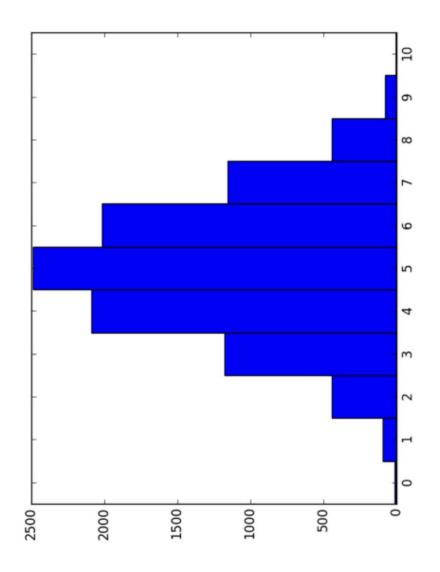
distribution.py

```
coin = np.random.randint(0, 2)
                                                                                                                                                                                                                              tails.append(tails[x] + coin)
                                                                                                                                                                                                                                                        final_tails.append(tails[-1])
                                                                                                                                                                                                                                                                                      plt.hist(final_tails, bins = 10)
                             import matplotlib.pyplot as plt
                                                                                                               for x in range(10000) : # <--
                                                                                                                                                                   for x in range(10) :
                                                        np.random.seed(123)
import numpy as np
                                                                                   final_tails = []
                                                                                                                                          tails = [0]
                                                                                                                                                                                                                                                                                                                  plt.show()
```



PDF is viewer

Histogram, 10,000 runs





Let's practice!