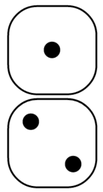
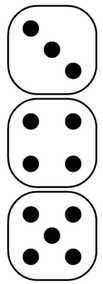
INTERMEDIATE PYTHON FOR DATA SCIENCE

Random Numbers

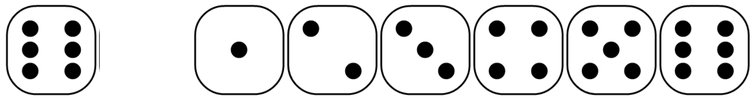
100 x



-1



+1



+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

How to solve?

- Analytical
- Simulate the process
  - Hacker statistics!

Random Generators

```
In [1]: import numpy as np

In [2]: np.random.rand()
Out[2]: 0.9535543896720104

In [3]: np.random.seed(123)

In [4]: np.random.rand()
Out[4]: 0.6964691855978616
In [5]: np.random.rand()
Out[5]: 0.2861393349503794 6

In [6]: np.random.seed(123)

In [7]: np.random.rand()
Out[7]: 0.6964691855978616
In [8]: np.random.rand()
Out[8]: 0.2861393349503794 6
```

Pseudo-random numbers
Mathematical formula
Starting from a seed

Same seed: same random numbers!
Ensures "reproducibility"

Coin Toss

game.py

```
import numpy as np
np.random.seed(123)
coin = np.random.randint(0,2)
print(coin)
```

Randomly generate 0 or 1

```
Output:
0
```

Coin Toss

game.py

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

```
Output:
0
heads
```

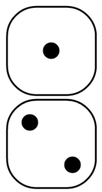
INTERMEDIATE PYTHON FOR DATA SCIENCE

Let’s practice!

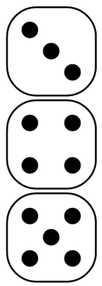
INTERMEDIATE PYTHON FOR DATA SCIENCE

Random Walk

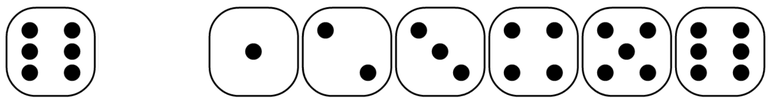
Random Step



-1



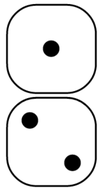
+1



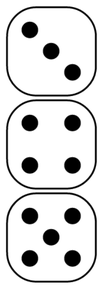
+1 +2 +3 +4 +5 +6

Random Walk

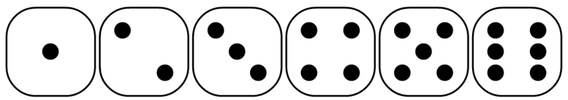
100 x



-1



+1



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

Known in Science

- Path of molecules
- Gambler's financial status

Heads or Tails

headtails.py

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

0 : heads
1 : tails

Output:

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

Heads or Tails: 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)

print(tails)
```

Output:

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

Step to Walk

outcomes

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

tails

```
Output:
[0, 0, 1, 1, 1, 1, 1, 1, 2, 3, 3]
```

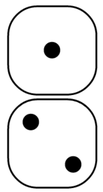
INTERMEDIATE PYTHON FOR DATA SCIENCE

Let’s practice!

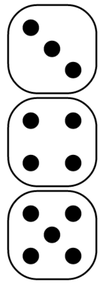
INTERMEDIATE PYTHON FOR DATA SCIENCE

Distribution

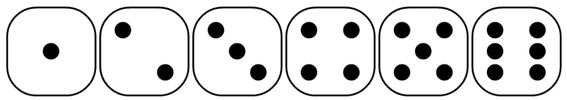
100 x



-1



+1



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

Each random walk has an end point

Simulate 10,000 times: 10,000 end points

Distribution!

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

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

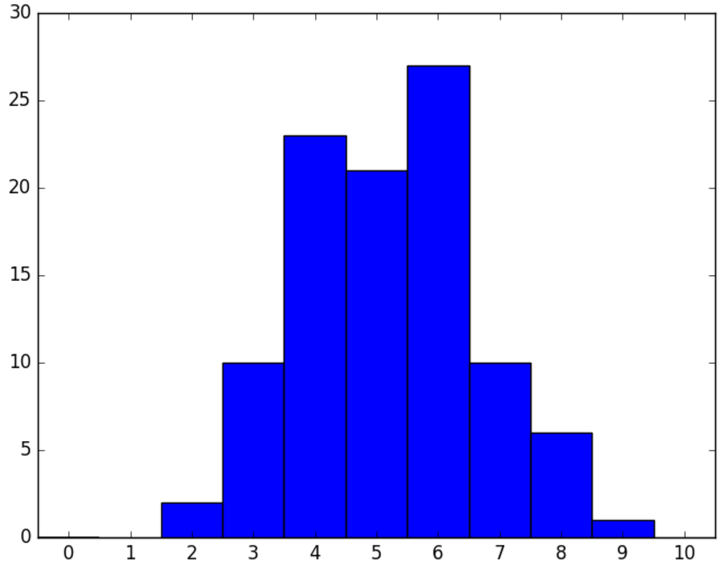
Output:

```
[3, 6, 4, 5, 4, 5, 3, 5, 4, 6, 6, 8, 6, 4, 7, 5, 7,
4, 3, 3, 4, 5, 8, 5, 6, 5, 7, 6, 4, 5, 8, 5, 8, 4,
6, 6, 3, 4, 5, 4, 7, 8, 9, 4, 3, 4, 5, 6, 4, 2, 6,
6, 5, 7, 5, 4, 5, 5, 6, 7, 6, 6, 6, 3, ..., 7]
```

Histogram, 100 runs

distribution.py

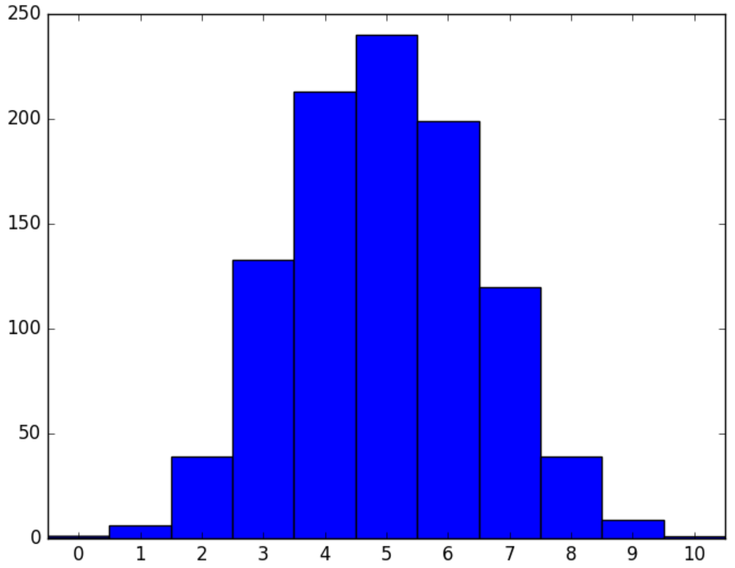
```
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(123)
final_tails = []
for x in range(100) :
    tails = [0]
    for x in range(10) :
        coin = np.random.randint(0,2)
        tails.append(tails[x] + coin)
    final_tails.append(tails[-1])
plt.hist(final_tails, bins = 10)
plt.show()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



Histogram, 1.000 runs

distribution.py

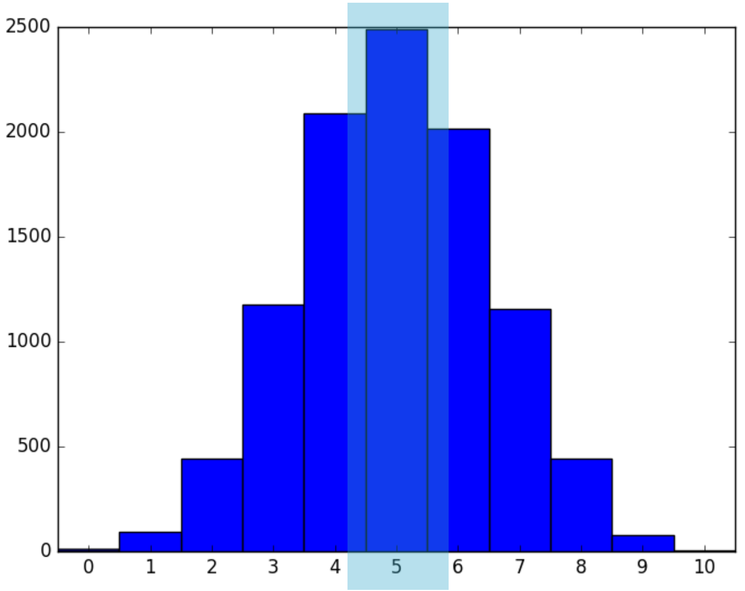
```
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(123)
final_tails = []
for x in range(1000) :
    tails = [0]
    for x in range(10) :
        coin = np.random.randint(0,2)
        tails.append(tails[x] + coin)
    final_tails.append(tails[-1])
plt.hist(final_tails, bins = 10)
plt.show()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



Histogram, 10.000 runs

distribution.py

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



INTERMEDIATE PYTHON FOR DATA SCIENCE

Let’s practice!