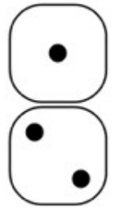
Random Numbers

INTERMEDIATE PYTHON

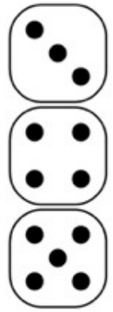
Hugo Bowne-Anderson
Data Scientist at DataCamp

datacamp

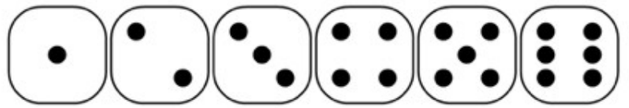
100 x



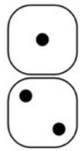
-1

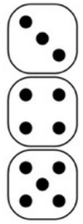


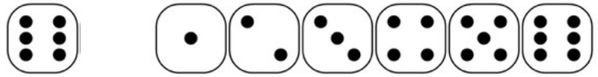
+1



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

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

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

Random generators

```
np.random.seed(123)
np.random.rand()
```

```
0.696469185597861    # Same seed: same random numbers!
```

```
np.random.rand()     # Ensures "reproducibility"
```

```
0.2861393349503794 6
```

Coin toss

game.py

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

```
0
```

Coin toss

game.py

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

```
0
heads
```

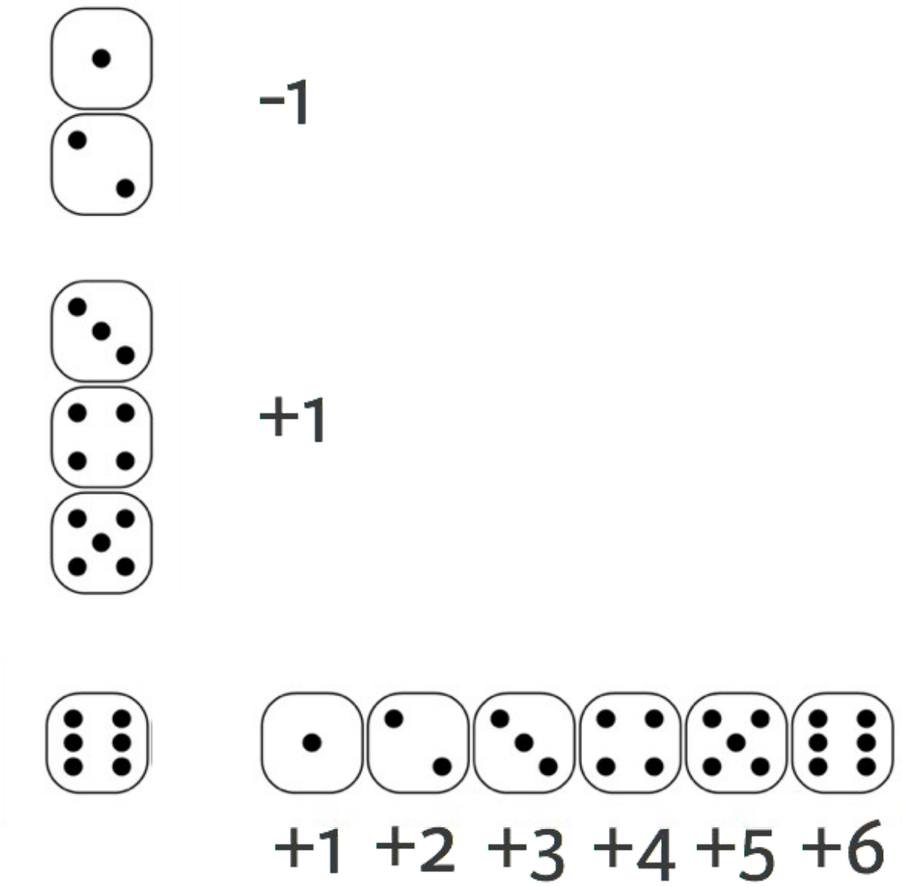
Random Walk

INTERMEDIATE PYTHON

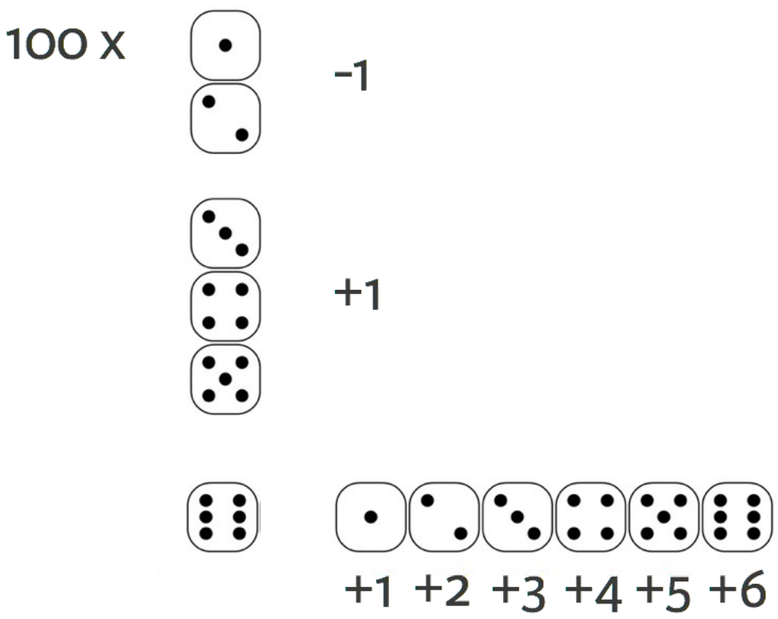
Hugo Bowne-Anderson
Data Scientist at DataCamp

datacamp

Random Step



Random Walk



Known in Science

- Path of molecules
- Gambler's nancial 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)
```

```
['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)
```

```
[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
```

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

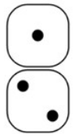
Distribution

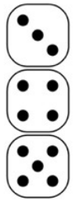
INTERMEDIATE PYTHON

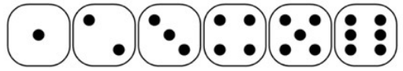
Hugo Bowne-Anderson
Data Scientist at DataCamp

datacamp

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)
```

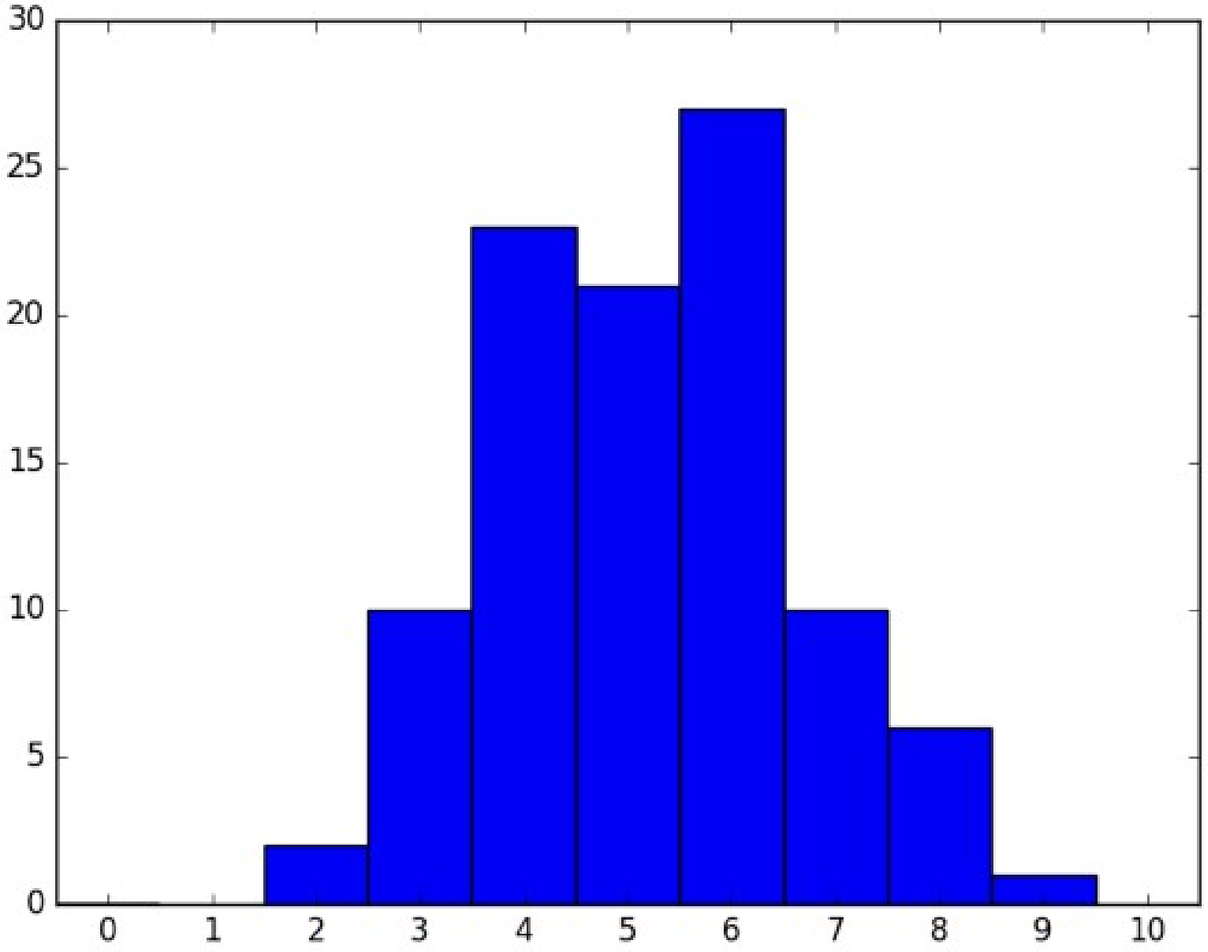
```
[3, 6, 4, 5, 4, 5, 3, 5, 4, 6, 6, 8, 6, 4, 7, 5, 7, 4, 3, 3, ..., 4]
```

Histogram, 100 runs

distribution.py

```
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)
plot.show()
```

Histogram, 100 runs

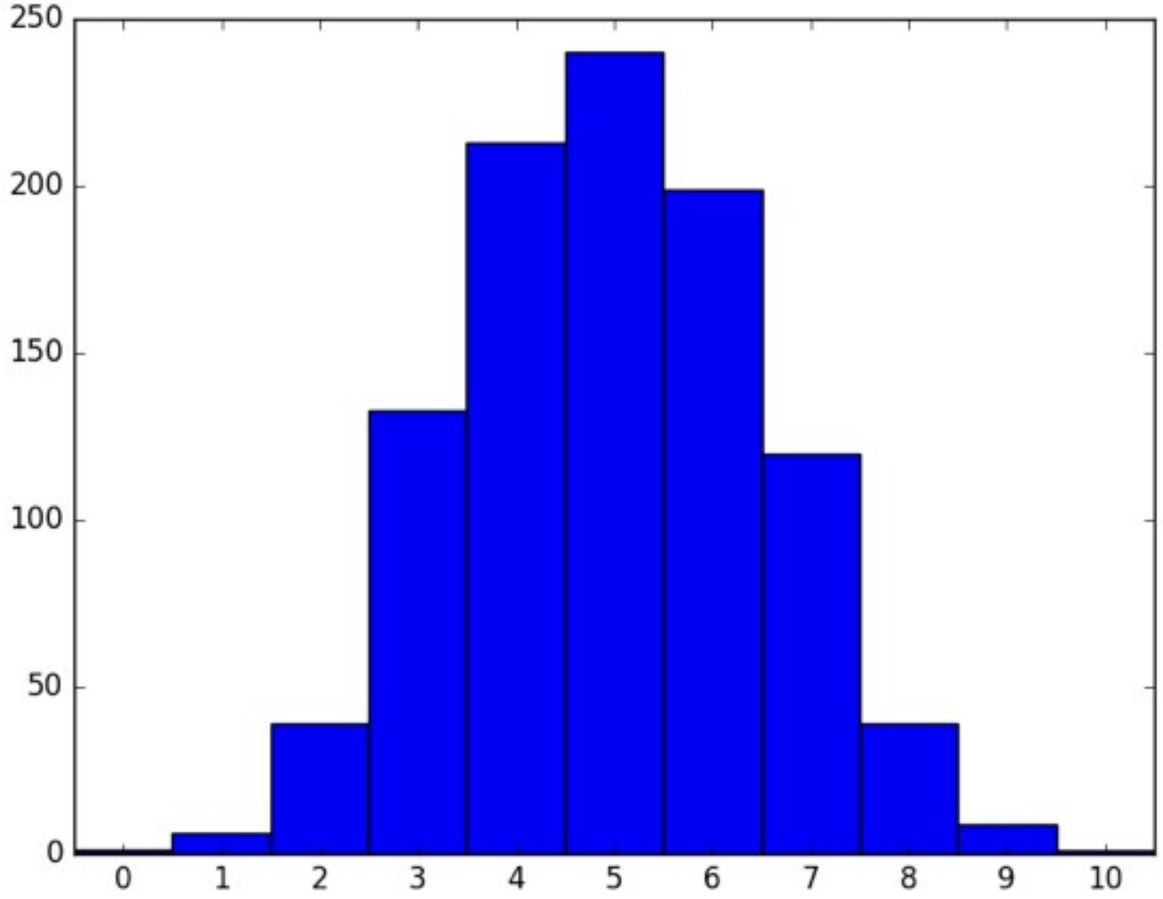


Histogram, 1,000 runs

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)
plot.show()
```

Histogram, 1,000 runs



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)
plot.show()
```

Histogram, 10,000 runs