

ENGG1003 - Thursday Week 4

Using random numbers, and reading from spreadsheets

Steve Weller & Sarah Johnson

University of Newcastle

18 March, 2021

Last compiled: March 18, 2021 11:54am +11:00

Lecture overview

- 1 Using random numbers
- 2 Reading from spreadsheets

1) Using random numbers

Recap: generating random numbers

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

In [2]: np.random.randint(1, 6, 4)      # ...4 integers from [1, 6)
Out[2]: array([1, 3, 5, 3])

In [3]: np.random.random(4)            # ...4 floats from [0, 1)
Out[3]: array([ 0.79183276,  0.01398365,  0.04982849,  0.11630963])

In [4]: np.random.uniform(10, 20, 4)   # ...4 floats from [10, 20)
Out[4]: array([ 10.95846078,  17.3971301 ,  19.73964488,  18.14332234])
```

Using numpy library:

- random integers
- random floats from $[0, 1)$
- random floats from $[a, b]$

Random integers: simulating coin toss

Example 1

Simulate the toss of a coin N times as follows:

- generate a length- N array of randomly chosen 0s and 1s
 - ▶ 0 = heads, 1 = tails
 - ▶ equally likely heads and tails ie: fair coin
- display *expected* number of heads observed
- display *actual* number of heads observed
- test/debug with $N = 100$, then $N = 100,000$

Coin toss simulation

```
1 import numpy as np
2
3 # generate random array of 0s and 1s
4 # 0==heads & 1==tails
5 # N integers from [0,2) ie: 0 or 1
6 N = 100000
7 x = np.random.randint(0, 2, N)
8 print(x)
9
10 headCnt = 0;
11 for i in range(0,N,1):
12     if x[i]==0:
13         headCnt += 1
14
15 print('Expected number of heads: {}'.format(N/2))
16 print('Observed number of heads: {}'.format(headCnt))
```

- Live demo of headsTails.py

Random floats: engineering tolerance

Example 2

- simulate values in a range
- need engineering application—part manufactured within a tolerance, calculate fraction outside range

Engineering tolerance simulation

```
1 import numpy as np
2
3 # generate random array of N floats in range [17,19]
4 N = 10000
5 x = np.random.uniform(17,19,N)
6 tolLow = 17.25
7 tolHigh = 18.75
8 #print(x)
9
10 goodCnt = 0;
11 for i in range(0,N,1):
12     if tolLow <= x[i] <= tolHigh:
13         goodCnt += 1
14
15 print('Percentage of parts within tolerance: {}'.format(100*goodCnt/N))
```

- Live demo of engTolerance.py

Random floats: simulate dartboard

Example 3

- values in circle
- plot red inside, blue outside

Dartboard simulation

- Python code for dartboard
- dartboard.py
- live demo

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

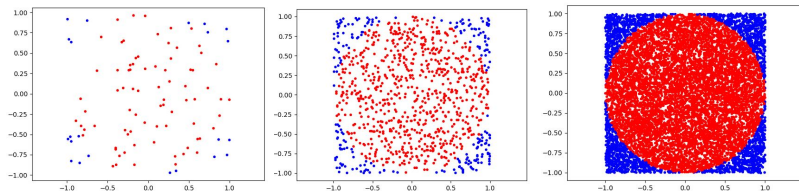
# generate random array of (x,y) pairs covering
# square with edge length 2
N = 10000
x = np.random.uniform(-1, 1, N)      # N floats from [-1,1]
y = np.random.uniform(-1, 1, N)      # N floats from [-1,1]

insideCnt = 0;
for i in range(0,N,1):
    if x[i]**2 + y[i]**2 <= 1:
        plt.plot(x[i],y[i],'r.')
    else:
        plt.plot(x[i],y[i],'b.')

plt.axis('equal')
plt.show()
```

Dartboard simulation

- dartboard red/blue simulation output plots
- $N=100, 1000, 10K$



Random floats: estimate π

Example 4

- modify previous example to count points inside circle, hence. . .
- estimate π

Random floats: estimate π

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

# generate random array of (x,y) pairs covering
# square with edge length 2
N = 10000
x = np.random.uniform(-1, 1, N)    # N floats from [-1,1]
y = np.random.uniform(-1, 1, N)    # N floats from [-1,1]

insideCnt = 0;
for i in range(0,N,1):
    if x[i]**2 + y[i]**2 <= 1:
        plt.plot(x[i],y[i], 'r.')
        insideCnt += 1
    else:
        plt.plot(x[i],y[i], 'b.')

R = insideCnt/N
print('Estimate of pi: {}'.format(4*R))

plt.axis('equal')
plt.show()
```

Live demo

2) Reading from spreadsheets

- Lets see our code to calculate the height of a ball (from week 1) as a function

```
1 # Function Definition
2 def ball_height(v0, t):           # Function header
3     g = 9.81                      # Function body
4     y = v0*t - 0.5*g*t**2
5     return y                     # Return statement
```