In [4]:
```
import numpy as np

def f(x):
    f = x
    return x

N = 1000
x_i = np.random.uniform(size=N)
print("Shape of x_i is", x_i.shape)
f_i = f(x_i)
I_N = np.sum(f_i)/N
print("Monte Carlo Approximation is %f" %I_N)
```

```
Shape of x_i is (1000,)
Monte Carlo Approximation is 0.507650
```

In [5]:
```
import numpy as np

def f(x):
    f = x
    return x

N = 1000
x_i = np.random.uniform(size=N)
print("Shape of x_i is", x_i.shape)
f_i = f(x_i)
I_N = np.sum(f_i)/N
print("Monte Carlo Approximation is %f" %I_N)
```

```
Shape of x_i is (1000,)
Monte Carlo Approximation is 0.500072
```

In [6]:
```
import numpy as np

def f(x):
    f = x
    return x

N = 1000
x_i = np.random.uniform(size=N)
print("Shape of x_i is", x_i.shape)
f_i = f(x_i)
I_N = np.sum(f_i)/N
print("Monte Carlo Approximation is %f" %I_N)
```

```
Shape of x_i is (1000,)
Monte Carlo Approximation is 0.508398
```

In [7]:
```
import numpy as np

def f(x):
    f = x
    return x

N = 1000
x_i = np.random.uniform(size=N)
print("Shape of x_i is", x_i.shape)
f_i = f(x_i)
I_N = np.sum(f_i)/N
print("Monte Carlo Approximation is %f" %I_N)
```

```
Shape of x_i is (1000,)
Monte Carlo Approximation is 0.496578
```

In [8]:
```
import numpy as np

def f(x):
    f = x
    return x

N = 1000
x_i = np.random.uniform(size=N)
print("Shape of x_i is", x_i.shape)
f_i = f(x_i)
I_N = np.sum(f_i)/N
print("Monte Carlo Approximation is %f" %I_N)
```

```
Shape of x_i is (1000,)
Monte Carlo Approximation is 0.499426
```

The value of Monte Carlo Approximation changes everytime on running the code since x_i takes random values.

Increasing the value of N

In [12]:
```
import numpy as np

def f(x):
    f = x
    return x

N = 10000000
x_i = np.random.uniform(size=N)
print("Shape of x_i is", x_i.shape)
f_i = f(x_i)
I_N = np.sum(f_i)/N
print("Monte Carlo Approximation is %f" %I_N)
```

```
Shape of x_i is (10000000,)
Monte Carlo Approximation is 0.499916
```

In [13]:
```
import numpy as np

def f(x):
    f = x
    return x

N = 1000000000
x_i = np.random.uniform(size=N)
print("Shape of x_i is", x_i.shape)
f_i = f(x_i)
I_N = np.sum(f_i)/N
print("Monte Carlo Approximation is %f" %I_N)
```

```
Shape of x_i is (1000000000,)
Monte Carlo Approximation is 0.500007
```

On increasing the value of N, Monte Carlo Estimate becomes close to 0.5

In [ ]: