

```
In [29]: import numpy as np
import random
```

```
In [37]: #1.a
def rolldice(n):
    a=0

    for i in range (1,n+1):
        y=list(range(1,7))
        x=[]
        b=0
        while(x!=y):
            b+=1
            z=random.randint(1,6)
            if z not in x:
                x.append(z)
                x.sort()
        a=a+b
    print(a/n)
```

```
In [39]: rolldice(100000)

14.73902
```

```
In [17]: random.randint(1,6)
```

```
Out[17]: 6
```

```
In [42]: #1.b
def rolldice3(n):
    a=0
    for i in range (1,n+1):
        x=[]
        y=list(range(3,19))
        b=0
        while(x!=y):
            b+=1
            z=random.randint(1,6)+random.randint(1,6)+random.randint(1,6)
            if z not in x:
                x.append(z)
                x.sort()
        a=a+b
    print(a/n)
```

```
In [44]: rolldice3(100000)

338.84801
```

```
In [45]: #3.a
def unifiid(n):
    a=0
    for i in range(1,n+1):
        b=0
        c=0
        while(b<1):
            b=np.random.uniform(0,1)
            c+=1
        a=a+c
    print(a/n)
```

```
In [46]: unifiid(100)
```

2.7

```
In [47]: unifiid(1000)
```

2.705

```
In [48]: unifiid(10000)
```

2.7261

```
In [69]: #3.b
def lamba():
    samples = [100, 1000, 10000]
    lambdaa = [1,2]
    for y in range(0, len(lambdaa)):
        for x in range(0, len(samples)):
            M = 0
            for j in range(1,samples[x]):
                total = 1
                n = np.array(np.random.uniform(0,1,samples[x]))
                for i in range(1, len(n)):
                    total = total*n[i]
                    if (total <= np.exp(-1*lambdaa[y])):
                        M = M + i - 1
                        break
            print("size is",samples[x], " lambda is", lambdaa[y], "answer is ", M)
```

```
In [70]: lamba()
```

```
size is 100  lambda is 1 answer is  1.03
size is 1000  lambda is 1 answer is  0.968
size is 10000  lambda is 1 answer is  0.9893
size is 100  lambda is 2 answer is  2.0
size is 1000  lambda is 2 answer is  1.951
size is 10000  lambda is 2 answer is  2.0067
```

In [81]: #4.c

```

p = np.array([0, 0.05, 0.1, 0.15, 0.1, 0.4, 0.2])
N = np.zeros(10000)
q = np.zeros(7)
Unif = np.random.uniform(0, 1, 10000)
for i in range(1,7):
    q[i] = p[i] + q[i-1]
    for n in range(10000):
        for i in range(7):
            if Unif[n] <= q[i]:
                X = i
                n = n + 1
                break
            else:
                i = i + 1
                N[n] = i
print("Expected value is ", np.mean(N))

```

Expected value is 4.2876

In [84]: #4.e

```

prob = np.array([0, 0.4, 0.2, 0.15, 0.1, 0.1, 0.05])
sequential=[0,5,6,3,2,4,1]
N = np.zeros(10000)
q = np.zeros(7)
U = np.random.uniform(0, 1, 10000)
for i in range(1,7):
    q[i] = prob[i] + q[i-1]
for n in range(10000):
    for i in range(7):
        if U[n] <= q[i]:
            X = sequential[i]
            n = n + 1
            break
        else:
            i = i + 1
            N[n] = i
print("Expected value is ", np.mean(N))

```

Expected value is 2.4482

In []: