

Lab05 Solutions

PSTAT 5A, compiled by Ethan

April 30, 2023

```
[1]: import numpy.random as npr
```

1 Task 1

```
[2]: npr.randint(1, 7, 5)
```

```
[2]: array([4, 4, 3, 5, 4])
```

Note that we need to specify 7 as the second argument as specifying 6 would only generate numbers from the set $\{1, 2, 3, 4, 5\}$.

```
[3]: type(npr.randint(1, 7, 5))
```

```
[3]: numpy.ndarray
```

It seems as though a call to `randint()` is stored as an array. (This is a different type of array than the one we dealt with in Lab02; specifically, this is an array as defined in the `numpy` module. You don't need to worry about the distinction between a `datascience` array and a `numpy` array right now.)

2 Task 2

Part (a)

```
[6]: npr.randint(1, 7)
```

```
[6]: 5
```

Indeed, with any luck you should **not** have observed the same number 3 times (though, there is a small chance that you would have.)

Part (b)

```
[9]: npr.seed(15)
     npr.randint(1, 7)
```

```
[9]: 1
```

Now, with the seed set, we *do* obtain the same number each time.

Part (c)

With any luck, your neighbors should have also got the number 1 when running the code cell in part (b).

```
[10]: import scipy.stats as sps
```

3 Task 3

Part (a)

```
[11]: sps.binom.pmf(20, 143, 0.153)
```

```
[11]: 0.08687059451566365
```

Part (b)

```
[12]: sps.binom.pmf(40, 143, 0.153)
```

```
[12]: 4.347048512074074e-05
```

The notation `e-05` is Python's syntax for scientific notation; i.e. the answer is approximately 4.347×10^{-5} .

4 Task 4

Part (a)

```
[13]: sps.norm.cdf(2, 3, 0.5)
```

```
[13]: 0.022750131948179195
```

Part (b)

```
[14]: 1 - sps.norm.cdf(1, -2, 1)
```

```
[14]: 0.0013498980316301035
```

Part (c)

```
[15]: sps.norm.cdf(1, 0, 1) - sps.norm.cdf(-1, 0, 1)
```

```
[15]: 0.6826894921370859
```

5 Task 5

Part (a)

```
[16]: x = sps.uniform.rvs(2, 10, 100)
      x[0:9]
```

```
[16]: array([10.13580447,  7.91354996,  3.10986325, 10.18611773, 11.26642824,
            2.24953032, 10.22298097,  9.19925099,  4.12108852])
```

Part (b)

```
[17]: y = sps.norm.rvs(98.2, 2.4, 150)
      y[0:9]
```

```
[17]: array([ 99.63850546, 101.25385389,  97.3515234 ,  94.81041896,
            98.77398478,  99.4530164 , 101.01055759, 100.37090096,
            98.78261138])
```