MATH 318 Homework 4

Joshua Lim (26928218)

February 10, 2023

Problem 1

We can first normalize the uniform distribution here:

$$P(X = a) = \begin{cases} \frac{1}{5} & 3 \leq x \leq 8 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Part A

$$P(T \leq a) = P(\frac{1}{2}X^2 \leq a) = P(X \leq \sqrt{2a}) = \int_{-\infty}^{\sqrt{2a}} P(X = s)\, ds$$

$$P(T \leq a) = \begin{cases} \frac{1}{5}(\sqrt{2a} - 3) & \frac{9}{2} \leq x \leq 32 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Part B

We can find the PDF of T by taking the derivative of the CDF:

$$pr(t) = \begin{cases} \frac{1}{5}\frac{1}{\sqrt{2t}} & \frac{9}{2} \leq x \leq 32 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Part C

$$\mu = \int_{-\infty}^{\infty} T(t) \cdot p_T(t)\, dt \ = \int_{\frac{9}{2}}^{32} t \frac{1}{5} \frac{1}{\sqrt{2t}}\, dt \ = \frac{1}{5\sqrt{2}} \int_{\frac{9}{2}}^{32} \sqrt{t}\, dt \ = \frac{2}{15\sqrt{2}}(32^{\frac{3}{2}} - \frac{9^{\frac{3}{2}}}{2}) = \frac{97}{6}$$

Problem 2

Part A

First, we can say that the system is memory-less and we have a desired event with a trial taking place at any point. The first coupon is always not in the collection and any coupon after that has a $\frac{n+i-1}{n}$ chance of not already being in the collection. So, we can model this scenario as a sum of Geometric RVs with one RV for each coupon accumulated.

A new one added to the collection changes the chance rate of a unique coupon being added. The probability of a coupon being added is:

$$X_i = Geo(\frac{n+i-1}{n})$$

$$T = \sum_{i=1}^{n} X_i$$

$$P(X_i = m) = (1 - \frac{n+i-1}{n})^{m-1}(\frac{n+i-1}{n})$$

Part B

For a Geometric RV, the expected value is $\frac{1}{p}$ as we covered in class lecture notes. Thus, we can write the expected value of X_i as:

$$< X_i >= \frac{n}{n+i-1}$$

Then it follows that the expected value of T is the sum of the expected values of each X_i:

$$< T >=< \sum_{i}^{n} X_i >= \sum_{i}^{n} < X_i >= \sum_{i=1}^{n} \frac{n}{n+i-1} = n(\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + ... + \frac{1}{n})$$

To conclude, this is approximately equal to $nlog(n)$ days.

Problem 3

Part A

We notice the events are independent so $P(x, y) = P_T(x)P_B(y)$:

$$P_T(T = x) = 6e^{-6x}$$

$$P_B(B = y) = 4e^{-4y}$$

$$P(x, y) = 24e^{-6x-4y}$$

Part B

We have a 2D space for probability distribution which we cut with the line y = x and we want to find the area under the curve to the left of the line. We can use the following formula to find the area under the curve:

$$\int_0^\infty \int_{y=x}^\infty f(x,y)\,dxdy \;=\int_0^\infty -6e^{-6x-4y}\,dx \;=\int_0^\infty -6(e^{-10x})dx = -\frac{3}{5}e^{-10x}\Big|_0^\infty = \frac{3}{5}$$

$$(T \le B) = \frac{3}{5}$$

Part C

It is easier to find the complement of the problem which avoids some of the intricacies of the Min function. Let X be the random variable for the time it takes to catch the first vehicle $P(X \le t) = 1 - P(X > t)$.

$$P(X > t) = P(T > t, B > t) = P(T > t)P(B > t) = e^{-6t}e^{-4t} = e^{-10t}$$
$$P(X \le t) = 1 - e^{-10t}$$
$$P_x(t) = 10e^{-10t} \Rightarrow X \sim exp(10)$$

We shall call X an exponential random variable with parameter 10.

Problem 4

We begin by solving for the case of X^2 by looking at the CDF:

$$F_{X^2} = P(X^2 \le t) = P(|X| \le \sqrt{t}) = 2\int_0^{\sqrt{t}} \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}\,dx$$

Let's do u-sub, using $u = x^2$ and $\frac{du}{\sqrt{2u}} = dx$:

$$F_{X^2} = \int_0^t \frac{1}{\sqrt{\pi u}}e^{-\frac{u}{2}}\,du$$

Taking the derivative gives us the pdf as well:

$$\frac{d}{dt}F_{X^2} = \frac{1}{\sqrt{\pi t}}e^{-\frac{t}{2}}$$

We can use the fact that X^2 and Y^2 are independent to get the distribution of $R^2 = X^2 + Y^2$:

$$F_z z = P(X^2 + Y^2 \le R) = \iint_{disk} (\frac{1}{\sqrt{\pi}})^2 \frac{1}{\sqrt{st}}e^{\frac{-s-t}{2}}\,ds\,dt$$

Subbing in with $x = \sqrt{s}$ and $y = \sqrt{t}$:

$$F_z z = \frac{1}{2\pi} \iint_{disk} e^{\frac{-x^2-y^2}{2}} \, dx \, dy$$

$$F_z z = \frac{1}{2\pi} \int_0^{2\pi} \int_0^R e^{\frac{-r^2}{2}} r \, dr \, d\theta \ = 1 - e^{\frac{-R^2}{2}}$$

Then the distribution $f_Z(z)$ is:

$$f_R(r) = re^{\frac{-r^2}{2}}$$

This distribution is commonly known as a *Rayleigh* distribution.

Problem 5

Part A

Since all points on the surface are of equal probability, we can compare the areas. The area of the unit disc is π and the area of a small piece dr that is cut is $2\pi r dr$.

Thus, the PDF is $f(r) = \frac{2\pi r}{\pi} = 2r$ and the CDF is $F_r(r) = \int_0^r 2s \, ds \ = r^2$.

Part B

The expected distance from the origin is:

$$E[r] = \int_0^1 r 2r dr = \frac{2}{3}$$

Part C

The PDF for the combination of both is equal at all points in space on the disk so it is uniform. The PDF is:

$$f(x, y) = \frac{1}{\pi} \qquad\qquad (x, y) \in Disk$$

We can get the marginal PDF of X and Y by integrating out the other variable. So the marginal PDF of X is:

$$f_x(x) = \int_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}} \frac{1}{\pi} = \frac{2\sqrt{1-x^2}}{\pi}$$

Part D

We get independence by satisfying one of many conditions, the one I show is not true is:

$$f(x,y) = f_x(x)f_y(y)$$
$$f_x(x)f_y(y) = \frac{2\sqrt{1-x^2}}{\pi}\frac{2\sqrt{1-y^2}}{\pi} = \frac{4}{\pi^2}(1-x^2-y^2+x^2y^2) = \frac{4x^2y^2}{\pi^2} \neq \frac{1}{\pi}$$

This is **not independent**.

Part E

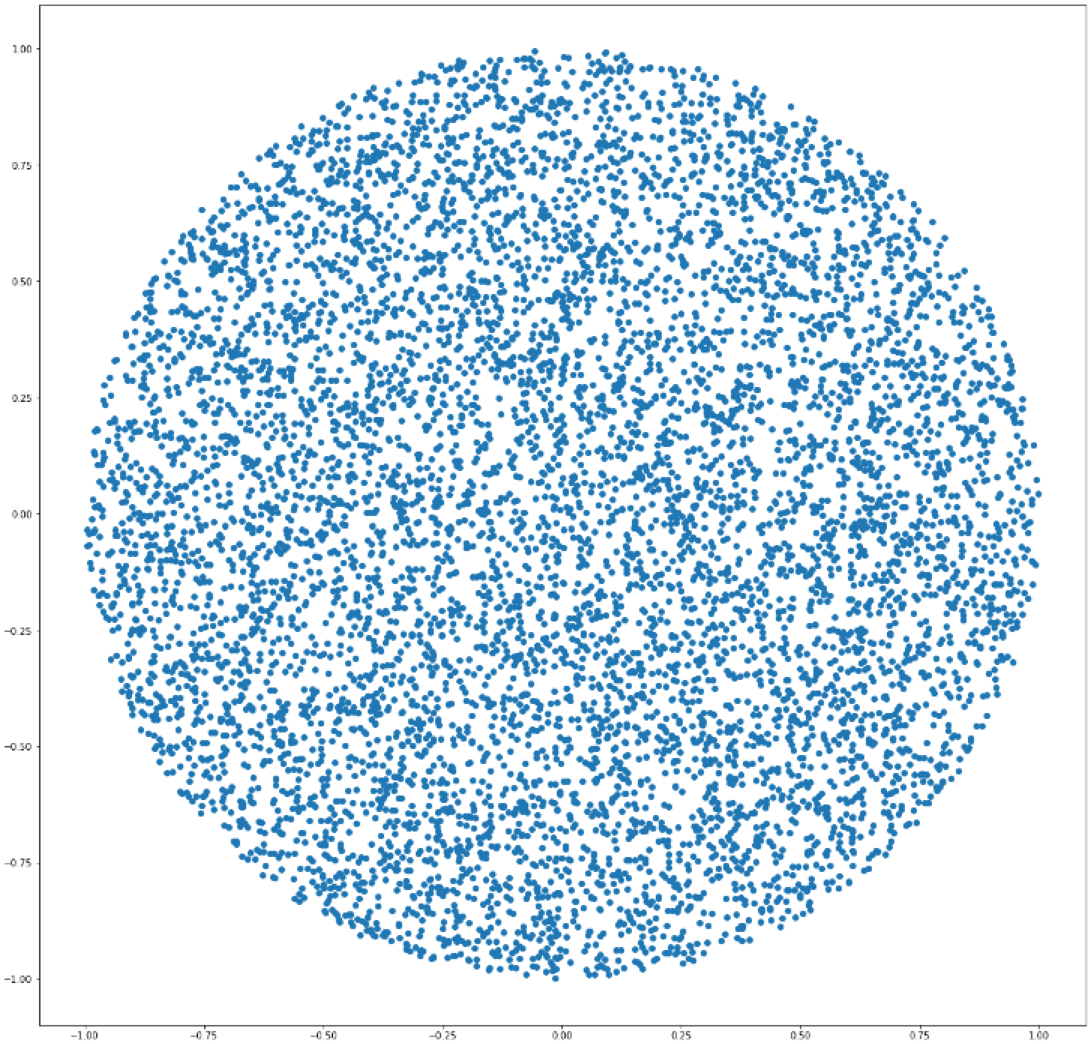
$$\frac{area(A)}{\pi} = \iint_A \frac{r}{\pi} dr d\theta$$

Uniform random points on the unit disc (square method):

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

def random_points_on_disc(n):
    x = np.random.uniform(-1, 1, n)
    y = np.random.uniform(-1, 1, n)
    d = x**2 + y**2
    inside = d <= 1
    return x[inside], y[inside]

x, y = random_points_on_disc(10000)
plt.figure(figsize=(20, 20))
plt.scatter(x, y)
plt.show()
```



Part F

Uniform random points on the unit disc (polar method):

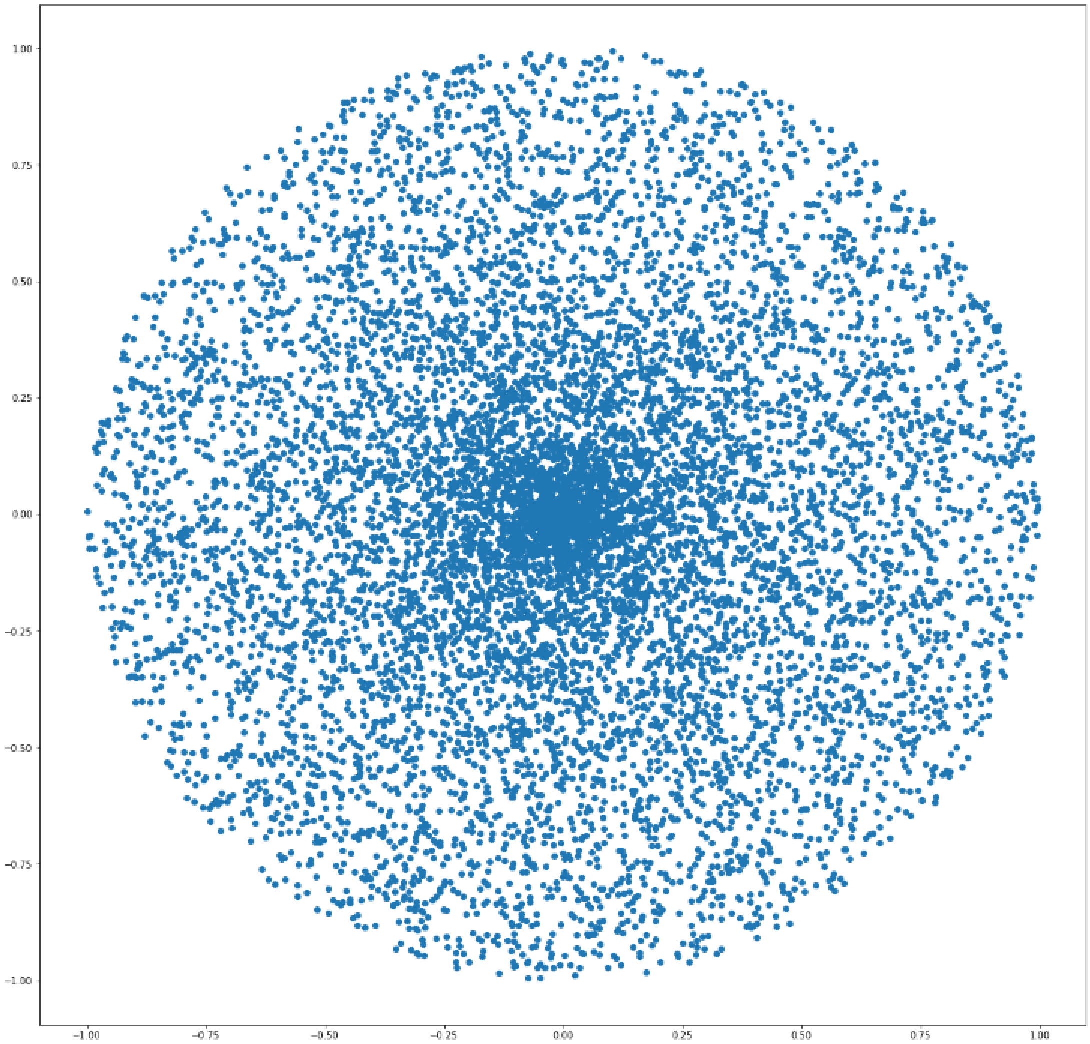
```
import matplotlib.pyplot as plt
import numpy as np

def generate_polar_coordinates(n):
    R = np.random.uniform(0, 1, n)
    Theta = np.random.uniform(0, 2 * np.pi, n)
    return R, Theta

def plot_polar_coordinates(R, Theta):
    x = R * np.cos(Theta)
    y = R * np.sin(Theta)
```

```
    plt.figure(figsize=(20, 20))
    plt.scatter(x, y)
    plt.show()

n = 10000
R, Theta = generate_polar_coordinates(n)
plot_polar_coordinates(R, Theta)
```



The polar method's scatter plot is denser in the center of the disk while the square method's scatter plot is mostly uniform all throughout the disk.

Part G

$$f(r, \theta) = \frac{r}{\pi}$$

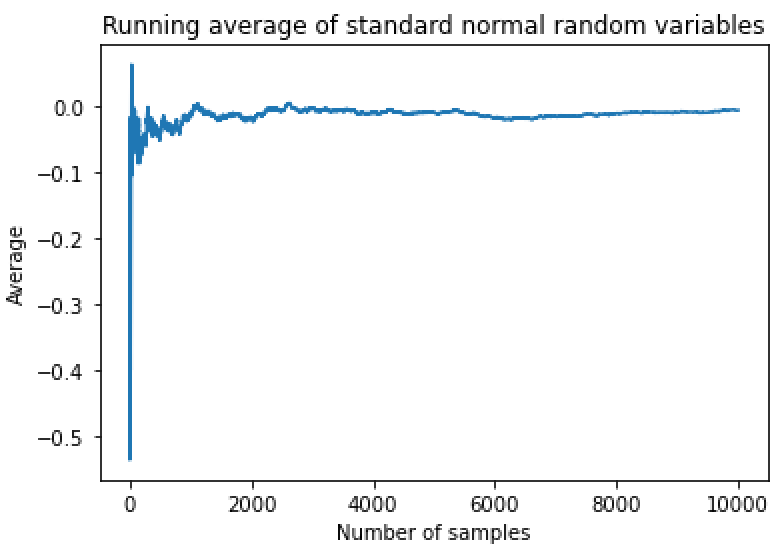
Problem 6

Part A

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

#(a) Plot running average of standard normal random variables
n = 10000
X = np.random.standard_normal(n)
running_sum = np.cumsum(X)
running_average = running_sum / np.arange(1, n+1)

plt.plot(running_average)
plt.title("Running average of standard normal random variables")
plt.xlabel("Number of samples")
plt.ylabel("Average")
plt.show()
```



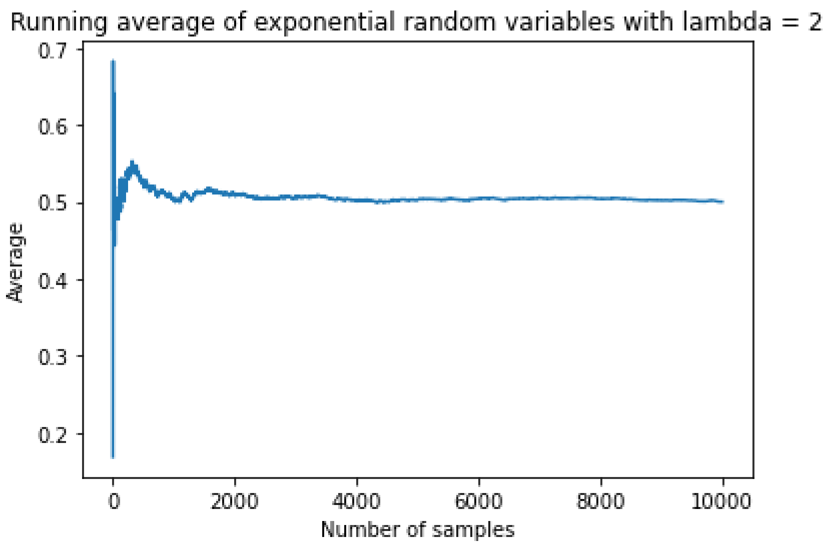
Part B

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

#(b) Plot running average of exponential random variables with parameter lambda␣
= 2
lambda_ = 2
X = np.random.exponential(scale=1/lambda_, size=n)
running_sum = np.cumsum(X)
```

```
running_average = running_sum / np.arange(1, n+1)

plt.plot(running_average)
plt.title("Running average of exponential random variables with lambda = 2")
plt.xlabel("Number of samples")
plt.ylabel("Average")
plt.show()
```

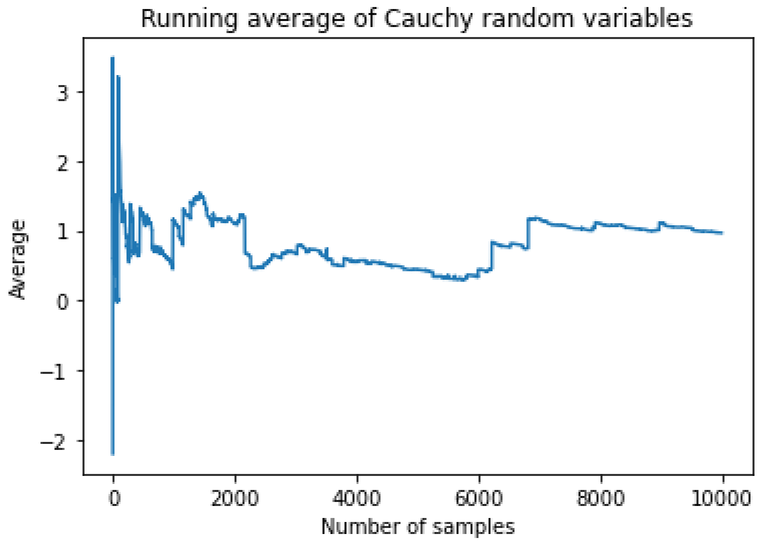


Part C

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

#(c) Plot running average of Cauchy random variables
X = np.tan(np.pi * (np.random.rand(n) - 0.5))
running_sum = np.cumsum(X)
running_average = running_sum / np.arange(1, n+1)

plt.plot(running_average)
plt.title("Running average of Cauchy random variables")
plt.xlabel("Number of samples")
plt.ylabel("Average")
plt.show()
```



Part D

The Gaussian distribution converges to its expectation value of 0. We completely expected this result. Next, the Exponential variables converge to $\frac{1}{\lambda} = .5$ which is its expectation value. The expectation value for the Cauchy variable is undefined due to the infinite variance as the plot does not converge to any value. Large spikes happen when an angle nears 0 or π as the tangent function is very large there and undefined there. Every time my code compiles and generates a new plot, it looks different every time due to its unstable behavior.