unit_2(b)_classwork

October 6, 2023

1 Unit 2(b) Classwork

The goals of this assignment are to help you (1) calculate probabilities, plot density functions, etc., involving random variables using R, (2) simulate probabilistic processes, and (3) estimate probabilities and other quantities through numerical (computer) simulation. Such simulations can be useful. We can use simulations to help confirm that we've calculated a probability "by hand" correctly, or to estimate other quantities, like areas/integrals!

2 Problem 1

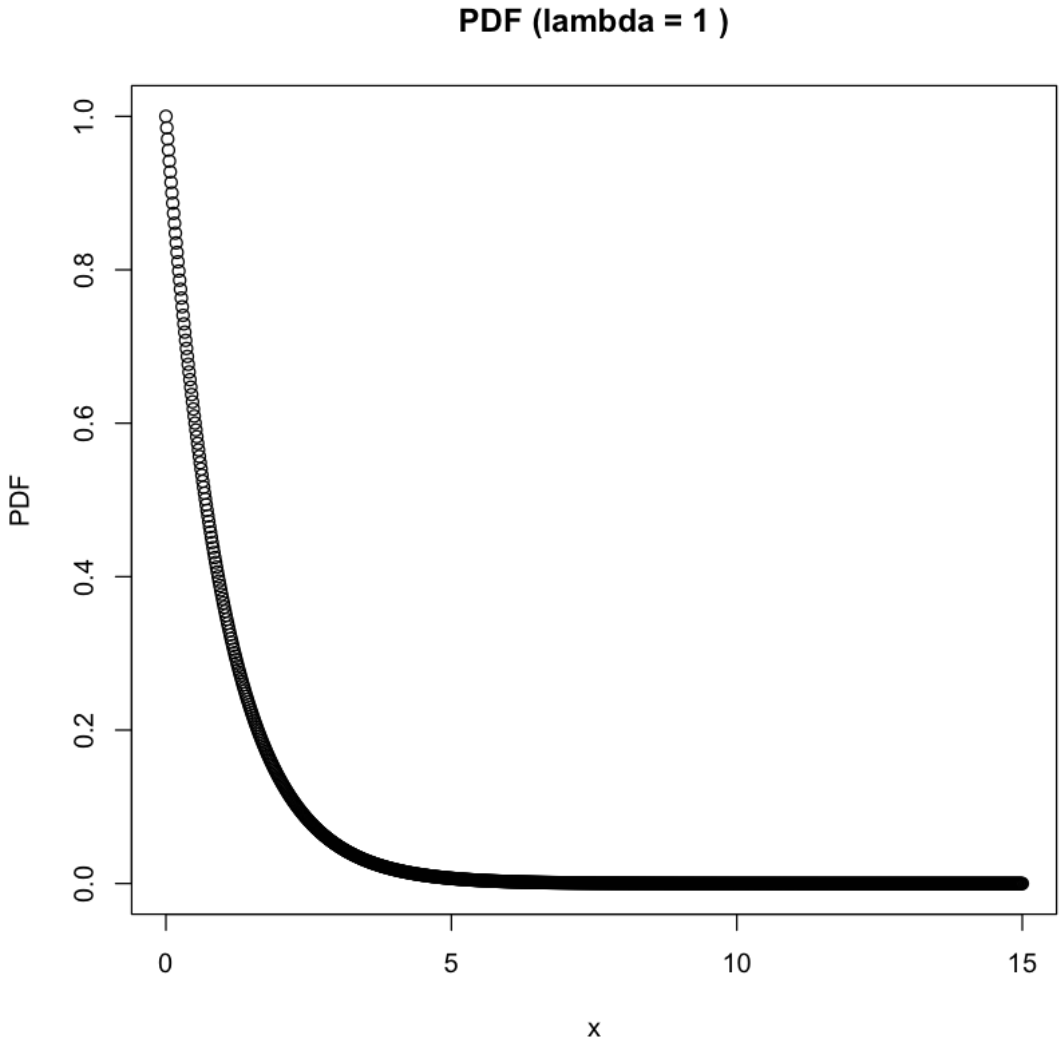
Use `dexp()` to plot the PDFs exponential distributions for $\lambda \in \{1, 2, 5, 7\}$. More concretely, create a vector consisting of the elements $1, 2, 5, 7$. Then, generate a grid of `x` values between 0 and 15 (with, say, 1000 points). Then, calculate the PDF of the exponential at each `x` value, and plot them in separate code cells.

[ ]:
```
lambda_values = c(1, 2, 5, 7)
x = seq(0, 15, length.out = 1000)

lambda_values
head(x)
```

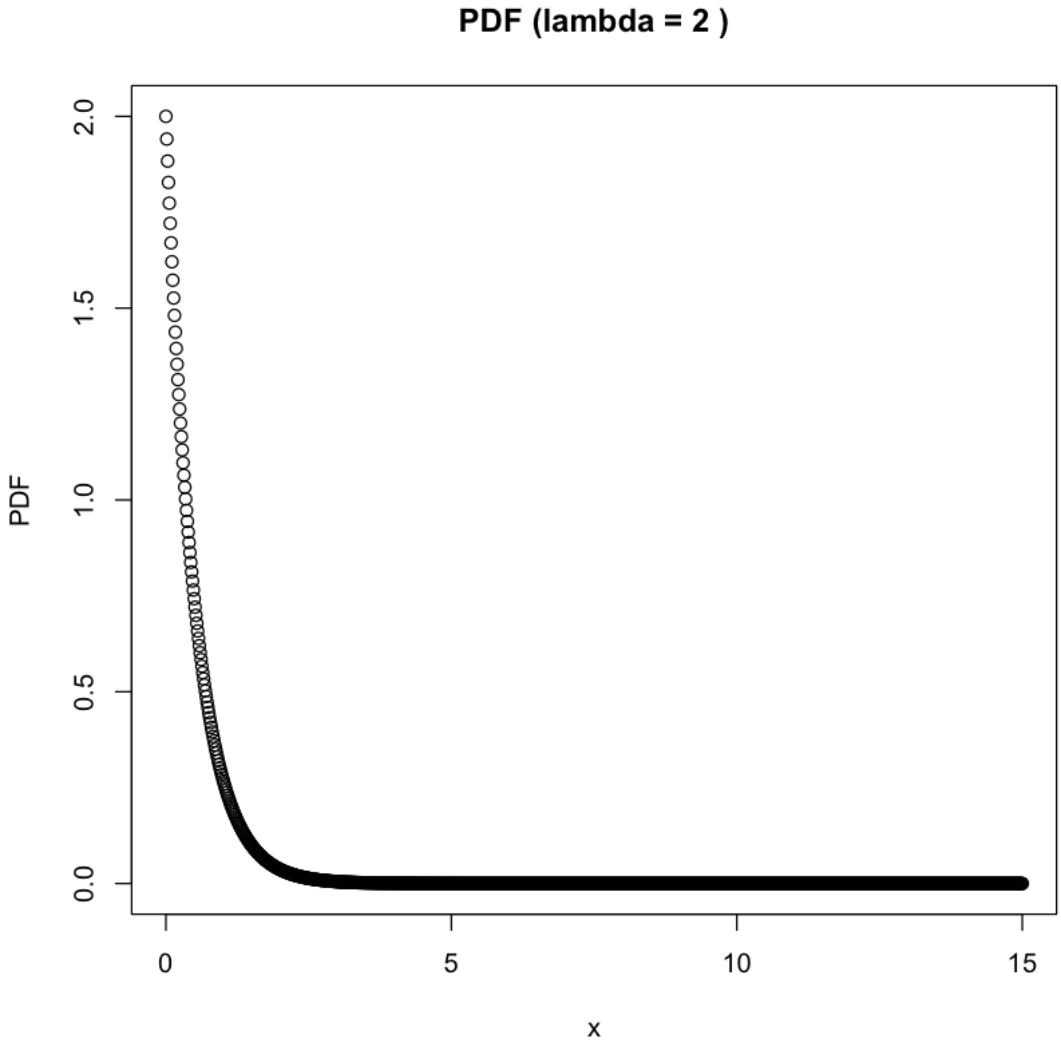
1. 1 2. 2 3. 5 4. 7

1. 0 2. 0.015015015015015 3. 0.03003003003003 4. 0.045045045045045 5. 0.0600600600600601 6. 0.0750750750750751

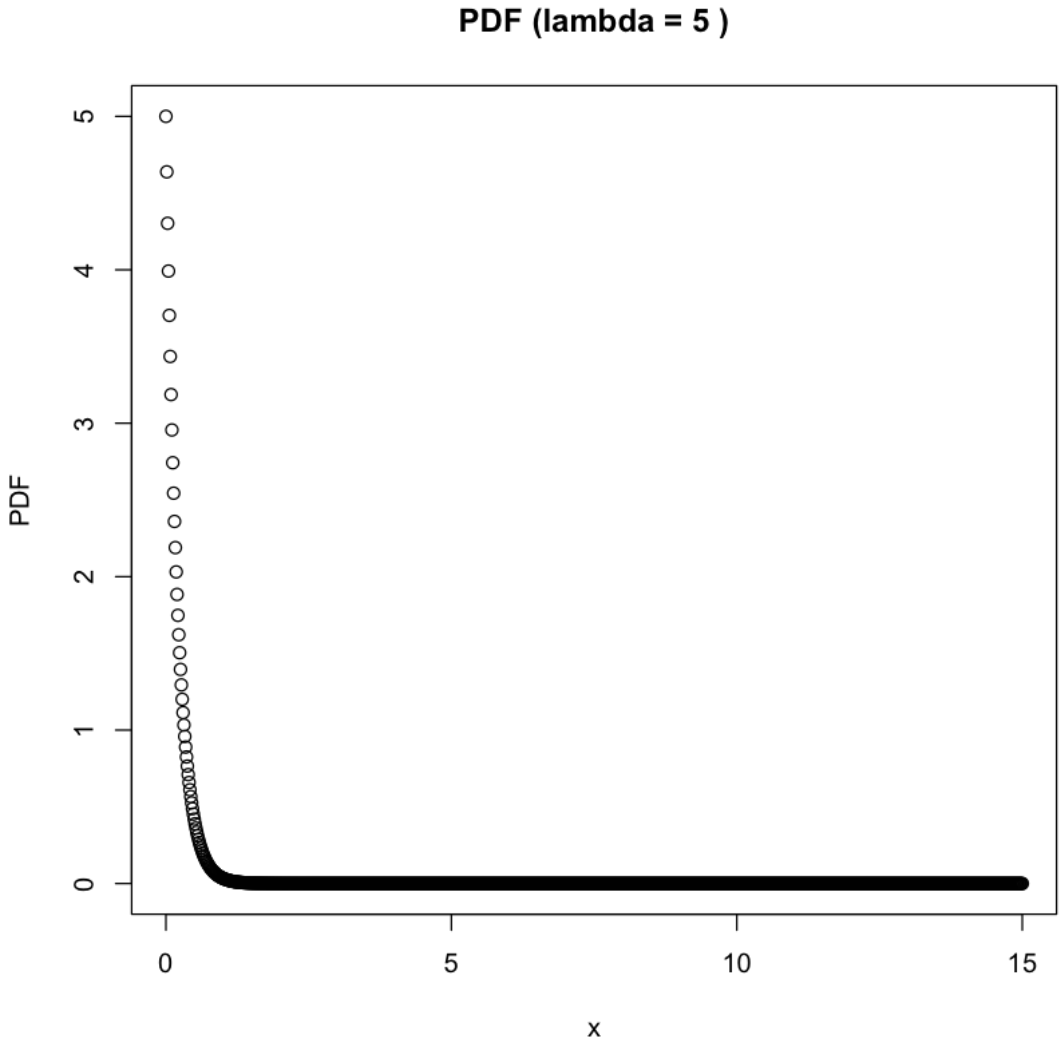
[ ]:
```
lambda = lambda_values[1]

pdf = dexp(x, rate = lambda)

# Create a plot for each lambda value
plot(x, pdf, ylim = c(0, max(pdf)),
    main = paste("PDF (lambda =", lambda, ")"),
    xlab = "x", ylab = "PDF")
```



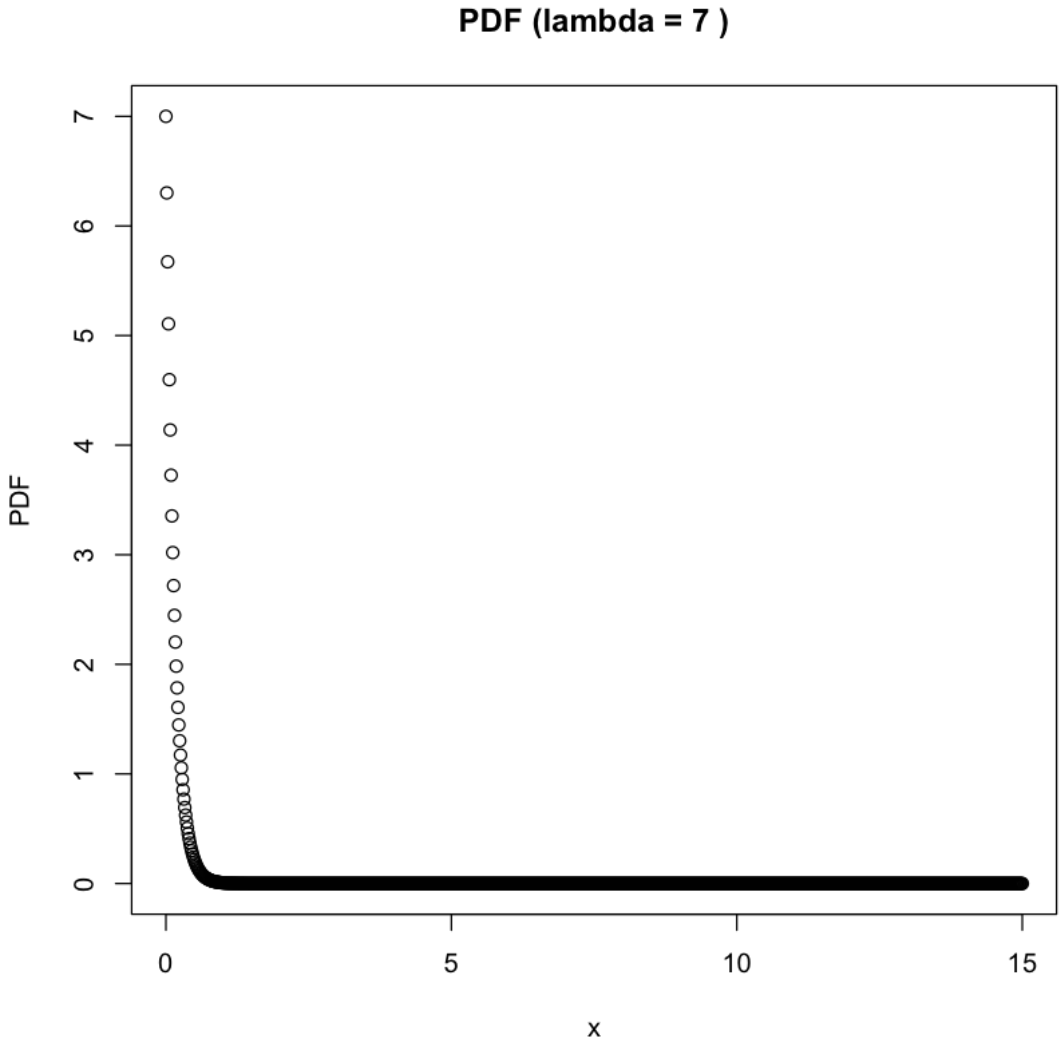
[ ]:
```
lambda = lambda_values[2]

pdf = dexp(x, rate = lambda)

# Create a plot for each lambda value
plot(x, pdf, ylim = c(0, max(pdf)),
     main = paste("PDF (lambda =", lambda, ")"),
     xlab = "x", ylab = "PDF")
```



[ ]:
```
lambda = lambda_values[3]

pdf = dexp(x, rate = lambda)

# Create a plot for each lambda value
plot(x, pdf, ylim = c(0, max(pdf)),
    main = paste("PDF (lambda =", lambda, ")"),
    xlab = "x", ylab = "PDF")
```



[ ]:
```
lambda = lambda_values[4]

pdf = dexp(x, rate = lambda)

# Create a plot for each lambda value
plot(x, pdf, ylim = c(0, max(pdf)),
     main = paste("PDF (lambda =", lambda, ")"),
     xlab = "x", ylab = "PDF")
```



3 Problem 2

Monte Carlo Estimation of π.

2.(a) Generate two vectors of $10,000$ random numbers, sampled from the uniform distribution on $(-1, 1)$. Call them x and y. Find the first quartile and 30th percentile of $X \sim U(-1, 1)$.

[ ]:
```
n = 10000

x = runif(n, min = -1, max = 1)
y = runif(n, min = -1, max = 1)
```

```
# Calculate the first quartile (25th percentile) of x
FirstQart = quantile(x, 0.25)
Thirty_percentile = quantile(x, 0.30)

FirstQart
Thirty_percentile
```

25\%: -0.482613006955944

30\%: -0.385564630106092

2.(b) Create a vector of logicals, called `incircle`, that is equal to TRUE if $x^2 + y^2 \leq 1^2$ and FALSE otherwise, where the values for x and y come from the vectors `x` and `y` generated in 2.(a)).

[ ]:
```
incircle = (x^2 + y^2 <= 1^2)
head(incircle)
```

1. TRUE 2. FALSE 3. FALSE 4. TRUE 5. TRUE 6. TRUE

2.(c) Calculate the number of values of `incircle` that are equal to TRUE. Use this value to estimate π. Describe why this works.

[ ]:
```
sum(incircle)

#find pi
est_pi = 4 * (sum(incircle) / n)

est_pi
```

7826

3.1304

We divide by number of points because it is analogous to the ratio of the area of the unit circle to the area of the enclosing "square". This is because in a uniform distribution, the probability of a point falling inside a region is proportional to the area of that region. The area of the unit circle is pi, and the area of a square enclosing the unit circle is 4 (since the diameter of the circle is 2, so 2 * 2 = 4). The ratio of the areas is pi/4.Since we are only working with one quarter as proved above, we need to multiply the sum of the TRUE statements by 4 (pi/4*4 = pi).

2.(d) How could you use this to estimate the area of a circle with radius 2 centered at the origin?

Using the same logic described in c, again we calculate the ratio of points inside the circle (incircle) to the total number of generated points (n). The ratio is the probability that a randomly generated point falls within the circle. We found in 2c that "in a uniform distribution, the probability of a point falling inside a region is proportional to the area of that region." We can just multiply this ratio by the area of the square (square_area) to find circle_area.

[ ]:
```
n = 100000
```

```r
x = runif(n, min = -2, max = 2)
y = runif(n, min = -2, max = 2)

incircle = (x^2 + y^2 <= 2^2)
square_area = (4^2)

circle_area = square_area * (sum(incircle) / n)

circle_area

#pi*4 #check answer
```

12.53936