

Lab 3: Part 2.

Dr. Alejandro Molina-Moctezuma;

2024-02-06

Before you start

This is a continuation of Lab 3. If you haven't finished lab 3 part 1, feel free to continue working on it.



You will need to submit your code. Please answer the questions by annotating your answers in the code (using the pound # symbol)



These boxes will inform you of things you need to submit or questions you need to answer!

1 Normal distribution

Let's start working on some continuous variables!

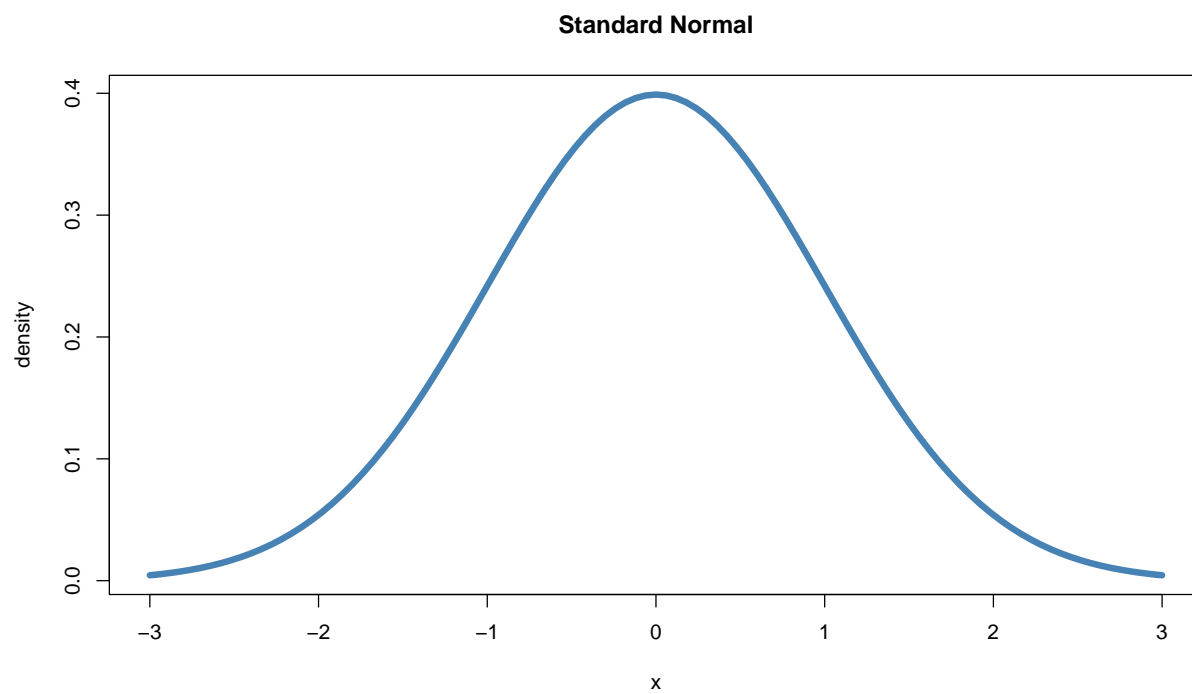
First let's remember that in continuous variables, we are calculating the area under a curve. We can't simply estimate the probability of an event.

Luckily, we know that for a normal distribution, about 95 percent of the area is found between -2, and 2 standard deviations from the mean. Actually, it 1.96, and that's the number we will be using moving forward.

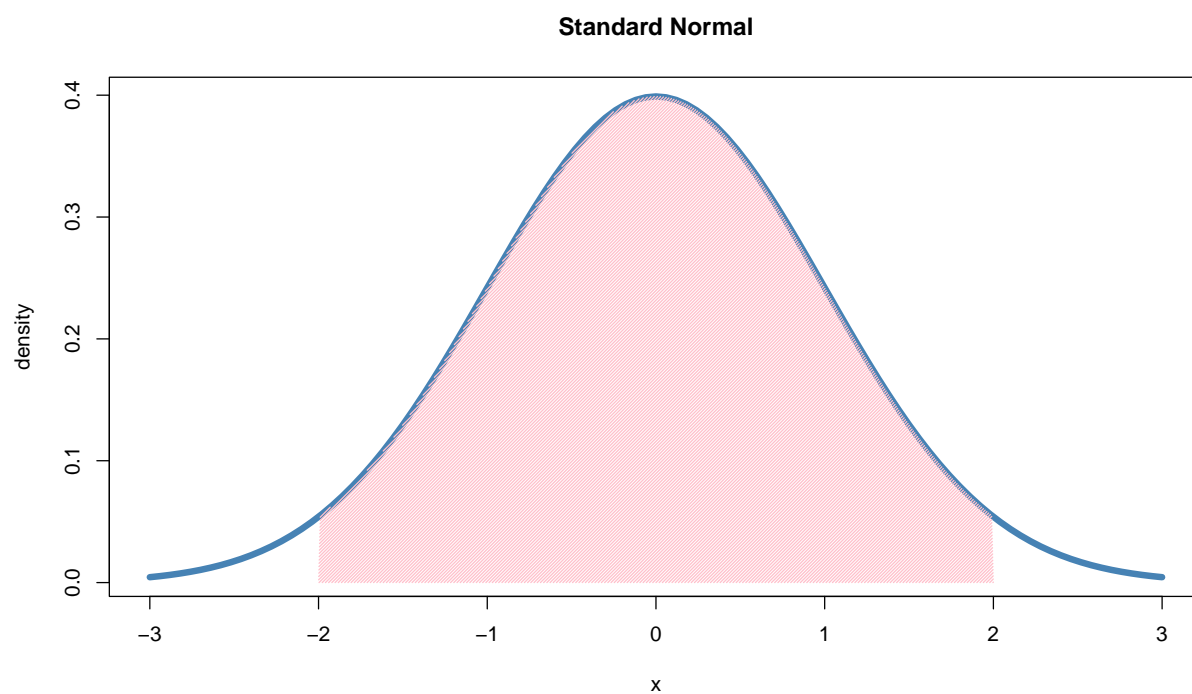
So, how do we estimate the area under the curve? Hopefully, you were paying attention during class, or check slide 54 of the week3_final.pptx presentation.

Let's imagine a normal distribution with $\mu = 0$ and with $\sigma = 1$. This means the data has a mean of zero, and a standard deviation of 1. So, if $x = 1$, that's exactly one standard deviation to the right of the mean. This makes thinking about this distribution way easier. Another way to look at it is graphically:

```
#You don't need to run this code, I am just providing so you can see how I made the plot  
curve(dnorm(x, 0, 1), from = -3, to = 3, lwd = 5, col = 'steelblue',  
      ylab = 'density', main = "Standard Normal")
```



Now, we are interested in estimating the area that's between -1.96, and 1.96. So, we want to estimate the shaded area:

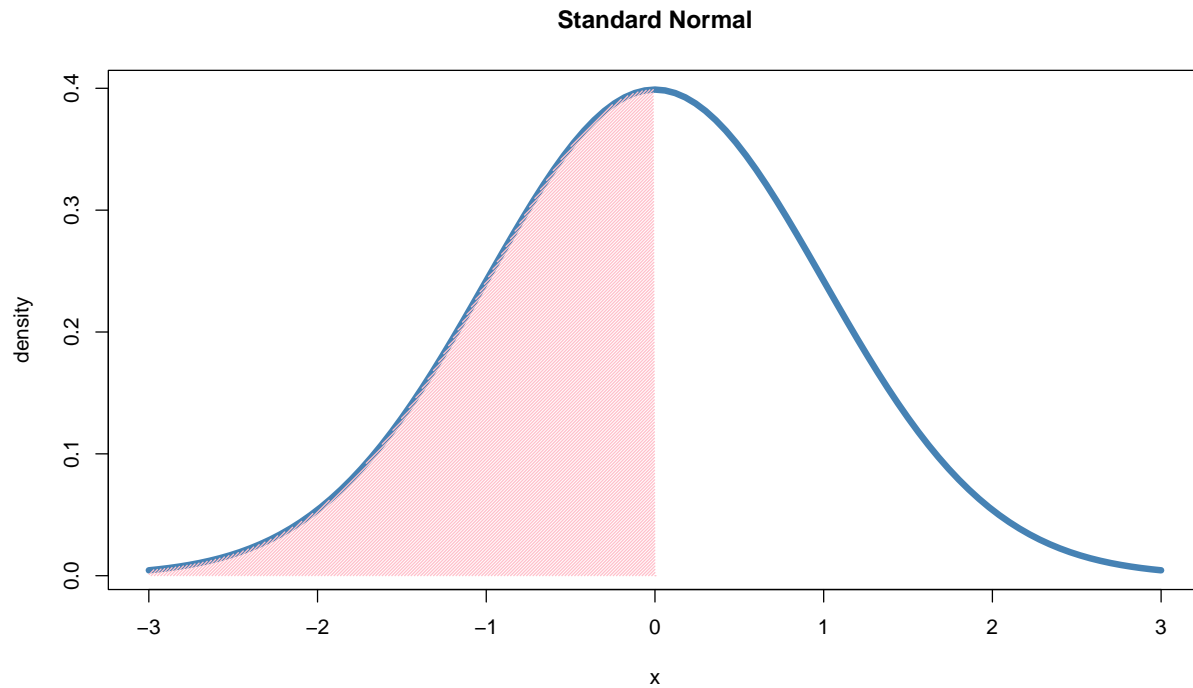


However, we can't do this directly, R only allows us to estimate the area under the curve from $-\infty$ to "x"

by using the following code:

```
pnorm(x, mean, sd)
```

So, if we wanted to estimate the area under the curve for the following:

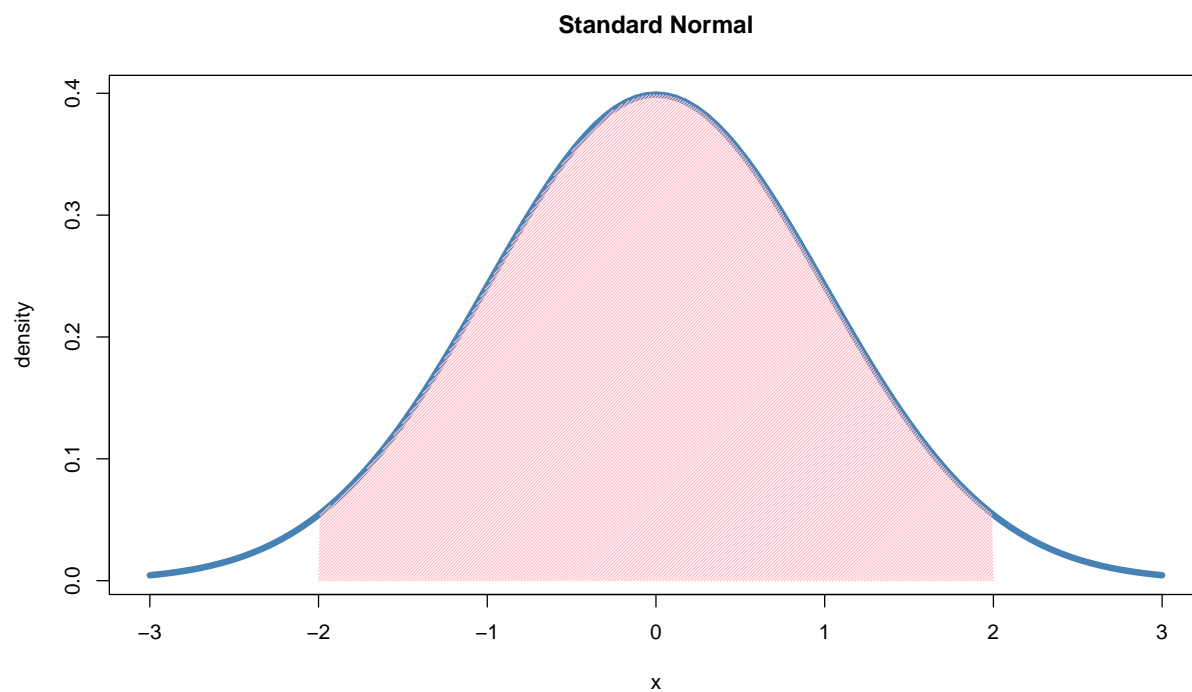


we would use:

```
pnorm(0, 0, 1)
```

[1] 0.5 Which gives us 0.5. I hope this makes sense! It's the area from $-\infty$ to 0. And because the mean = 0, this represents half of the area.

So, let's go back to our original question. We are trying to obtain the following area:

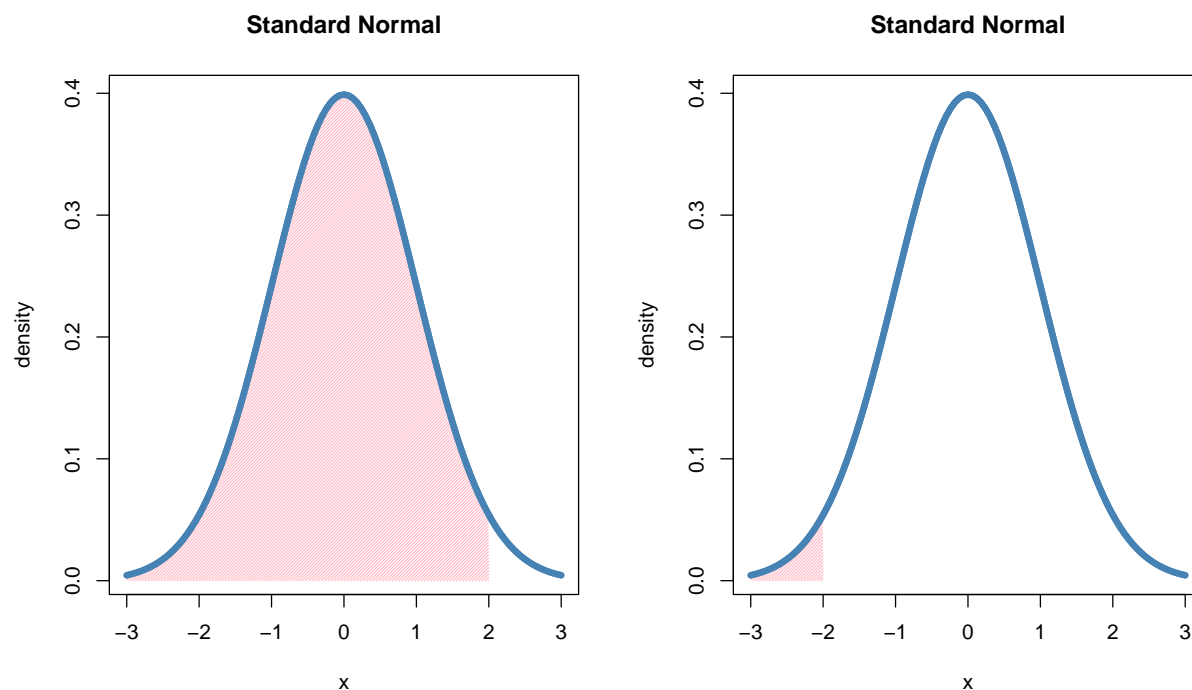


We can't really estimate this directly. But we can estimate the area from $-\infty$ to 1.96, and the area from $-\infty$ to -1.96. We can estimate both those areas using the following code that I showed previously:

```
pnorm(x, mean, sd)
```

Remember that the code gives you the area under the curve from $-\infty$ to x , for a specific mean and sd (in this case the mean is 0 and sd is 1).

So, we can estimate the following areas:



However, that is enough information to estimate the area under the curve between -1.96 and 1.96! If you are wondering how to do it, look at slide 54 for week3_final!



Q1. 4 pts. Write all the needed code to obtain the area under the curve between -1.96 and 1.96 for a normal distribution with mean = 0 and sd = 1. Report the results and interpret it

Make sure to understand what you did! It will be very important moving forward and will help you understand inferential statistics! Before we start, let's look at some useful definitions

OK, so, you are almost done with this lab!

2 Normal distribution with different values of μ and σ

Now, let's look at how different values of μ and σ affect the normal distribution.

2.1 Interactive tool

Follow the instructions to create an interactive tool that will allow you to explore the differences. **If you absolutely can't run this, skip to 2.2.**

Step 1: Downloading the shiny package In R, we often times use packages that expand the kind of stuff that we can do. In this case, the package shiny allows us to create interactive apps. To download the package, run:

```
install.packages("shiny")
```

Step 2: Create a new file Then, go to file > new file > Shiny Web App

This should open a new window. Name your app normal (or name it whatever you want), and select single file. And a new R window (with a lot of stuff in it) will open. Go ahead and delete everything that's in that window.

Step 3: code and run Copy and paste the following code:

```
library(shiny)

# Define UI for application that draws a histogram
shinyApp(

  ui = fluidPage(
    titlePanel("Mean and Standard Deviation"),
    mainPanel(
      column(4, wellPanel(
        sliderInput("mean", "Mean Blue Distribution:",
                    min = -5, max = 5, value = 0, step = 1)
      ),
        wellPanel(
          sliderInput("sd", "SD Blue Distribution:",
                      min = 0.5, max = 5, value = 1, step = 0.25)
        ),
        wellPanel(
          sliderInput("mean2", "Mean Red Distribution:",
                      min = -5, max = 5, value = 2, step = 1)
        ),
        wellPanel(
          sliderInput("sd2", "SD Red Distribution:",
                      min = 0.5, max = 5, value = 2, step = .25)
        )
      ),
      column(8,
        plotOutput('curves'),

        br(),
        br()
      )
    )
  ),

  server = function(input, output, session) {
    # Create a random name for the log file

    x <- seq(-20, 20, 0.01)
```

```

# This observer adds an entry to the log file every time
# input$n changes.

maxvalue <- reactive({
  max(c(dnorm(x, input$mean, input$sd), dnorm(x, input$mean2, input$sd2)))
})

output$curves <- renderPlot({
  plot(NULL, xlim=c(-20,20), ylim=c(0, maxvalue()), ylab = 'Probability', xlab = 'X',
    main = "Two Normal Distributions" )
  curve(dnorm(x, input$mean, input$sd), from = -20, to = 20, lwd = 4,
    col = 'steelblue', add=T)
  curve(dnorm(x, input$mean2, input$sd2), add = TRUE, lwd = 4, col = 'tomato')

})

}
)

```

Finally, you might have noticed that a little play button that says “Run App” is where the “Run” code button used to be. Go ahead and click that.

If it worked, you are now looking at a fully functional app that you created in R. This is a great way to present results



Q2. 2 pts. a) Set one of the means as 0 and sd as 1 and describe the shape of that distribution (is it symmetrical, concentrated, spread, etc.), b) set the other distribution with the same mean, but a higher sd. and explain what differences do you see?, c) play around with different values from μ and σ and answer, how does changing μ impacts the distribution, how does changing σ impacts it?

Congrats! You are done with lab 3!

2.2 Alternative activity

If you were able to run the app, then you are done, and this section isn't necessary for you. If you couldn't run it, then keep reading.

Don't worry if the app didn't work for you. Let's explore the effects of μ and σ on the distribution.

run the following code:

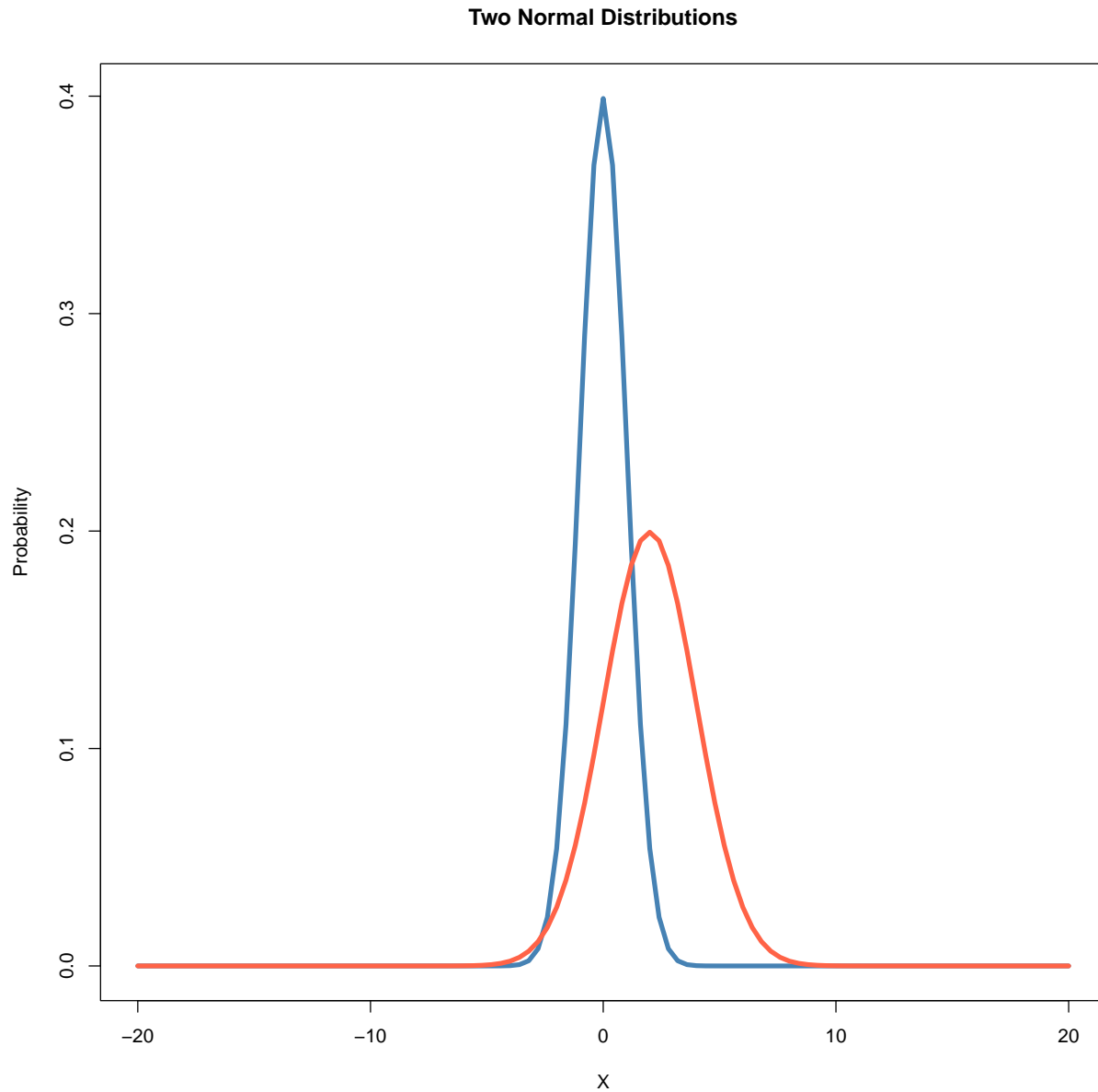
```

m1<-0
sd1<-1
m2<-2
sd2<-2

maxvalue <-      max(c(dnorm(x, m1, sd1), dnorm(x, m2, sd2)))
plot(NULL, xlim=c(-20,20), ylim=c(0, maxvalue), ylab = 'Probability', xlab = 'X',
  main = "Two Normal Distributions" )
  curve(dnorm(x, m1, sd1), from = -20, to = 20, lwd = 4,

```

```
col = 'steelblue',add=T)
curve(dnorm(x, m2, sd2), add = TRUE, lwd = 4, col = 'tomato')
```



Now, play with that code, change the values of the means ($m1$ and $m2$), choose any value between -5 and 5. Change the values of sd as well, you can choose any number between 0.1 and 5. Play with it for a while and see how the mean and sd affect the distribution.

Once you are done, answer Q2 (in section 2.1 of this lab)

Resources

These are [hyperlinks](#)

The base R cheat-sheet

Dr. Fordyce Poisson

An introduction to R

Mark: a Gentel Introduction

End of document