**Valor de la media esperada:**

Si lo dice el enunciado usar ese, sí no:

# Función para calcular el valor esperado de la media de una muestra

calcular\_valor\_esperado\_media <- function(valores\_muestra) {

# Calculamos la media de la muestra

media\_muestra <- mean(valores\_muestra)

return(media\_muestra)

}

# Ejemplo de uso de la función con valores de muestra ingresados por el usuario

# Puedes ingresar tus propios valores separados por comas entre paréntesis

mis\_valores <- c(10, 15, 20, 25, 30) # Aquí puedes cambiar estos valores por los tuyos

# Llamamos a la función con los valores ingresados

valor\_esperado\_media <- calcular\_valor\_esperado\_media(mis\_valores)

# Imprimimos el resultado

print(valor\_esperado\_media)

**Calculo de variancia con datos de probabilidad:**

# Encontrar el puntaje z para el percentil 0.881

z\_score <- qnorm(0.881)

# Media poblacional

media\_poblacional <- 801

# Encontrar la desviación estándar poblacional (sigma)

desviacion\_estandar\_poblacional <- (830 - media\_poblacional) / z\_score

# Tamaño de la muestra

n <- 64

# Calcular la varianza del sueldo de los trabajadores y trabajadoras

varianza\_sueldo <- (desviacion\_estandar\_poblacional^2) \* n

# Imprimir el resultado

print(varianza\_sueldo)

**probabilidad de que la suma de n de menor/mayor a p: (inc)**

menor

# Parámetros dados en el problema

media <- 801

n <- 64

# Suma máxima deseada

suma\_maxima <- 51635

# Calcular la desviación estándar para la suma de los sueldos

# La varianza de la suma de variables aleatorias independientes es n veces la varianza original

# Por lo tanto, la desviación estándar es la raíz cuadrada de n veces la varianza original

desviacion\_suma <- sqrt(n) \* sqrt(801^2) # Dado que la varianza es 801^2 en una distribución normal

# Calcular la probabilidad de que la suma sea menor que suma\_maxima

probabilidad <- pnorm(suma\_maxima, mean = n \* media, sd = desviacion\_suma)

# Imprimir la probabilidad

print(probabilidad)

mayor

# Parámetros dados en el problema

media <- 801

n <- 64

# Suma mínima deseada

suma\_minima <- 51635

# Calcular la desviación estándar para la suma de los sueldos

desviacion\_suma <- sqrt(n) \* sqrt(801^2) # Raíz cuadrada de n veces la varianza original

# Calcular la probabilidad de que la suma sea mayor que suma\_minima

probabilidad <- 1 - pnorm(suma\_minima, mean = n \* media, sd = desviacion\_suma)

# Imprimir la probabilidad

print(probabilidad)

Ejercicios propuestos sesión 5 -> lab4

ejercicios propuestos sesión 6 -> lab5

**generar funcion de densidad valor medio y des estandart:**

# Parámetros de la distribución

media <- 5.1

desviacion\_estandar <- 1.055

# Valores de x para los que queremos calcular la densidad de probabilidad

x <- seq(0, 10, length.out = 1000)

# Calcular los valores de densidad de probabilidad para los valores de x

densidad\_probabilidad <- dnorm(x, mean = media, sd = desviacion\_estandar)

# Graficar la función de densidad

plot(x, densidad\_probabilidad, type = "l", lwd = 2, col = "blue",

xlab = "Valor", ylab = "Densidad de probabilidad",

main = "Función de Densidad de una Distribución Normal")

**Probabilidad de un valor mayor a x con desv estandart y valor medio:**

# Parámetros de la distribución

media <- 5.1

desviacion\_estandar <- 1.055

valor <- 4

# Calcular el Z-score (estandarización)

z\_score <- (valor - media) / desviacion\_estandar

# Calcular la probabilidad usando la función de distribución acumulada (1 - P(X <= 4))

probabilidad\_mas\_4cl <- 1 - pnorm(valor, mean = media, sd = desviacion\_estandar)

# Imprimir la probabilidad

print(probabilidad\_mas\_4cl)

**probabilidad de que la muestra este entre x y:**

# Parámetros de la distribución

media <- 5.1

desviacion\_estandar <- 1.055

valor\_inferior <- 3

valor\_superior <- 5.1

# Calcular la probabilidad usando la función de distribución acumulada

probabilidad <- pnorm(valor\_superior, mean = media, sd = desviacion\_estandar) -

pnorm(valor\_inferior, mean = media, sd = desviacion\_estandar)

# Imprimir la probabilidad

print(probabilidad)

**Calculo percentil x fdensidad:**

# Parámetros de la distribución

media <- 5.1

desviacion\_estandar <- 1.055

percentil <- 0.74

# Calcular el percentil usando la función qnorm()

percentil\_074 <- qnorm(percentil, mean = media, sd = desviacion\_estandar)

# Imprimir el percentil

print(percentil\_074)

**Contenido de x muestras, el valor sea menor a n:**

# Parámetros de la distribución individual de una probeta

media <- 5.1

desviacion\_estandar <- 1.055

# Parámetros para la suma de 14 probetas

n <- 14

media\_total <- n \* media

desviacion\_total <- sqrt(n) \* desviacion\_estandar

# Valor límite

valor\_limite <- 74.5

# Calcular la probabilidad usando la función de distribución acumulada

probabilidad <- pnorm(valor\_limite, mean = media\_total, sd = desviacion\_total)

# Imprimir la probabilidad

print(probabilidad)

**Nos dan un dato de un valor medio x por un intervalo de tiempo:**

**prob diferente a x:**

# Parámetro lambda

lambda <- 9

# Número límite de llamadas deseadas

num\_llamadas\_limite <- 12

# Calcular la probabilidad utilizando la función de distribución acumulada de Poisson

probabilidad\_menos\_12 <- ppois(num\_llamadas\_limite - 1, lambda)

# Imprimir la probabilidad

print(probabilidad\_menos\_12)

**Probabilidad de otro valor x en otro intervalo de tiempo:**

# Parámetro lambda para 2 horas

lambda\_2\_horas <- 9 \* 2 # La tasa promedio se multiplica por el número de horas

# Número de llamadas deseadas

num\_llamadas <- 18

# Calcular la probabilidad utilizando la función de masa de probabilidad de Poisson

probabilidad\_18\_llamadas <- dpois(num\_llamadas, lambda\_2\_horas)

# Imprimir la probabilidad

print(probabilidad\_18\_llamadas)

**Simulacion mas experimentos:**

# Número de simulaciones

num\_simulaciones <- 100000

# Parámetro lambda

lambda <- 9

# Simular el número de llamadas en una hora para 100000 experimentos

llamadas\_simuladas <- rpois(num\_simulaciones, lambda)

# Calcular la varianza de las llamadas simuladas

varianza\_simulada <- var(llamadas\_simuladas)

# Imprimir la varianza simulada

print(varianza\_simulada)

**grafica funcion de funcion de probabilidad de variable en tiempo:**

# Parámetro lambda

lambda <- 9

# Números posibles de llamadas (de 0 a 20, por ejemplo)

llamadas <- 0:20

# Calcular la función de masa de probabilidad de Poisson

probabilidades <- dpois(llamadas, lambda)

# Graficar la función de probabilidad

plot(llamadas, probabilidades, type = "h", lwd = 2, col = "blue",

xlab = "Cantidad de llamadas en una hora", ylab = "Probabilidad",

main = "Distribución de Poisson para llamadas en una hora")