# Vector de mascotas  
mascotas <- c('loro', 'perro', 'gato', 'gallina', 'hamster', 'cerdo', 'ternero', 'caballo', 'cabra')  
  
# Agregar 20 mascotas más  
mascotas <- c(mascotas, 'conejo', 'tortuga', 'pez', 'canario', 'hurón', 'iguana', 'serpiente', 'ratón', 'araña', 'camaleón', 'pato', 'pavo', 'oveja', 'vaca', 'burro', 'ganso', 'paloma', 'cisne', 'lagarto', 'erizo')  
  
# Shuffle the mascotas vector  
mascotas <- sample(mascotas)  
  
# Seleccionar una muestra aleatoria de 3 elementos sin repetición  
selmascota <- sample(mascotas, 4)  
  
nombremascota1 <- selmascota[1]  
nombremascota2 <- selmascota[2]  
nombremascota3 <- selmascota[3]  
nombremascota4 <- selmascota[4]  
  
# Porcentaje total de encuestados  
total\_encuestados <- 100  
  
# Crear secuencia del 60 al 1000 de 10 en 10  
numeros <- seq(60, 1000, 10)  
  
# Eliminamos el número 100 del vector  
numeros\_sin\_100 <- setdiff(numeros, 100)  
  
# Ahora hacemos el muestreo de este nuevo vector  
enkuestados <- sample(numeros\_sin\_100, 1)  
  
# Ensure that enkuestados is a multiple of 10  
enkuestados <- enkuestados - (enkuestados %% 10)

# Generar tres números enteros para los porcentajes. Su suma siempre debe ser igual a 100  
generar\_vector\_unico <- function() {  
 # Generate two random integers that are multiples of 10 between 10 and 60  
 num1 <- sample(seq(10, 60, 10), 1)  
 num2 <- sample(seq(10, 60, 10), 1)  
   
 # Ensure that the sum of the two numbers is less than 100  
 while (num1 + num2 >= 100) {  
 num1 <- sample(seq(10, 60, 10), 1)  
 num2 <- sample(seq(10, 60, 10), 1)  
 }  
   
 # Calculate the third number  
 num3 <- 100 - num1 - num2  
   
 # Return the vector of numbers  
 return(c(num1, num2, num3))  
}  
  
# Generar y mostrar el vector de porcentajes  
vector\_resultado <- generar\_vector\_unico()  
  
mashor <- max(vector\_resultado)  
  
porxentaje1 <- vector\_resultado[1]  
porxentaje2 <- vector\_resultado[2]  
porxentaje3 <- vector\_resultado[3]  
####################################################  
  
maskota1 <- (enkuestados\*vector\_resultado[1])/100 # Personas que adoptan maskota1  
maskota2 <- (enkuestados\*vector\_resultado[2])/100  
maskota3 <- (enkuestados\*vector\_resultado[3])/100  
  
mashiormaskota <- max(maskota1, maskota2, maskota3)

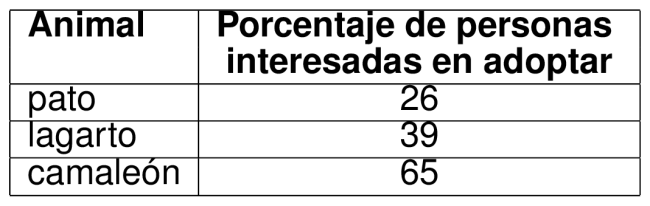
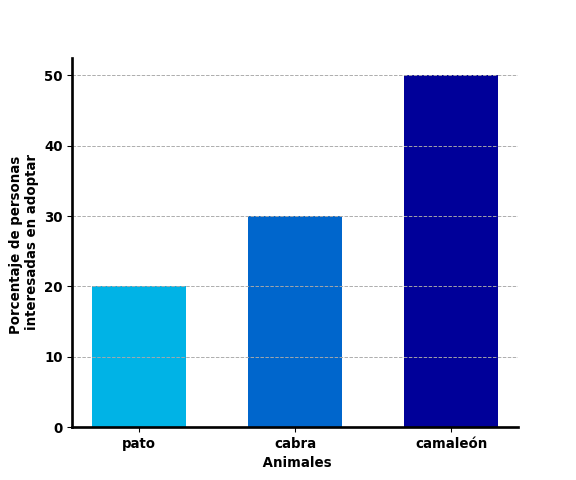
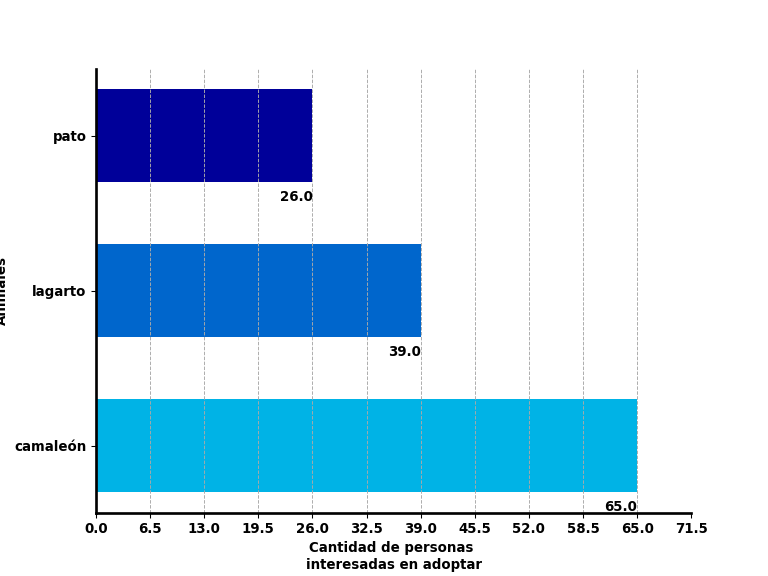
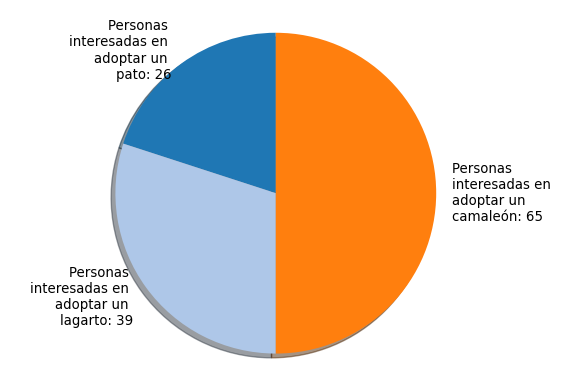
image01 <- '  
\\begin{tikzpicture}  
 \\node{  
\\begin{tabular}{|l|c|}  
\\hline  
\\textbf{Animal} & \\textbf{Porcentaje de personas } \\\\   
 & \\textbf{interesadas en adoptar} \\\\ \\hline  
 %s & %s \\\\ \\hline  
 %s & %s \\\\ \\hline  
 %s & %s \\\\ \\hline  
\\end{tabular}  
};  
\\end{tikzpicture}  
'  
  
demas <- sample  
  
maskota1\_format <- sprintf("%.0f", maskota1)  
maskota2\_format <- sprintf("%.0f", maskota2)  
maskota3\_format <- sprintf("%.0f", maskota3)  
  
image01 <-sprintf(image01, selmascota[1], maskota1\_format, selmascota[2], maskota2\_format, selmascota[3], maskota3\_format)

# Question

El líder de un programa de adopción de mascotas encuestó a 130 personas para conocer qué animal les interesaría adoptar. Del total de encuestados, el 20% adoptaría un pato, el 30% adoptaría un(a) lagarto y el 50% adoptaría un(a) camaleón.

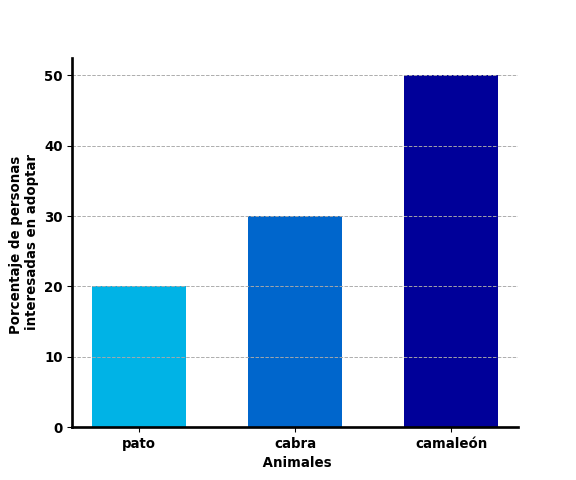
¿Cuál de las siguientes representaciones NO muestra correctamente la información recolectada en la encuesta?

## Answerlist

* 
* 
* 
* 

# Solution

La gráfica que representa la opción correcta es



# Meta-information

exname: I\_1796473/2023-Cuadernillo-Matematicas-11-2(single-choice) extype: schoice exsolution: 0100 exshuffle: TRUE

# ```{r unit\_tests, echo = FALSE, results = “hide”}

# library(testthat)

# # Pruebas unitarias adicionales

# test\_that(“Pruebas de validación de datos”, {

# # Pruebas de estructura

# expect\_type(mascotas, “character”)

# expect\_type(selmascota, “character”)

# expect\_length(selmascota, 3)

# # Pruebas de rango

# expect\_true(all(vector\_resultado >= 0 & vector\_resultado <= 100))

# expect\_equal(sum(vector\_resultado), 100)

# expect\_true(enkuestados >= 60 && enkuestados <= 1000)

# # Pruebas de distribución (ejemplo)

# expect\_true(mashor == max(vector\_resultado))

# # Pruebas de relación entre variables

# tolerance <- 1

# expect\_true(abs(maskota1 + maskota2 + maskota3 - enkuestados) < tolerance)

# })