

PEC2: Programación y probabilidad

Marco Russo

Noviembre, 2025

Sección 1. Fundamentos de programación y BBDD (5 puntos)

Ejercicio 1 (1 punto)

Se pide de mostrar el resultado obtenido de esta matriz:

```
genes <- matrix(c(
  "BRCA1", 120, "Reparación de ADN",
  "TP53", 300, "Supresor tumoral",
  "MYC", 500, "Oncogén",
  "GAPDH", 1000, "Control interno",
  "EGFR", 250, "Receptor de membrana"
), nrow = 5, byrow = TRUE)
print("Matriz de genes:")
```

```
## [1] "Matriz de genes:"
```

```
print(genes)
```

```
##      [,1]  [,2]  [,3]
## [1,] "BRCA1" "120" "Reparación de ADN"
## [2,] "TP53"  "300" "Supresor tumoral"
## [3,] "MYC"   "500" "Oncogén"
## [4,] "GAPDH" "1000" "Control interno"
## [5,] "EGFR"  "250"  "Receptor de membrana"
```

1.1

De acorde a la documentación oficial, se utiliza `c` porque una `matrix` se crea de la unión de varios vectores. `c` está por **combine**, de hecho junta valores en un vector o en una lista. Una matriz como tal está formata por vectores mínimo 1x2. En nuestro ejemplo, `genes` es una `matrix` 5x3

1.2

Teniendo en cuenta que una matriz debería ser vectores de un solo tipo de datos. En nuestro caso un `dataframe` sería la estructura óptima para representar genes, por tener diferentes tipos de datos, es decir valores heterogéneos.

1.3

Si probamos a quitar el parámetro `nrow=5` y `byrow=TRUE` observamos que ocurre a nuestra matriz:

```
genes_2 <- matrix(c(
  "BRCA1", 120, "Reparación de ADN",
  "TP53", 300, "Supresor tumoral",
  "MYC", 500, "Oncogén",
  "GAPDH", 1000, "Control interno",
  "EGFR", 250, "Receptor de membrana"
))
print("Matriz de genes:")
```

```
## [1] "Matriz de genes:"
```

```
print(genes_2)
```

```
##      [,1]
## [1,] "BRCA1"
## [2,] "120"
## [3,] "Reparación de ADN"
## [4,] "TP53"
## [5,] "300"
## [6,] "Supresor tumoral"
## [7,] "MYC"
## [8,] "500"
## [9,] "Oncogén"
## [10,] "GAPDH"
## [11,] "1000"
## [12,] "Control interno"
## [13,] "EGFR"
## [14,] "250"
## [15,] "Receptor de membrana"
```

En práctica si el primer argumento, `nrow` crearía una matriz de una sola columna con 15 filas. Con el segundo argumento, `byrow`, los datos se introducirían por columna, por lo que vemos la alteración de la estructura original.

1.4

Crearíamos un *chunk* que recorra las filas de la matriz y muestre en pantalla lista de genes que sean oncogenes o supresores tumorales, junto con su nivel de expresión.

```
cat("\nLista de genes relevantes en cáncer:\n")

##
## Lista de genes relevantes en cáncer:

for (i in 1:nrow(genes)) {
  categoria <- genes[i, 3]

  if (categoria == "Oncogén" || categoria == "Supresor tumoral") {
    nombre_gen <- genes[i, 1]
    expresion <- genes[i, 2]
    cat("Gen:", nombre_gen, "- Expresión:", expresion, "TPM\n")
  }
}
```

```
## Gen: TP53 - Expresión: 300 TPM
## Gen: MYC - Expresión: 500 TPM
```

Obtenemos este resultado utilizando un bucle `for` para recorrer las filas de la matriz `genes` y guardamos en `categoria` el resultado de la tercera columna. Finalmente con la condición `if` filtramos si la categoría está entre Oncogén o Supresor tumoral para que podamos mostrar los resultados específicos.

Finalmente mostramos los resultados en pantalla como de ejemplo.

Ejercicio 2 (2 puntos)

Se pide calcular el índice de expresión relativa de un gen en un experimento de qPCR comparando el nivel de expresión de un gen de interés con el de un gen de referencia.

2.1

Para ello definimos una función llamada `indice_expresion` que recibe dos parámetros:

- `exp_interes`
- `exp_referencia`

para obtener el siguiente:

$$Indice = \frac{exp_interes}{exp_referencia}$$

La función en R sería el siguiente:

```

indice_expresion <- function(exp_interes, exp_referencia) {
  # Comprobaremos si son numéricos para que admite solo estos tipo de valores
  if (!is.numeric(exp_interes) || !is.numeric(exp_referencia)) {
    stop("Error: Los parámetros deben ser numéricos")
  }

  # Validaremos el valor de referencia sea diferente de cero. Por cuando
  # tengamos la operación de división no incurramos al error division by zero.
  if (exp_referencia == 0) {
    stop("Error: El valor de referencia no puede ser cero")
  }

  # Calcularemos el índice
  indice <- exp_interes / exp_referencia

  return(indice)
}

```

2.2

Aplicamos la función teniendo en cuenta los datos de `exp_interes 250`, `exp_referencia+ 100`

```
indice_expresion(250, 100)
```

```
## [1] 2.5
```

El valor resultado del índice de expresión es 2.5.

2.3

Probaremos la función con diferentes combinaciones para poder detectar eventuales errores y cuando los valores sean 0 o no numéricos.

```

# Prueba 1
indice_expresion(50, 200)

```

```
## [1] 0.25
```

```

# Prueba 2
indice_expresion(0, 100)

```

```
## [1] 0
```

```
# Prueba 6
indice_expresion(200, 15.25)
```

```
## [1] 13.11475
```

2.4

En la fase de análisis de esta función, podemos utilizar el comando `readline()` que nos permite leer en modo interactivo los valores desde la terminal.

Aplicaremos la modificación para realizarlo:

El uso de la función `readline` permite mayor flexibilidad, ya que la validación visto anteriormente es manual. Por lo que es una ventaja no tener que declarar los valores en cada celda. Sin embargo, incurre a errores de entrada de los valores de parte del usuario, por lo que siempre requiere una intervención manual.

Una de las ventajas es facilitar el uso sin conocimiento de programación. Aunque si integráramos `trycatch` o flujos condicionales en caso de `errors`, podría tener ciertos uso recomendados en los casos educativos.

Por contra, no es un uso recomendado para procesos automatizados o en entornos productivos, por lo que es preferible utilizarlos solo para los fines donde el usuario mantiene un cierto control sobre la función.

Ejercicio 3 (2 puntos)

Se procede con el utilizar el dataset indicado desde Rdocumentation ToothGrowth cuyas variables son

- `[,1]` len numeric Tooth length
- `[,2]` supp factor Supplement type (VC or OJ).
- `[,3]` dose numeric Dose in milligrams/day

3.1

Utilizaremos los comandos de base de datos RSQLite para obtener los resultados de los siguientes ejercicios.

```
# Cargaremos las librerías
if (!require('RSQLite')) install.packages('RSQLite')
```

```
## Loading required package: RSQLite
```

```
library(RSQLite)
library(datasets)
```

```
# Cargamos el dataset ToothGrowth
data("ToothGrowth")
```

```
# Efectuaremos la conexión a una base de datos en memoria
con <- dbConnect(RSQLite::SQLite(), ":memory:")
```

```
# Guardamos el dataset en la base de datos
dbWriteTable(con, "ToothGrowth", ToothGrowth)

# Realizamos un control que la tabla se ha creado correctamente
cat("Tablas en la base de datos:\n")
```

```
## Tablas en la base de datos:
```

```
print(dbListTables(con))
```

```
## [1] "ToothGrowth"
```

```
cat("\nPrimeras filas de la tabla:\n")
```

```
##
## Primeras filas de la tabla:
```

```
print(dbGetQuery(con, "SELECT * FROM ToothGrowth LIMIT 5"))
```

```
##      len supp dose
## 1  4.2    VC  0.5
## 2 11.5    VC  0.5
## 3  7.3    VC  0.5
## 4  5.8    VC  0.5
## 5  6.4    VC  0.5
```

Para poder mostrar la longitud media del diente, característica `len` filtrado para aquellas observaciones que recibieron una dosis de al menos 1 mg/día

```
# Consulta SQL
query_a <- "
  SELECT AVG(len) AS longitud_media
  FROM ToothGrowth
  WHERE dose >= 1
"

# Aplicamos la query y obtenemos el resultado
resultado_a <- dbGetQuery(con, query_a)
print(resultado_a)
```

```
##      longitud_media
## 1              22.9175
```

```
cat("\nResultado 3a: La longitud media del diente en las cobayas que recibieron")
```

```
##
## Resultado 3a: La longitud media del diente en las cobayas que recibieron
```

```
cat("\nal menos 1 mg/día de vitamina C es de", round(resultado_a$longitud_media, 2), "unidades.\n")
```

```
##
```

```
## al menos 1 mg/día de vitamina C es de 22.92 unidades.
```

3.2

Mostramos ahora la longitud media del diente `len` en las observaciones que recibieron una dosis de al menos 1 mg/día

```
# creamos la query SQL
query_b <- "
  SELECT len AS longitud, dose AS dosis
  FROM ToothGrowth
  WHERE supp = 'OJ' AND len > 20
  ORDER BY len DESC
"

resultado_b <- dbGetQuery(con, query_b)
print(resultado_b)
```

```
##      longitud dosis
## 1         30.9   2.0
## 2         29.4   2.0
## 3         27.3   1.0
## 4         27.3   2.0
## 5         26.4   1.0
## 6         26.4   2.0
## 7         26.4   2.0
## 8         25.8   1.0
## 9         25.5   2.0
## 10        25.2   1.0
## 11        24.8   2.0
## 12        24.5   2.0
## 13        23.6   1.0
## 14        23.3   1.0
## 15        23.0   2.0
## 16        22.4   2.0
## 17        21.5   0.5
## 18        21.2   1.0
```

```
cat("\nSe encontraron", nrow(resultado_b), "cobayas tratadas con jugo de naranja")
```

```
##
```

```
## Se encontraron 18 cobayas tratadas con jugo de naranja
```

```
cat("\ncon longitud del diente superior a 20 unidades.\n")
```

```
##
## con longitud del diente superior a 20 unidades.
```

3.3

Mostramos cuántos animales fueron tratados con cada tipo de suplemento `supp` agrupando los resultados por suplemento y ordenándolos de mayor a menor número de animales.

```
# creamos la query SQL
query_c <- "
  SELECT
    supp AS tipo_suplemento,
    COUNT(*) AS numero_animales
  FROM ToothGrowth
  GROUP BY supp
  ORDER BY numero_animales DESC
"

resultado_c <- dbGetQuery(con, query_c)
print(resultado_c)
```

```
##   tipo_suplemento numero_animales
## 1                VC              30
## 2                OJ              30
```

```
cat("\nResultado obtenido:\n")
```

```
##
## Resultado obtenido:
```

```
for (i in 1:nrow(resultado_c)) {
  tipo <- ifelse(resultado_c$tipo_suplemento[i] == "OJ",
                 "Jugo de naranja (OJ)",
                 "Ácido ascórbico (VC)")
  cat("-", tipo, ":", resultado_c$numero_animales[i], "animales\n")
}
```

```
## - Ácido ascórbico (VC) : 30 animales
## - Jugo de naranja (OJ) : 30 animales
```

3.4

Mostramos finalmente la longitud media del diente agrupada por dosis y tipo de suplemento `supp` y dose

```
# creamos la query SQL
query_d <- "
  SELECT
    supp AS tipo_suplemento,
    dose AS dosis,
    AVG(len) AS longitud_media,
    COUNT(*) AS num_observaciones
  FROM ToothGrowth
  GROUP BY supp, dose
  ORDER BY supp, dose
"

resultado_d <- dbGetQuery(con, query_d)
print(resultado_d)
```

```
##      tipo_suplemento dosis longitud_media num_observaciones
## 1                OJ   0.5           13.23                10
## 2                OJ   1.0           22.70                10
## 3                OJ   2.0           26.06                10
## 4                VC   0.5            7.98                10
## 5                VC   1.0           16.77                10
## 6                VC   2.0           26.14                10
```

```
cat("\n--- Análisis de resultados ---\n")
```

```
##
## --- Análisis de resultados ---
```

```
cat("\nJugo de naranja (OJ):\n")
```

```
##
## Jugo de naranja (OJ):
```

```
oj_data <- resultado_d[resultado_d$tipo_suplemento == "OJ", ]
for (i in 1:nrow(oj_data)) {
  cat("  Dosis", oj_data$dosis[i], "mg/día: longitud media =",
      round(oj_data$longitud_media[i], 2), "unidades\n")
}
```

```
##   Dosis 0.5 mg/día: longitud media = 13.23 unidades
##   Dosis 1 mg/día: longitud media = 22.7 unidades
##   Dosis 2 mg/día: longitud media = 26.06 unidades
```

```
cat("\nÁcido ascórbico (VC):\n")
```

```
##
## Ácido ascórbico (VC):
```

```
vc_data <- resultado_d[resultado_d$tipo_suplemento == "VC", ]
for (i in 1:nrow(vc_data)) {
  cat(" Dosis", vc_data$dosis[i], "mg/día: longitud media =",
      round(vc_data$longitud_media[i], 2), "unidades\n")
}
```

```
## Dosis 0.5 mg/día: longitud media = 7.98 unidades
## Dosis 1 mg/día: longitud media = 16.77 unidades
## Dosis 2 mg/día: longitud media = 26.14 unidades
```

```
cat("\nObservación: Se observa que el crecimiento del diente aumenta con la dosis")
```

```
##
## Observación: Se observa que el crecimiento del diente aumenta con la dosis
```

```
cat("\nen ambos tipos de suplemento. El jugo de naranja (OJ) parece ser más efectivo")
```

```
##
## en ambos tipos de suplemento. El jugo de naranja (OJ) parece ser más efectivo
```

```
cat("\nque el ácido ascórbico (VC) en dosis bajas y medias.\n")
```

```
##
## que el ácido ascórbico (VC) en dosis bajas y medias.
```

Finalmente cerraremos la base de datos

```
dbDisconnect(con)
cat("\nBase de datos cerrada correctamente.\n")
```

```
##
## Base de datos cerrada correctamente.
```

Sección 2. Probabilidad y simulaciones

Ejercicio 4 (2.5 puntos)

Preparamos el entorno cargando el resto de librerías que serán útiles para realizar un análisis exploratorio de los datos.

Definiremos los parámetros de las distribuciones log-normales

```
# Sin antibiótico
meanlog_sin <- 1.2
sdlog_sin <- 0.3

# Con antibiótico
meanlog_con <- 1.5
sdlog_con <- 0.4
```

```

cat("Parámetros de las distribuciones log-normales:\n")

## Parámetros de las distribuciones log-normales:

cat("Sin antibiótico: meanlog =", meanlog_sin, ", sdlog =", sdlog_sin, "\n")

## Sin antibiótico: meanlog = 1.2 , sdlog = 0.3

cat("Con antibiótico: meanlog =", meanlog_con, ", sdlog =", sdlog_con, "\n\n")

## Con antibiótico: meanlog = 1.5 , sdlog = 0.4

# Creamos la secuencia de valores para el eje x
x <- seq(0, 15, length.out = 500)

# Calcularemos las densidades
densidad_sin <- dlnorm(x, meanlog = meanlog_sin, sdlog = sdlog_sin)
densidad_con <- dlnorm(x, meanlog = meanlog_con, sdlog = sdlog_con)

# Creamos el dataframe para utilizarlo sucesivamente en ggplot
df_densidades <- data.frame(
  tiempo = rep(x, 2),
  densidad = c(densidad_sin, densidad_con),
  escenario = rep(c("Sin antibiótico", "Con antibiótico"), each = length(x))
)

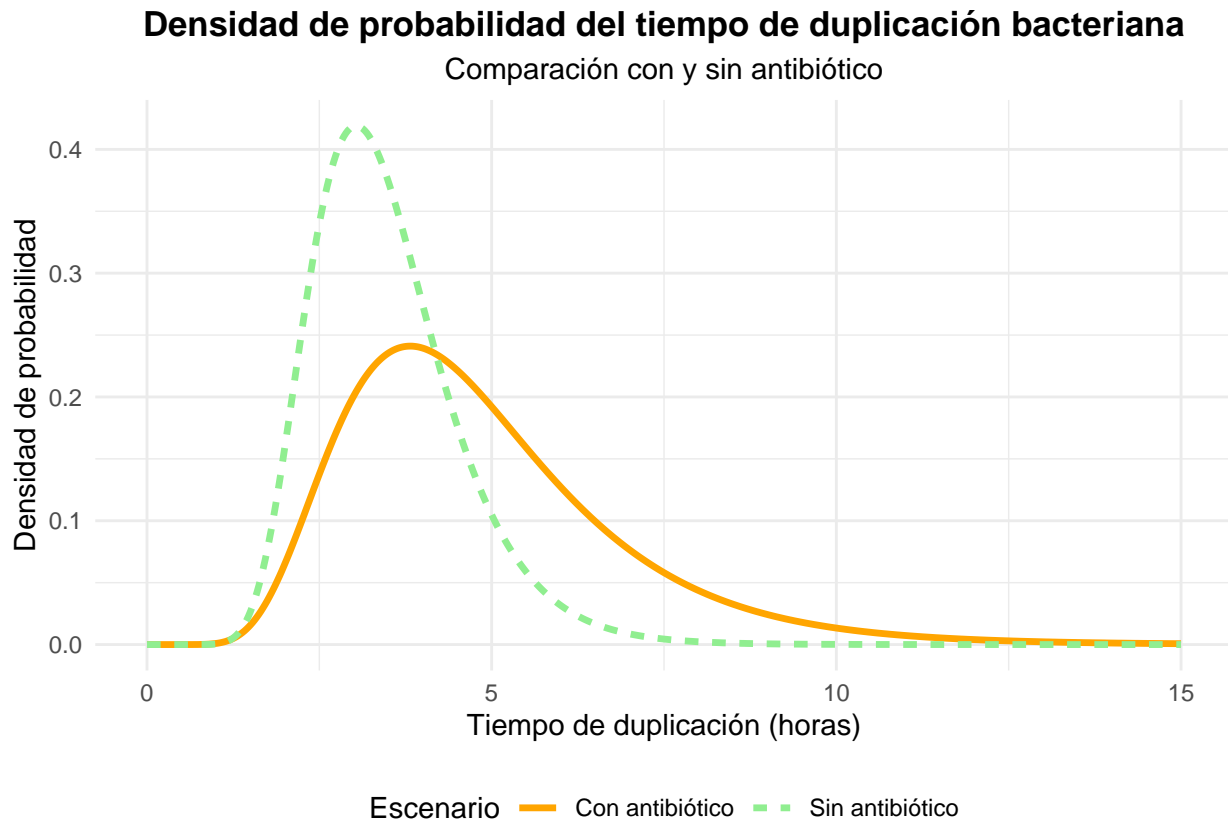
# Gráficamos
grafico_densidades <- ggplot(df_densidades, aes(x = tiempo, y = densidad,
  color = escenario,
  linetype = escenario)) +

  geom_line(size = 1.2) +
  labs(title = "Densidad de probabilidad del tiempo de duplicación bacteriana",
    subtitle = "Comparación con y sin antibiótico",
    x = "Tiempo de duplicación (horas)",
    y = "Densidad de probabilidad",
    color = "Escenario",
    linetype = "Escenario") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"),
    plot.subtitle = element_text(hjust = 0.5),
    legend.position = "bottom") +
  scale_color_manual(values = c("Sin antibiótico" = "lightgreen",
    "Con antibiótico" = "orange"))

## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

```

```
print(grafico_densidades)
```



El gráfico representado nos indica que el tiempo de duplicación en horas es más alta la probabilidad en las observaciones **sin antibiótico**. Tenemos el pico del 40% a las 2.5/3h mientras que es inferior al 25% en los casos **con antibiótico** donde el tiempo de duplicación está superando las 4 horas.

4.2

Calcularemos la probabilidad de que el tiempo sea inferior a 2 horas en ambos casos

```
# Definiremos las probabilidades con y sin antibióticos
prob_sin_menor2 <- plnorm(2, meanlog = meanlog_sin, sdlog = sdlog_sin)
prob_con_menor2 <- plnorm(2, meanlog = meanlog_con, sdlog = sdlog_con)
```

```
cat(" Sin antibiótico: P(T < 2) =", round(prob_sin_menor2, 4),
    "(", round(prob_sin_menor2 * 100, 2), "%)\n")
```

```
## Sin antibiótico: P(T < 2) = 0.0456 ( 4.56 %)
```

```
cat(" Con antibiótico: P(T < 2) =", round(prob_con_menor2, 4),
    "(", round(prob_con_menor2 * 100, 2), "%)\n\n")
```

```
## Con antibiótico: P(T < 2) = 0.0218 ( 2.18 %)
```

En esta simulación los tiempos de duplicación inferior a 2 horas, la probabilidad de duplicar su tamaño es del 2.18% en los casos **con antibiótico** respecto al 4.56% a los **sin antibiótico**. Es decir, que inferior a las 2 horas, una colonia de bacterias sin antibiótico, tarda el doble para duplicar su tamaño respecto a un escenario sin antibiótico.

Calcularemos en los casos sea superior a las 5 horas.

```
prob_sin_mayor5 <- 1 - plnorm(5, meanlog = meanlog_sin, sdlog = sdlog_sin)
prob_con_mayor5 <- 1 - plnorm(5, meanlog = meanlog_con, sdlog = sdlog_con)

cat(" Sin antibiótico: P(T > 5) =", round(prob_sin_mayor5, 4),
    "(", round(prob_sin_mayor5 * 100, 2), "%)\n")
```

```
## Sin antibiótico: P(T > 5) = 0.0862 ( 8.62 %)
```

```
cat(" Con antibiótico: P(T > 5) =", round(prob_con_mayor5, 4),
    "(", round(prob_con_mayor5 * 100, 2), "%)\n\n")
```

```
## Con antibiótico: P(T > 5) = 0.3922 ( 39.22 %)
```

En los casos de un escenario con antibiótico, la probabilidad de duplicar su tamaño, es del 39.22% respecto al escenario sin antibiótico, que se queda en un 8.62%. Es decir que una colonia es más resistente en un corto plazo de tiempo, como vimos era inferior a las 3 horas, mientras que superando las 5 horas, estamos frente a 4 veces más su pérdida de eficacia.

4.3

Realizaremos ahora unas 10,000 simulaciones para los dos escenarios, estimando la media, varianza y percentiles 25,50,75 respecto al tiempo de duplicación.

```
# Primero fijaremos la semilla para poder reproducirlo después.
set.seed(17)

# Configuramos el número de simulaciones
n_sim <- 10000

# Aplicamos las simulaciones sin antibiótico
sim_sin <- rlnorm(n_sim, meanlog = meanlog_sin, sdlog = sdlog_sin)

# Aplicamos las simulaciones con antibiótico
sim_con <- rlnorm(n_sim, meanlog = meanlog_con, sdlog = sdlog_con)
```

Mostramos los resultados de las simulaciones sin antibiótico:

```
cat("---ESCENARIO SIN ANTIBIÓTICO ---\n\n")
```

```
## ---ESCENARIO SIN ANTIBIÓTICO ---
```

```
# Valores estadísticos simulaciones
media_sim_sin <- mean(sim_sin)
varianza_sim_sin <- var(sim_sin)
p25_sim_sin <- quantile(sim_sin, 0.25)
p50_sim_sin <- quantile(sim_sin, 0.50)
p75_sim_sin <- quantile(sim_sin, 0.75)

cat("Resultados de las simulaciones:\n")
```

```
## Resultados de las simulaciones:
```

```
cat("  Media:      ", round(media_sim_sin, 4), "\n")
```

```
##  Media:      3.4789
```

```
cat("  Varianza:   ", round(varianza_sim_sin, 4), "\n")
```

```
##  Varianza:   1.1305
```

```
cat("  Percentil 25%:", round(p25_sim_sin, 4), "\n")
```

```
##  Percentil 25%: 2.7223
```

```
cat("  Percentil 50%:", round(p50_sim_sin, 4), "\n")
```

```
##  Percentil 50%: 3.3244
```

```
cat("  Percentil 75%:", round(p75_sim_sin, 4), "\n\n")
```

```
##  Percentil 75%: 4.0582
```

Aplicaremos ahora con los valores exactos teóricos:

Para distribución log-normal:

$$E[X] = \exp(\mu + \sigma^2/2)$$

$$Var[X] = (\exp(\sigma^2) - 1) * \exp(2\mu + \sigma^2)$$

```
media_exacta_sin <- exp(meanlog_sin + sdlog_sin^2/2)
varianza_exacta_sin <- (exp(sdlog_sin^2) - 1) * exp(2*meanlog_sin + sdlog_sin^2)
p25_exacto_sin <- qlnorm(0.25, meanlog = meanlog_sin, sdlog = sdlog_sin)
p50_exacto_sin <- qlnorm(0.50, meanlog = meanlog_sin, sdlog = sdlog_sin)
p75_exacto_sin <- qlnorm(0.75, meanlog = meanlog_sin, sdlog = sdlog_sin)

cat("Valores exactos (teóricos):\n")
```

```
## Valores exactos (teóricos):
```

```
cat(" Media:      ", round(media_exacta_sin, 4), "\n")
```

```
## Media:      3.4729
```

```
cat(" Varianza:   ", round(varianza_exacta_sin, 4), "\n")
```

```
## Varianza:   1.1359
```

```
cat(" Percentil 25%:", round(p25_exacto_sin, 4), "\n")
```

```
## Percentil 25%: 2.7119
```

```
cat(" Percentil 50%:", round(p50_exacto_sin, 4), "\n")
```

```
## Percentil 50%: 3.3201
```

```
cat(" Percentil 75%:", round(p75_exacto_sin, 4), "\n\n")
```

```
## Percentil 75%: 4.0647
```

Realizaremos ahora y mostraremos las diferencias:

```
cat("Diferencias (Simulación - Exacto):\n")
```

```
## Diferencias (Simulación - Exacto):
```

```
cat(" Media:      ", round(media_sim_sin - media_exacta_sin, 4), "\n")
```

```
## Media:      0.0059
```

```
cat(" Varianza:   ", round(varianza_sim_sin - varianza_exacta_sin, 4), "\n")
```

```
## Varianza:   -0.0054
```

```
cat(" Percentil 25%:", round(p25_sim_sin - p25_exacto_sin, 4), "\n")
```

```
## Percentil 25%: 0.0104
```

```
cat(" Percentil 50%:", round(p50_sim_sin - p50_exacto_sin, 4), "\n")
```

```
## Percentil 50%: 0.0043
```

```
cat(" Percentil 75%:", round(p75_sim_sin - p75_exacto_sin, 4), "\n\n")
```

```
## Percentil 75%: -0.0066
```

Las diferencias absolutas son cercanas al cero excepto para el percentil 25% donde tenemos una variación respecto al exacto de 1 punto.

A continuación aplicaremos lo mismo para el escenario con antibiótico.

```
cat("\n--- ESCENARIO CON ANTIBIÓTICO ---\n\n")
```

```
##
```

```
## --- ESCENARIO CON ANTIBIÓTICO ---
```

```
# Valores estadísticos simulaciones
```

```
media_sim_con <- mean(sim_con)
```

```
varianza_sim_con <- var(sim_con)
```

```
p25_sim_con <- quantile(sim_con, 0.25)
```

```
p50_sim_con <- quantile(sim_con, 0.50)
```

```
p75_sim_con <- quantile(sim_con, 0.75)
```

```
cat("Resultados de las simulaciones:\n")
```

```
## Resultados de las simulaciones:
```

```
cat("  Media:      ", round(media_sim_con, 4), "\n")
```

```
##  Media:      4.8945
```

```
cat("  Varianza:   ", round(varianza_sim_con, 4), "\n")
```

```
##  Varianza:   4.2347
```

```
cat("  Percentil 25%:", round(p25_sim_con, 4), "\n")
```

```
##  Percentil 25%: 3.4449
```

```
cat("  Percentil 50%:", round(p50_sim_con, 4), "\n")
```

```
##  Percentil 50%: 4.5227
```

```
cat("  Percentil 75%:", round(p75_sim_con, 4), "\n\n")
```

```
##  Percentil 75%: 5.9122
```

```
# Valores exactos (teóricos)
```

```
media_exacta_con <- exp(meanlog_con + sdlog_con^2/2)
```

```
varianza_exacta_con <- (exp(sdlog_con^2) - 1) * exp(2*meanlog_con + sdlog_con^2)
```

```
p25_exacto_con <- qlnorm(0.25, meanlog = meanlog_con, sdlog = sdlog_con)
```

```
p50_exacto_con <- qlnorm(0.50, meanlog = meanlog_con, sdlog = sdlog_con)
```

```
p75_exacto_con <- qlnorm(0.75, meanlog = meanlog_con, sdlog = sdlog_con)
```

```
cat("Valores exactos (teóricos):\n")
```

```
## Valores exactos (teóricos):
```

```

cat("  Media:      ", round(media_exacta_con, 4), "\n")

##  Media:      4.855

cat("  Varianza:   ", round(varianza_exacta_con, 4), "\n")

##  Varianza:   4.0898

cat("  Percentil 25%:", round(p25_exacto_con, 4), "\n")

##  Percentil 25%: 3.4219

cat("  Percentil 50%:", round(p50_exacto_con, 4), "\n")

##  Percentil 50%: 4.4817

cat("  Percentil 75%:", round(p75_exacto_con, 4), "\n\n")

##  Percentil 75%: 5.8697

cat("Diferencias (Simulación - Exacto):\n")

## Diferencias (Simulación - Exacto):

cat("  Media:      ", round(media_sim_con - media_exacta_con, 4), "\n")

##  Media:      0.0395

cat("  Varianza:   ", round(varianza_sim_con - varianza_exacta_con, 4), "\n")

##  Varianza:   0.145

cat("  Percentil 25%:", round(p25_sim_con - p25_exacto_con, 4), "\n")

##  Percentil 25%: 0.023

cat("  Percentil 50%:", round(p50_sim_con - p50_exacto_con, 4), "\n")

##  Percentil 50%: 0.041

cat("  Percentil 75%:", round(p75_sim_con - p75_exacto_con, 4), "\n\n")

##  Percentil 75%: 0.0426

```

Para este tipo de escenario con antibiótico, las diferencia van cambiando con un casi ~ 4 en la media, un 14.5 en la varianza. Por que se podría considerar una cierta heterogeneidad en la respuesta al antibiótico respecto al escenario sin antibiótico.

4.4

Existe un dilema ético relacionado con el uso excesivo de antibióticos y la resistencia bacteriana. El problema de la resistencia bacteriana reside en su uso excesivo e inadecuado de antibióticos ha llevado a uno de los mayores desafíos de salud pública del siglo XXI: la resistencia bacteriana. Por ejemplo, hemos visto en las simulaciones, los antibióticos ralentizan el crecimiento bacteriano, pero su uso indiscriminado selecciona cepas resistentes que son cada vez más difíciles de tratar.

Si hablamos de responsabilidades, se puede confirmar que el uso excesivo de antibióticos por parte de individuos, uno por no completar los tratamientos y segundo por automedicación, trae consecuencias para toda la sociedad. Además, existe una cierta presión en agricultura y ganadería ya que el uso masivo de antibióticos en la producción animal contribuye significativamente al problema, pero su restricción podría afectar la seguridad alimentaria y economía.

Además de los problemas de acceso vs restricción; beneficio presente vs futuro; desarrollo de nuevos antibióticos, existen ciertas implicaciones para la práctica clínica y científica.

Se debería hacer un uso racional, cuando los antibióticos deben prescribirse solo cuando sean estrictamente necesarios (infecciones bacterianas confirmadas). Hay que educar a profesionales sanitarios y público general sobre el uso responsable de antibióticos. Los gobiernos deberían incrementar las inversiones I+D+i, así como las empresas privadas deberían ser más concientes de estas nuevas inversiones privadas orientadas a las nuevas terapias alternativas, (fagos, inmunoterapias) y mejores métodos de diagnóstico rápido. Requiere la implementación de sistemas de monitorización de resistencias a nivel global.

Nosotros mismos como estudiantes universitarios y futuros investigadores y profesionales en bioinformática y bioestadística, tenemos la responsabilidad de mejorar ciertos procesos. Ejemplo, desarrollar herramientas para detectar y predecir resistencias; analizar datos epidemiológicos para guiar políticas de salud pública; promover la investigación en terapias alternativas; comunicar efectivamente los riesgos del uso inadecuado de antibióticos.

Finalmente y para concluir esta pequeña reflexión, podemos hacer mención que la resistencia bacteriana es un problema que requiere acción inmediata desde múltiples frentes: científico, político, social y ético.

Ejercicio 5 (2.5 puntos)

Pasamos ahora a querer analizar el rendimiento de una vacuna contra una nueva enfermedad infecciosa. Suponiendo que el 3% de la población no desarrolla inmunidad tras recibir una vacuna, analizaremos si una persona no desarrolla inmunidad, la probabilidad de infectarse es del 40% al contrario sería del 2%

5.1

Queremos ver la probabilidad de que al menos 5 se infecten, utilizando binomial y haciendo 10,000 simulaciones.

Creamos el escenario inicial para poder contestar a la primera pregunta.

```
# definiremos los parámetros del problema indicado
prop_sin_inmunidad <- 0.03
prop_con_inmunidad <- 0.97

prob_infeccion_sin_inmunidad <- 0.40
prob_infeccion_con_inmunidad <- 0.02
```

Pasamos al cálculo de la probabilidad de infección para una persona vacunada, siendo la fórmula:

$$P(\text{inf}) = P(\text{sinInmunidad}) * P(\text{inf}|\text{sinInmunidad}) + P(\text{conInmunidad}) * P(\text{inf}|\text{conInmunidad})$$

```
# fijamos el número de personas vacunadas
```

```
vacunadas <- 200
```

```
# Calcular probabilidad de infección para una persona vacunada
```

```
prob_infeccion_general <- prop_sin_inmunidad * prob_infeccion_sin_inmunidad +  
                          prop_con_inmunidad * prob_infeccion_con_inmunidad
```

```
cat("Probabilidad de infección para una persona vacunada:\n")
```

```
## Probabilidad de infección para una persona vacunada:
```

```
cat("P(infección) =", prop_sin_inmunidad, "x", prob_infeccion_sin_inmunidad,  
    "+", prop_con_inmunidad, "x", prob_infeccion_con_inmunidad, "\n")
```

```
## P(infección) = 0.03 × 0.4 + 0.97 × 0.02
```

```
cat("P(infección) =", round(prob_infeccion_general, 4),  
    "(", round(prob_infeccion_general * 100, 2), "%)\n\n")
```

```
## P(infección) = 0.0314 ( 3.14 %)
```

Si aplicaríamos el cálculo exacto con la binomial tendríamos:

$$P(X \geq 5) = 1 - P(X \leq 4)$$

```
prob_ge_5 <- 1 - pbinom(4, size = vacunadas, prob = prob_infeccion_general)
```

```
cat("En 200 personas vacunadas, el número de infecciones sigue una distribución:\n")
```

```
## En 200 personas vacunadas, el número de infecciones sigue una distribución:
```

```
cat("X ~ Binomial(n =", vacunadas, ", p =", round(prob_infeccion_general, 4), ")\n\n")
```

```
## X ~ Binomial(n = 200 , p = 0.0314 )
```

```
cat("P(X >= 5) = 1 - P(X <= 4) =", round(prob_ge_5, 6), "\n")
```

```
## P(X >= 5) = 1 - P(X <= 4) = 0.755088
```

```
cat("P(X >= 5) =", round(prob_ge_5 * 100, 2), "%\n\n")
```

```
## P(X >= 5) = 75.51 %
```

Calcularemos ahora a través de 10,000 simulaciones

```
# Fijamos la semilla para su reproducción
set.seed(17)
n_sim <- 10000

# Simulación
simulaciones_a <- replicate(n_sim, {
  # Para cada persona, simularemos si se infecta
  infecciones <- rbinom(vacunadas, size = 1, prob = prob_infeccion_general)
  sum(infecciones)
})
```

```
# Probabilidad simulada de que al menos 5 se infecten
prob_simulada_5 <- mean(simulaciones_a >= 5)

cat("Número de infecciones por simulación (resumen):\n")
```

```
## Número de infecciones por simulación (resumen):
```

```
print(summary(simulaciones_a))
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.000   5.000   6.000   6.267   8.000  18.000
```

Veámos los resultados obtenidos:

```
cat("P(X >= 5) mediante simulación:", round(prob_simulada_5, 6), "\n")
```

```
## P(X >= 5) mediante simulación: 0.7548
```

```
cat("P(X >= 5) =", round(prob_simulada_5 * 100, 2), "%\n\n")
```

```
## P(X >= 5) = 75.48 %
```

Comparamos los resultados y comentaremos los resultados obtenidos.

```
cat("--- COMPARACIÓN DE RESULTADOS ---\n\n")
```

```
## --- COMPARACIÓN DE RESULTADOS ---
```

```
cat("Método exacto:      ", round(prob_ge_5, 6),
    "(", round(prob_ge_5 * 100, 2), "%)\n")
```

```
## Método exacto:      0.755088 ( 75.51 %)
```

```
cat("Método simulación: ", round(prob_simulada_5, 6),
    "(", round(prob_simulada_5 * 100, 2), "%)\n")
```

```
## Método simulación:  0.7548 ( 75.48 %)
```

```
cat("Diferencia absoluta:", round(abs(prob_ge_5 - prob_simulada_5), 6), "\n\n")
```

```
## Diferencia absoluta: 0.000288
```

Tanto el cálculo exacto que las simulaciones nos llevan a una diferencia absoluta de casi cercano al cero. Por lo que con 10,000 simulaciones el valor esperado es exactamente igual al valor mediante cálculo exacto a través de la binomial.

5.2

Pasamos al siguiente escenario de 50 personas que no desarrollan inmunidad y calcular la probabilidad de que ninguna se infecte. Finalmente seleccionaremos 150 que sí desarrollan inmunidad y calculando la probabilidad de que al menos 3 se infecten. Repiteremos a través de cálculo exactos y a través de las simulaciones visto anteriormente.

Pasamos al primer **Escenario sin inmunidad**

```
n_sin_inmunidad <- 50

# A través del cálculo exacto
prob_exacta_ninguna_sin <- dbinom(0, size = n_sin_inmunidad,
                                prob = prob_infeccion_sin_inmunidad)

cat("CÁLCULO EXACTO:\n")
```

```
## CÁLCULO EXACTO:
```

```
cat("X ~ Binomial(n =", n_sin_inmunidad, ", p =", prob_infeccion_sin_inmunidad, ")\n")
```

```
## X ~ Binomial(n = 50 , p = 0.4 )
```

```
cat("P(X = 0) ", format(prob_exacta_ninguna_sin, scientific = TRUE), "\n")
```

```
## P(X = 0) 8.082813e-12
```

```
cat("P(X = 0) ", round(prob_exacta_ninguna_sin * 100, 10), "%\n\n")
```

```
## P(X = 0) 8e-10 %
```

Pasamos ahora a las simulaciones:

```
# fijamos la semilla
set.seed(17)
sim_sin_inmunidad <- replicate(n_sim, {
  infecciones <- rbinom(n_sin_inmunidad, size = 1, prob = prob_infeccion_sin_inmunidad)
  sum(infecciones)
})

prob_simulada_ninguna_sin <- mean(sim_sin_inmunidad == 0)

cat("SIMULACIÓN (10000 réplicas):\n")
```

```
## SIMULACIÓN (10000 réplicas):
```

```
cat("P(X = 0) ", format(prob_simulada_ninguna_sin, scientific = TRUE), "\n")
```

```
## P(X = 0) 0e+00
```

```
cat("P(X = 0) ", round(prob_simulada_ninguna_sin * 100, 10), "%\n\n")
```

```
## P(X = 0) 0 %
```

```
cat("Número de simulaciones donde ninguna se infectó:",  
    sum(sim_sin_inmunidad == 0), "de", n_sim, "\n\n")
```

```
## Número de simulaciones donde ninguna se infectó: 0 de 10000
```

La probabilidad de que ninguna persona se infecte es extremadamente baja, es prácticamente cero. Esto es esperado porque sin inmunidad, cada persona tiene un 40% de probabilidad de infectarse, y con 50 personas, es casi seguro que al menos algunas se infectarán. Si vemos el número esperado de infecciones es dado de : $E[X] = 50 \times 0.4 = 20$ personas

Pasamos al grupo de 150 con inmunidad y queremos ver la probabilidad que 3 se infecten. Primero con el cálculo exacto.

```
n_con_inmunidad <- 150
```

```
#  $X \sim \text{Binomial}(n = 150, p = 0.02)$ 
```

```
#  $P(X \geq 3) = 1 - P(X \leq 2)$ 
```

```
prob_exacta_3 <- 1 - pbinom(2, size = n_con_inmunidad,  
                           prob = prob_infeccion_con_inmunidad)
```

```
cat("CÁLCULO EXACTO:\n")
```

```
## CÁLCULO EXACTO:
```

```
cat("X ~ Binomial(n =", n_con_inmunidad, ", p =", prob_infeccion_con_inmunidad, ")\n")
```

```
## X ~ Binomial(n = 150 , p = 0.02 )
```

```
cat("P(X >= 3) = 1 - P(X <= 2) =", round(prob_exacta_3, 6), "\n")
```

```
## P(X >= 3) = 1 - P(X <= 2) = 0.579074
```

```
cat("P(X >= 3) =", round(prob_exacta_3 * 100, 2), "%\n\n")
```

```
## P(X >= 3) = 57.91 %
```

Pasamos a la simulación.

```
# Fijamos la semilla
set.seed(17)
sim_con_inmunidad <- replicate(n_sim, {
  infecciones <- rbinom(n_con_inmunidad, size = 1, prob = prob_infeccion_con_inmunidad)
  sum(infecciones)
})

prob_sim_3 <- mean(sim_con_inmunidad >= 3)

cat("SIMULACIÓN (10000 réplicas):\n")
```

```
## SIMULACIÓN (10000 réplicas):
```

```
cat("P(X >= 3) =", round(prob_sim_3, 6), "\n")
```

```
## P(X >= 3) = 0.5843
```

```
cat("P(X >= 3) =", round(prob_sim_3 * 100, 2), "%\n\n")
```

```
## P(X >= 3) = 58.43 %
```

Observamos unas ciertas diferencias, y si probamos a detallar las simulaciones para comprobar a partir de cuantos tengamos un valor de infectos:

```
cat("Distribución de infecciones en las simulaciones:\n")
```

```
## Distribución de infecciones en las simulaciones:
```

```
print(table(sim_con_inmunidad))
```

```
## sim_con_inmunidad
##    0    1    2    3    4    5    6    7    8    9   10   11
## 539 1431 2187 2254 1770 1038  461  213   76   23    7    1
```

```
cat("\n")
```

Comienza a decrecer a partir de 11 infectos,

```
cat("--- COMPARACIÓN DE RESULTADOS ---\n\n")
```

```
## --- COMPARACIÓN DE RESULTADOS ---
```

```
cat("Método exacto:      ", round(prob_exacta_3, 6),
    "(", round(prob_exacta_3 * 100, 2), "%)\n")
```

```
## Método exacto:      0.579074 ( 57.91 %)
```

```
cat("Método simulación: ", round(prob_sim_3, 6),
    "(", round(prob_sim_3 * 100, 2), "%)\n")
```

```
## Método simulación: 0.5843 ( 58.43 %)
```

```
cat("Diferencia absoluta:", round(abs(prob_exacta_3 - prob_sim_3), 6), "\n\n")
```

```
## Diferencia absoluta: 0.005226
```

La diferencia se aprecia mínima de casi 0.005 puntos. Por lo que hay una probabilidad del 57.9 % de que al menos 3 personas con inmunidad se infecten. Aunque la tasa de infección es baja (2%), con 150 personas, el número esperado de infecciones es: $E[X] = 150 \times 0.02 = 3$ personas. Es decir la inmunidad reduce drásticamente la tasa de infección (de 40% a 2%), pero incluso en grupos con inmunidad, pueden ocurrir algunas infecciones.

5.3

Graficamos los grupos de personas con inmunidad y sin inmunidad.

```
# Grupo sin inmunidad (50 personas)
x_sin <- 0:n_sin_inmunidad
prob_sin <- dbinom(x_sin, size = n_sin_inmunidad, prob = prob_infeccion_sin_inmunidad)

# Grupo con inmunidad (150 personas)
x_con <- 0:20
prob_con <- dbinom(x_con, size = n_con_inmunidad, prob = prob_infeccion_con_inmunidad)

# Crearemos los dataframes
df_sin <- data.frame(
  infecciones = x_sin,
  probabilidad = prob_sin,
  grupo = "Sin inmunidad (n=50, p=0.40)"
)

df_con <- data.frame(
  infecciones = x_con,
  probabilidad = prob_con,
  grupo = "Con inmunidad (n=150, p=0.02)"
)

# combinaremos para crear el para gráfico comparativo
df_combinado <- rbind(df_sin, df_con)

# Creamos el gráfico de de barras comparativo
grafico_distribuciones <- ggplot(df_combinado,
                                aes(x = infecciones, y = probabilidad, fill = grupo)) +
  geom_bar(stat = "identity", position = "identity", alpha = 0.7) +
  facet_wrap(~grupo, scales = "free", ncol = 1) +
  labs(title = "Distribución de probabilidad del número de infecciones",
```

```

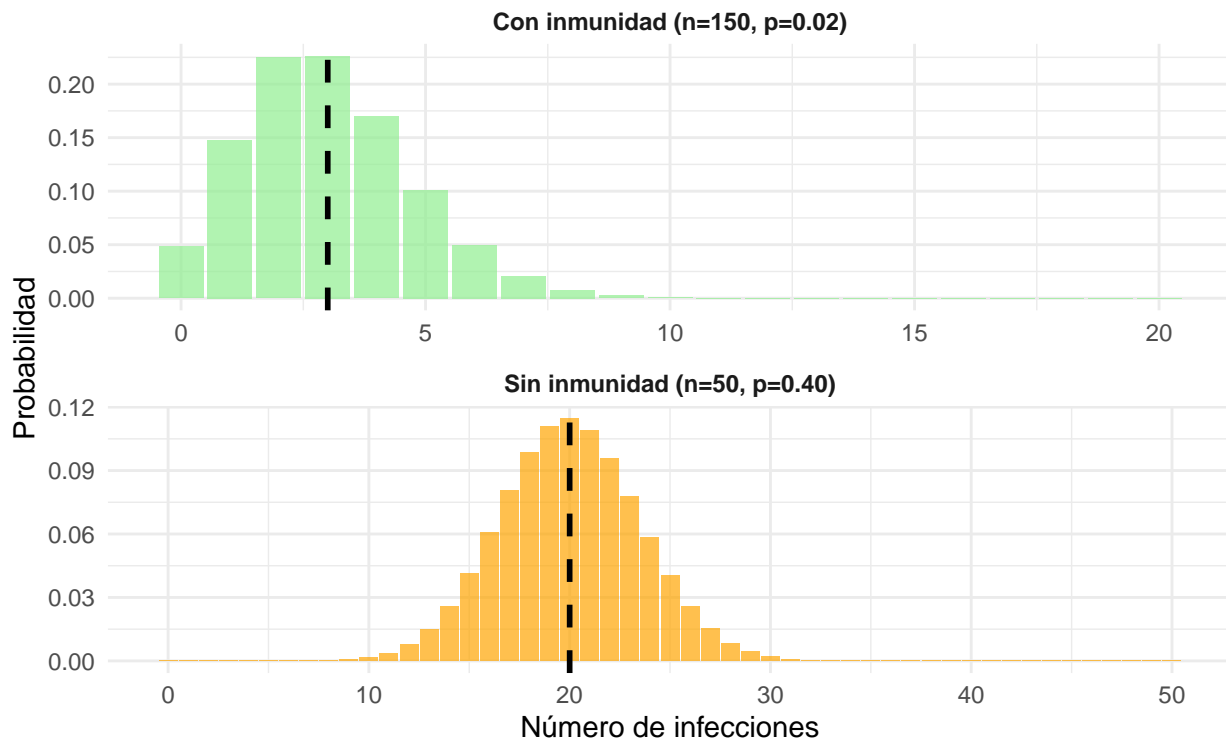
    subtitle = "Comparación entre grupos con y sin inmunidad",
    x = "Número de infecciones",
    y = "Probabilidad",
    fill = "Grupo") +
theme_minimal() +
theme(plot.title = element_text(hjust = 0.5, face = "bold"),
      plot.subtitle = element_text(hjust = 0.5),
      legend.position = "none",
      strip.text = element_text(face = "bold")) +
scale_fill_manual(values = c("Sin inmunidad (n=50, p=0.40)" = "orange",
                             "Con inmunidad (n=150, p=0.02)" = "lightgreen")) +
geom_vline(data = data.frame(grupo = c("Sin inmunidad (n=50, p=0.40)",
                                       "Con inmunidad (n=150, p=0.02)"),
                             media = c(n_sin_inmunidad * prob_infeccion_sin_inmunidad,
                                       n_con_inmunidad * prob_infeccion_con_inmunidad)),
          aes(xintercept = media), linetype = "dashed", color = "black", size = 1)

print(grafico_distribuciones)

```

Distribución de probabilidad del número de infecciones

Comparación entre grupos con y sin inmunidad



Podemos observar las diferencias de los gráficos de comparativa:

Para el grupo SIN inmunidad: - Media: $E[X] = 20$ infecciones - La distribución está centrada alrededor de 20 infecciones - Forma aproximadamente simétrica (cercana a normal por n grande y p moderado) - Desviación estándar: $\sigma = 3.46$

Para el grupo CON inmunidad: - Media: $E[X] = 3$ infecciones - La distribución está centrada

alrededor de 3 infecciones - Forma asimétrica positiva (cola hacia la derecha) - Desviación estándar: $\sigma = 1.71$

El grupo sin inmunidad tiene mucha mayor dispersión debido a la mayor probabilidad de infección ($p = 0.40$). El grupo con inmunidad tiene baja dispersión, con la mayoría de la probabilidad concentrada en valores bajos (0-10 infecciones)

Respecto al rango de valores obtenidos, el grupo Sin inmunidad tiene valores posibles de 0 a 50, pero la mayoría de la probabilidad está entre 10 y 30 infecciones. Mientras que el grupo con inmunidad, aunque sea mayor $n=150$, la mayoría de la probabilidad está concentrada en 0-10 infecciones debido al bajo $p=0.02$

Por lo que, el grupo sin inmunidad es altamente vulnerable: es casi seguro que habrá un número significativo de infecciones (15-25 personas). El grupo con inmunidad está mucho mejor protegido: lo más probable es tener solo 2-4 infecciones incluso con 150 personas. La vacuna reduce el número esperado de infecciones de 20/50 (40%) a 3/150 (2%), una reducción del 95% en la tasa de infección.

Esto tiene unas implicaciones epidemiológicas ya que la inmunidad de grupo es crucial: aunque un 3% no desarrolle inmunidad, el 97% que sí la desarrolla reduce dramáticamente la transmisión. Esto protege indirectamente a quienes no pueden desarrollar inmunidad. La diferencia en las distribuciones demuestra la efectividad de la vacuna

5.4

En este último apartado, hagamos una pequeña reflexión sobre estrategias de prevención. Es mejor reforzar inmunidad individual respecto a la reducción de la exposición general. Si a partir de los resultados obtenidos, podemos analizar la eficacia de diferentes estrategias de prevención.

- Un 3% de vacunados no desarrolla inmunidad (inevitable con tecnología actual)
- Las personas sin inmunidad tienen 40% de probabilidad de infectarse
- Las personas con inmunidad tienen solo 2% de probabilidad de infectarse
- La vacuna reduce la tasa de infección en un 95% (de 40% a 2%)
- En 200 personas vacunadas, esperamos ~ 6.28 infecciones
 - De las cuales ~ 2.4 provienen del 3% sin inmunidad
 - Y ~ 3.88 provienen del 97% con inmunidad

Si el objetivo es lograr que el 100% desarrolle inmunidad efectiva tenemos:

Ventajas: * Protección directa a nivel individual * Reduce drásticamente las infecciones en vacunados * Efecto duradero si la inmunidad es sostenida * Permite cierta normalización de actividades

Desafíos: * Límite biológico: siempre habrá individuos que no respondan (inmunocomprometidos, variabilidad genética, edad avanzada) * Requiere investigación y desarrollo de nuevas formulaciones de vacunas * Costoso y requiere tiempo * No protege a quienes no pueden vacunarse * La inmunidad puede disminuir con el tiempo (requiere refuerzos)

Impacto estimado: Si se lograra el 100% de inmunidad efectiva en lugar de 97%: * Infecciones esperadas: de 6.28 a 4 (- 2.28) * Reducción relativa: $\sim 36.3\%$

Si el objetivo es disminuir las oportunidades de contacto con el virus:

Ventajas: * Protege TANTO a personas con inmunidad como sin inmunidad * Reduce la probabilidad de infección de 40% a X% en no inmunizados y de 2% a Y% en inmunizados (donde $X < 40\%$ y $Y < 2\%$) * Efecto inmediato (no requiere desarrollo tecnológico) * Protege a poblaciones vulnerables que no pueden vacunarse * Reduce la circulación viral y aparición de variantes

Desafíos: * Requiere cambios de comportamiento colectivo sostenidos * Impacto socioeconómico (restricciones de movilidad, trabajo remoto) * Difícil de mantener a largo plazo (fatiga pandémica) * Puede generar resistencia social

Impacto estimado: Si reducimos la exposición a la mitad ($P(\text{inf}|\text{sin inm}) = 20\%$, $P(\text{inf}|\text{con inm}) = 1\%$):
* Nueva $P(\text{infección}) = 0.0157$ vs 0.0314 actual * Infecciones esperadas: de 6.28 a 3.14 (- 3.14) * Reducción relativa: $\sim 50\%$

Por cada estrategia veámos la eficacia de forma aislada:

De un lado reforzar la inmunidad (de 97% a 100% con inmunidad): * Reduce infecciones en $\sim 36.3\%$ * Principalmente beneficia al 3% que actualmente no desarrolla inmunidad

De otro lado reducir exposición a la mitad: * Reduce infecciones en $\sim 50\%$ * Beneficia a TODA la población, independientemente del estado de inmunidad

A la pregunta ¿Es mejor centrarse en que más gente tenga inmunidad perfecta o en reducir el contacto con el virus? No hay una correcta y mi respuesta corta sería las dos cosas, pero si hay que priorizar, reducir la exposición gana.

Vacunarse es súper importante y es la base de todo, pero no es suficiente por sí solo. Incluso la gente vacunada puede contagiarse (aunque sea poco probable). Por eso, combinar vacunas con medidas tipo “evitar aglomeraciones” o “ventilar bien” es lo más efectivo.

Es como conducir: el cinturón de seguridad (vacuna) es esencial, pero también necesitas respetar los límites de velocidad y mantener distancia de seguridad (reducir exposición). Las dos cosas juntas te mantienen a salvo.