Tarea 2

Sara

2024-02-06

```
library(ggplot2)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v lubridate 1.9.3     v tibble    3.2.1
v purrr     1.0.2     v tidyr     1.3.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

```
library(patchwork)
```

Modelo poisson

1. Estimando una media Poisson usando una inicial discreta. Supongan que son dueños de una compañía de transporte con una flota grande de camiones. Las descomposturas ocurren aleatoriamente en el tiempo y supóngase que el número de descomposturas durante un intervalo de t días sigue un distribución Poisson con media λt. El parámetro λ es la tasa de descompostura diaria. Los posibles valores para λ son 0.5, 1, 1.5, 2, 2.5 y 3, con respectivas probabilidades 0.1, 0.2, 0.3, 0.2, 0.15 y 0.05. Si uno observa y descomposturas, entonces la probabilidad posterior de λ es proporcional a

$$g(\lambda)exp(-t\lambda)(t\lambda)y,$$

donde g es la distribución inicial.

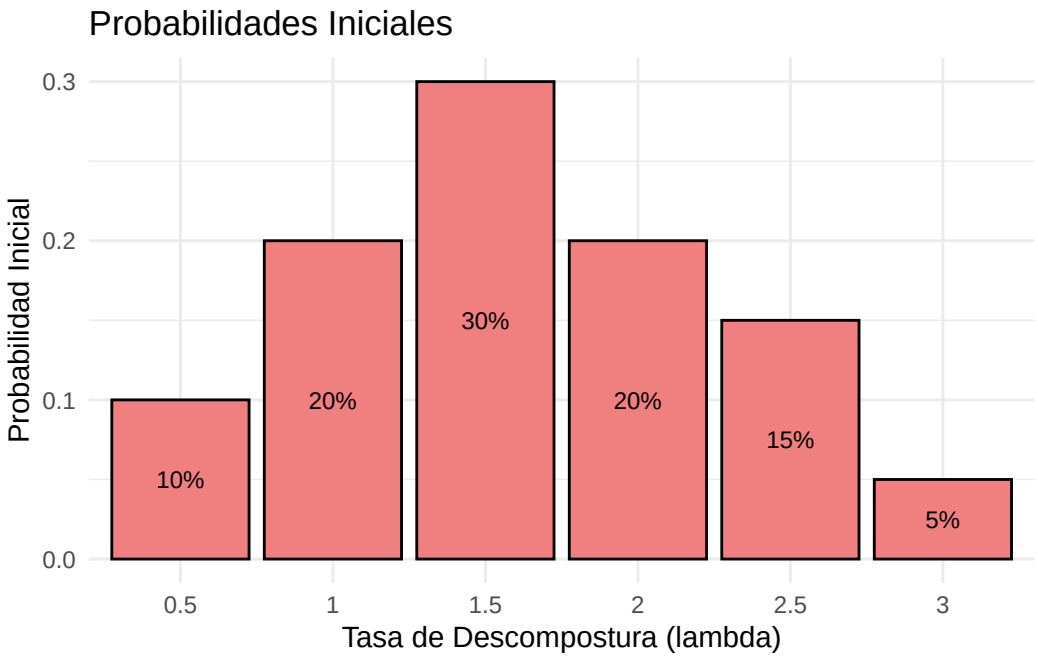
a. Si 12 camiones se descomponen en un periodo de 6 días, encontrar la probabilidad posterior para las diferentes tasas.

Datos proporcionados

```
lambda_values <- c(0.5, 1, 1.5, 2, 2.5, 3)
probabilidades_iniciales <- c(0.1, 0.2, 0.3, 0.2, 0.15, 0.05)
t <- 6  # días
y <- 12  # descomposturas observadas
```

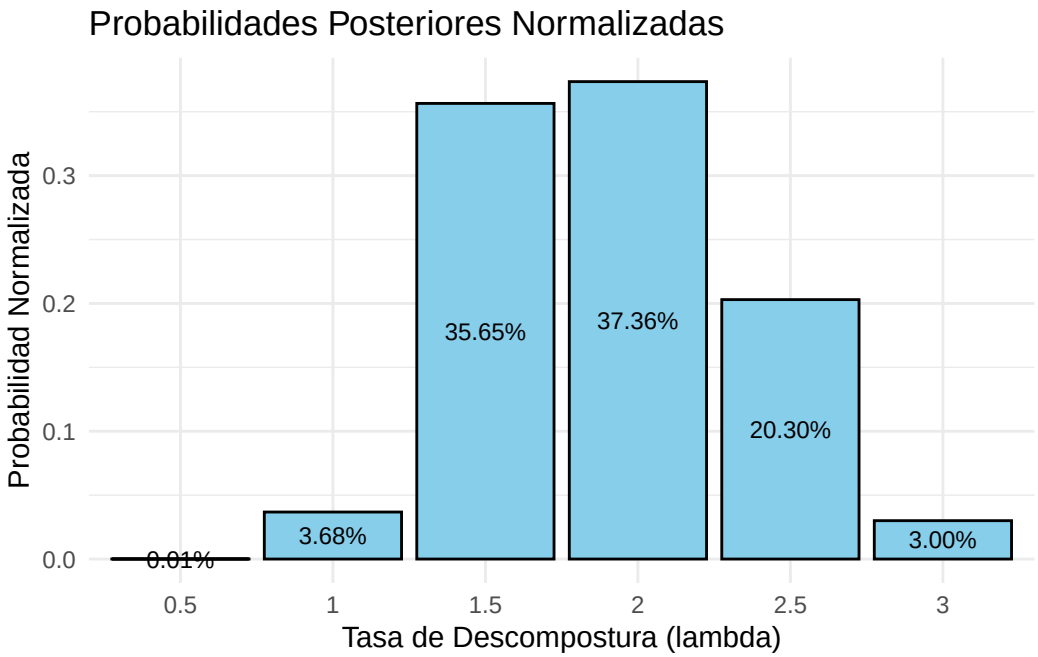
Probabilidades iniciales

```
iniciales <- data.frame(lambda = lambda_values, prob_inicial = probabilidades_iniciales)

# Creamos un gráfico de barras para visualizar las probabilidades iniciales
ggplot(iniciales, aes(x = as.factor(lambda), y = prob_inicial,label = scales::percent(prob
  geom_bar(stat = "identity", fill = "lightcoral", color = "black") +
  geom_text(position = position_stack(vjust = 0.5), size = 3) +
  labs(title = "Probabilidades Iniciales",
       x = "Tasa de Descompostura (lambda)",
       y = "Probabilidad Inicial") +
  theme_minimal()
```



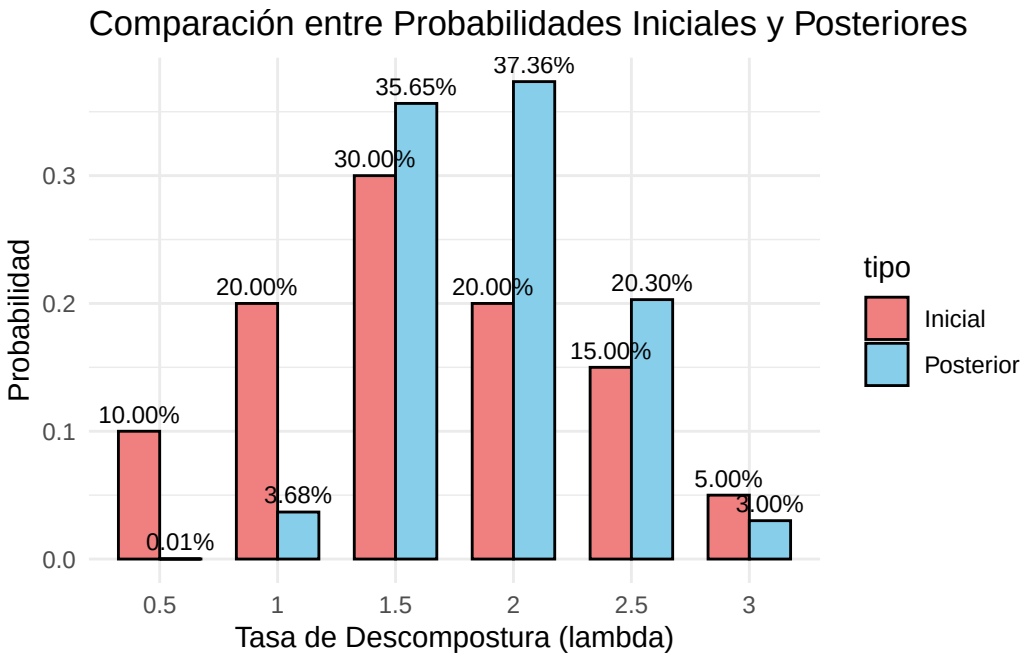
Cálculo de probabilidades posteriores

```r
# Calculamos las probabilidades posteriores proporcionales
prob_posterior_proporcional <- probabilidades_iniciales * exp(-t * lambda_values) * (t * l

# Normalizamos las probabilidades
prob_posterior_normalizadas <- prob_posterior_proporcional / sum(prob_posterior_proporcion

# Creamos un data frame con los resultados
resultados <- data.frame(lambda = lambda_values, prob_posterior = prob_posterior_normaliza

# Crea un gráfico de barras para visualizar las probabilidades normalizadas
ggplot(resultados, aes(x = as.factor(lambda), y = prob_posterior,label = scales::percent(p
  geom_bar(stat = "identity", fill = "skyblue", color = "black") +
  geom_text(position = position_stack(vjust = 0.5), size = 3) +
  labs(title = "Probabilidades Posteriores Normalizadas",
       x = "Tasa de Descompostura (lambda)",
       y = "Probabilidad Normalizada") +
  theme_minimal()
```



Comparativo de probabilidades iniciales y posteriores

```
# Creamos data frames con los resultados iniciales y posteriores
iniciales <- data.frame(lambda = lambda_values, prob = probabilidades_iniciales, tipo = "I
posteriores <- data.frame(lambda = lambda_values, prob = prob_posterior_normalizadas, tipo


# Combina los data frames para la visualización comparativa
resultados_combinados <- rbind(iniciales, posteriores)


ggplot(resultados_combinados, aes(x = as.factor(lambda), y = prob, fill = tipo, label = sc
  geom_bar(stat = "identity", position = "dodge", color = "black", width = 0.7) +
  geom_text(position = position_dodge(width = 0.7), vjust = -0.5, size = 3) +
  labs(title = "Comparación entre Probabilidades Iniciales y Posteriores",
       x = "Tasa de Descompostura (lambda)",
       y = "Probabilidad") +
  scale_fill_manual(values = c("lightcoral", "skyblue")) +
  theme_minimal()
```



b. Encontrar la probabilidad de que no haya descomposturas durante la siguiente semana.

Hint: Si la tasa es λ, la probabilidad condicional de no descomposturas durante un periodo de 7 días está dado por $\exp(-7\lambda)$. Se puede calcular esta probabilidad predictiva multiplicando la lista de probabilidades condicionales por las probabilidades posteriores de λ y encontrando la suma de los productos

Cálculo de probabilidad mediante la función de masa de probabilidad de la distribución poisson.

La probabilidad de que no haya descomposturas ($P(X = 0)$) en un periodo de tiempo dado se puede calcular usando la función de masa de probabilidad de la distribución Poisson:

$P(X = 0) = \frac{!e^{-\lambda} \cdot \lambda}{0}$

Donde X es la variable aleatoria que representa el número de descomposturas, λ es la tasa de descompostura diaria y e es la base del logaritmo natural.

La probabilidad de no tener descomposturas en una semana (7 días) sería la probabilidad de no tener descomposturas en un día $P(X = 0)$) elevada a la potencia de 7 (días):

$P(No\_descomposturas\_en\_\_\_7\_das) = [P(X = 0)]^7$

```r
# Probabilidad predictiva condicional de no tener descomposturas en 7 días
prob_predictiva_condicional_7_dias <- exp(-7 * lambda_values)

# Probabilidad ponderada por las probabilidades posteriores
prob_no_descomposturas_7_dias_ponderada <- sum(prob_predictiva_condicional_7_dias * prob_p

cat("Probabilidad de no tener descomposturas en 7 días:", prob_no_descomposturas_7_dias_po
```

```
Probabilidad de no tener descomposturas en 7 días: 4.640932e-05
```

Probabilidad de no tener descomposturas en 7 días: 4.640932e-05

2. **Estimando una proporción y predicción de una muestra futura.** Un estudio reporta sobre los efectos de largo plazo de exposición a bajas dosis de plomo en niños. Los investigadores analizaron el contenido de plomo en la caída de los dientes de leche. De los niños cuyos dientes tienen un contenido de plomo mayor que 22.22 ppm, 22 eventualmente se graduaron de la preparatoria y 7 no. Supongan que su densidad inicial para p, la proporción de todos tales niños que se graduaron de preparatoria es $beta(1, 1)$, y posterior es $beta(23, 8)$.

a. Encontrar un intervalo estimado de 90 % para p.

Dado que la distribución posterior es $beta(23, 8)$, podemos utilizar cuantiles de esta distribución para obtener el intervalo de credibilidad.

El intervalo de credibilidad del 90% para p se obtiene utilizando los cuantiles de la distribución $beta(23, 8)$.

```r
# Parámetros de la distribución posterior
alpha_posterior <- 23
beta_posterior <- 8

# Intervalo de credibilidad del 90% para p
intervalo_credibilidad <- qbeta(c(0.05, 0.95), alpha_posterior, beta_posterior)

cat("Intervalo de 90% para p:", intervalo_credibilidad, "\n")
```

```
Intervalo de 90% para p: 0.6060526 0.8598149
```

b. Probabilidad de que p exceda 0.6 Para encontrar la probabilidad de que p exceda 0.6,utilizamos la función de densidad de probabilidad acumulativa (CDF) de la distribución $beta(23, 8)$.

```r
# Probabilidad de que p exceda 0.6
prob_exceder_0.6 <- 1 - pbeta(0.6, alpha_posterior, beta_posterior)

cat("Probabilidad de que p exceda 0.6:", prob_exceder_0.6, "\n")
```

```
Probabilidad de que p exceda 0.6: 0.9564759
```

3. **Estimando una media normal posterior con una inicial discreta.** Supongamos que están interesados en estimar el promedio de caida de lluvia por año μ en (cm) para una ciudad grande del Centro de México. Supongan que la caída anual individual $y1, ..., yn$ son obtenidas de una población que se supone $N(\mu, 100)$. Antes de recolectar los datos, supongan que creen que la lluvia media puede estar en los siguiente valores con respectivas probabilidades | | 20 | 30 | 40 | 50 | 60 | 70 | | g( ) | 0.1 | 0.15 | 0.25 | 0.25 | 0.15 | 0.1 |

a. Supongan que se observan los totales de caída de lluvia 38.6, 42.4, 57.5, 40.5, 51.7, 67.1, 33.4, 60.9, 64.1, 40.1, 40.7 y 6.4. Calcular la media.

```r
# Datos de caída de lluvia
datos_lluvia <- c(38.6, 42.4, 57.5, 40.5, 51.7, 67.1, 33.4, 60.9, 64.1, 40.1, 40.7, 6.4)

# Calcular la media muestral
media_lluvia <- mean(datos_lluvia)

# Imprimir el resultado
cat("La media de la caída de lluvia es:", media_lluvia, "centímetros\n")
```

```
La media de la caída de lluvia es: 45.28333 centímetros
```

b. Calcular la función de verosimilitud utilizando como estadística suficiente la media $\bar{y}$.

La función de verosimilitud es una medida de la probabilidad de observar los datos dados ciertos valores de los parámetros. En este caso, la función de verosimilitud se refiere a la probabilidad de observar los totales de caída de lluvia y dados distintos valores de la media μ, asumiendo una distribución normal con varianza conocida ((^2 = 100)).

La función de verosimilitud ($L(\mu|\mathbf{y})$) se puede expresar para una muestra de (n) observaciones como el producto de las densidades de probabilidad de cada observación:

$$L(\mu|\mathbf{y}) = \prod_{i=1}^{n} f(y_i|\mu)$$

Dado que estamos asumiendo una distribución normal, la función de densidad de probabilidad para una observación individual (y_i) es:

$$f(y_i|\mu) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i-\mu)^2}{2\sigma^2}}$$

Sustituimos esta expresión en la función de verosimilitud:

$$L(\mu|\mathbf{y}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i-\mu)^2}{2\sigma^2}}$$

Dado que la varianza es conocida ((^2 = 100)), podemos simplificar aún más:

$$L(\mu|\mathbf{y}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi \cdot 100}} e^{-\frac{(y_i-\mu)^2}{2 \cdot 100}}$$

```
# Parámetro de la distribución (media poblacional)
mu <- seq(20, 70, by = 1)  # Valores posibles de la media

# Función de verosimilitud
verosimilitud <- prod(dnorm(datos_lluvia, mean = mu, sd = sqrt(100)))

# Imprimir el resultado
cat("La función de verosimilitud para y barra es:", verosimilitud, "\n")
```

```
La función de verosimilitud para y barra es: 7.042525e-126
```

*Calcular las probabilidades posteriores para μ.

Para calcular las probabilidades posteriores para ( ) utilizando el enfoque bayesiano, necesitamos combinar la información de la función de verosimilitud y la distribución prior. Las probabilidades posteriores se obtienen aplicando el Teorema de Bayes.

La forma general de la distribución posterior $f(\mu|\mathbf{y})$ es proporcional al producto de la función de verosimilitud y la distribución prior:

$$f(\mu|\mathbf{y}) \propto g(\mu) \cdot f(\mathbf{y}|\mu)$$

Donde: - $g(\mu)$ es la distribución prior discreta para ( ) con probabilidades (0.1, 0.15, 0.25, 0.25, 0.15, 0.1), y - $f(\mathbf{y}|\mu)$ es la función de verosimilitud calculada a partir de los datos de caída de lluvia observados.

En este caso, podemos calcular las probabilidades posteriores para ( ) para cada uno de los valores posibles utilizando las probabilidades de la prior y la función de verosimilitud.

Las probabilidades están normalizadas para sumar 1.

```
# Parámetros prior
data <- data.frame(mu=c(20, 30, 40, 50, 60, 70),
                   prob_prior=c(0.1, 0.15, 0.25, 0.25, 0.15, 0.1))

# Probabilidades posteriores
data <- data %>%
  mutate(verosim=exp(-12*((media_lluvia - mu)^2)/(2*100)),
         dist_post=verosim*prob_prior,
         prob_post=dist_post/sum(dist_post))

#Resultado
res <- data[, c("mu", "prob_post")]
res
```

```
  mu    prob_post
1 20 1.954479e-17
2 30 1.091063e-06
3 40 4.158078e-01
4 50 5.841881e-01
5 60 3.025731e-06
6 70 1.069871e-16
```

*Encontrar un intervalo de probabilidad de 80% para μ.

```
# Probabilidad acumulada
data <- data %>%
  mutate(cum_post=cumsum(prob_post))

mu_10 <- data$mu[which.max(data$cum_post >= 0.1)]
mu_90 <- data$mu[which.max(data$cum_post >= 0.9)]

# Imprimir el intervalo de probabilidad del 80%
cat("Intervalo de probabilidad del 80% para mu:", mu_10, "a", mu_90, "\n")
```

```
Intervalo de probabilidad del 80% para mu: 40 a 50
```

Modelo Cauchy

4. Modelo muestral Cauchy. Supongan que se observa una muestra aleatoria $y_1, ..., y_n$ de una densidad Cauchy con parámetro de localización θ y parámetro de escala 1. Si una inicial uniforme se considera para θ, entonces la densidad posterior, ¿cuál es? Supongan que se observan los datos $0, 10, 9, 8, 11, 3, 3, 8, 8, 11$.

Con base en el Teorema de Bayes: la densidad posterior es proporcional al producto de la función de verosimilitud y la densidad a priori.

La función de verosimilitud es la densidad de una distribución Cauchy con parámetro de localización θ y parámetro de escala 1 :

$f(yi \ \mid \theta) = \frac{1}{\pi(1+(yi \ -\theta)2)}$

La densidad a priori es una uniforme para θ, por lo que $\pi(\theta) = constante$.

Por lo tanto la densidad posterior es:

$Posterior(\theta \mid y1 \ , ..., yn \ ) \propto \prod_{i=1}^{n} \ f(yi \ \mid \theta) \cdot \pi(\theta)$

Por simplicidad, podemos trabajar con el logaritmo de la densidad posterior para facilitar los cálculos.

$log(Posterior(\theta \mid y1 \ , ..., yn \ )) \propto \sum_{i=1}^{n} \ log(f(yi \ \mid \theta)) + log(\pi(\theta))$

Por lo tanto:

$log(Posterior(\theta \mid y1 \ , ..., yn \ )) \propto \sum_{i=1}^{10} \ log(\frac{1}{\pi(1+(yi \ -\theta)2)} \ ) + constante$

a. Calcula un grid para θ de -2 a 12 en pasos de 0.1

b. Calcula la densidad posterior en este grid.

c. Grafica la densidad y comenten sobre sus características principales.

Datos observados

```
datos <- c(0, 10, 9, 8, 11, 3, 3, 8, 8, 11)
```

Función de densidad de la distribución Cauchy

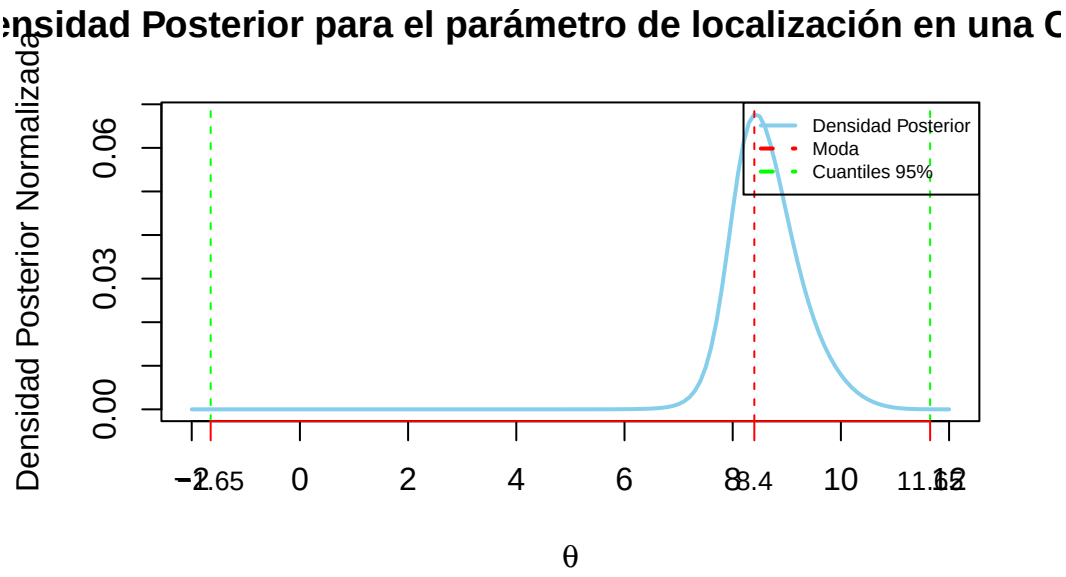
```
densidad_cauchy <- function(theta, datos) {
  sum(log(1 / (pi * (1 + (datos - theta)^2))))
}
```

Definimos una secuencia de valores para theta

```
theta_valores <- seq(-2, 12, by = 0.1)
```

Cálculo de la posterior

```
# Calculamos la densidad posterior no normalizada
posterior_no_normalizada <- exp(sapply(theta_valores, function(theta) densidad_cauchy(thet

# Normalizamos la densidad posterior
posterior_normalizada <- posterior_no_normalizada / sum(posterior_no_normalizada)

#Moda posterior (donde la densidad posterior es máxima) y cuantiles
moda <- theta_valores[which.max(posterior_normalizada)]
cuantiles_95 <- quantile(theta_valores, c(0.025, 0.975))

# Graficamos la densidad posterior
plot(theta_valores, posterior_normalizada, type = 'l', col = 'skyblue', lwd = 2,
     xlab = expression(theta), ylab = 'Densidad Posterior Normalizada',
     main = 'Densidad Posterior para el parámetro de localización en una Cauchy')

# Agregamos líneas para destacar la moda y los cuantiles
abline(v = moda, col = 'red', lty = 2)
abline(v = cuantiles_95, col = 'green', lty = 2)

# Etiquetas para la moda y los cuantiles a lo largo del eje x
axis(1, at = c(moda, cuantiles_95[1], cuantiles_95[2]), labels = c(round(moda, 2), round(c

# Leyenda en una ubicación más adecuada
legend("topright", legend = c("Densidad Posterior", "Moda", "Cuantiles 95%"), col = c('sky
```



Entre las características podemos observar:

1. Al tratarse de una previa uniforme, su contribución a la densidad posterior será constante lo que significa que la posterior estará dominada principalmente por la verosimilitud, en este caso la distribución Cauchy.
2. La moda de los datos se encuentra en $\theta = 8.4$
3. Para estimar la variabilidad o incertidumbre podemos basarnos en un intervalo de credibilidad del 95% (-1.65 y 11.65)

d. Calcula la media posterior y desviación estándar posterior.

Para calcular la media posterior y la desviación estándar posterior utilizamos la densidad posterior. La media posterior se calcula como la integral del producto de la variable y la densidad posterior. La varianza es equivalente a la suma de los cuadrados de la resta de los valores de la variable - la media posterior y estos multiplicados por la densidad posterior.

```
# Media posterior
media_posterior <- sum(theta_valores * posterior_normalizada)

# Varianza posterior
varianza_posterior <- sum((theta_valores - media_posterior)^2 * posterior_normalizada)

# Desviación estándar posterior
```

```r
desviacion_estandar_posterior <- sqrt(varianza_posterior)

# Imprimimos los resultados
cat("Media Posterior:", round(media_posterior, 2), "\n")
```

```
Media Posterior: 8.63
```

```r
cat("Desviación Estándar Posterior:", round(desviacion_estandar_posterior, 2), "\n")
```

```
Desviación Estándar Posterior: 0.65
```

5. **Robustez Bayesiana.** Supongan que están a punto de lanzar una moneda que creen que es honesta. Si p denota la probabilidad de obtener sol, entonces su mejor creencia es que $p = 0.5$

Adicionalmente, creen que es altamente probable que la moneda sea cercana a honesta, lo que cuantifican como $P(0.44 \leq p \leq 0.56) = 0.9$. Consideren las siguientes dos iniciales para p:

P1 $p \sim beta(100, 100)$

P2 $p \sim 0.9beta(500, 500) + 0.1beta(1, 1)$

a. Simular 1000 valores de cada densidad inicial P1 y P2. Resumiendo las muestras simuladas, mostrar que ambas iniciales concuerdan con las creencias iniciales acerca de la probabilidad p del lanzamiento de moneda.

```r
#semilla
set.seed(12345)
#Parámetros
alpha_1_ini = 100
beta_1_ini = 100
###
alpha_21_ini = 500
beta_21_ini = 500
alpha_22_ini = 1
beta_22_ini = 1

#Simulación de P1 y P2 inicial
P1_inicial <-tibble (p = rbeta(1000, alpha_1_ini, beta_1_ini))
P2_inicial <-tibble (p =0.9*rbeta(1000, alpha_21_ini, beta_21_ini)+0.1*rbeta(1000, alpha_2
# Comprobamos que al menos el 90% esté en el intervalo [0.44, 0.56]
print(sum(P1_inicial>=0.44 & P1_inicial<=0.56) / 1000)
```

```
[1] 0.918
```

```
print(sum(P2_inicial>=0.44 & P2_inicial<=0.56) / 1000)
```

```
[1] 0.966
```

```
quantile(P1_inicial$p , c(0.05,0.95))
```

```
      5%       95% 
0.4409330 0.5523635 
```

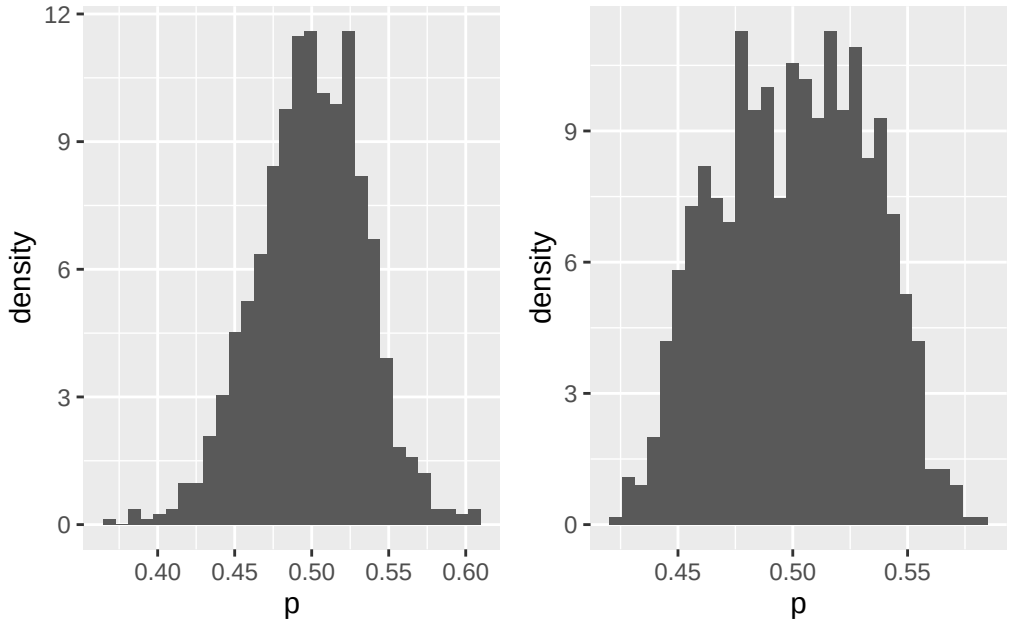
```
quantile(P2_inicial$p , c(0.05,0.95))
```

```
      5%       95% 
0.4490553 0.5511169 
```

Por tanto cumplen con tener al menos el 90% de probabilidad de $p \in [0.44, 0.56]$

```
g1 <- ggplot(P1_inicial) +
  geom_histogram(aes(x = p, y = ..density..), bins = 30)
g2 <- ggplot(P2_inicial) +
  geom_histogram(aes(x = p, y = ..density..), bins = 30)
g1+g2
```

```
Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
i Please use `after_stat(density)` instead.
```



b. Supongan que lanzan la moneda 100 veces y obtienen 45 soles. Simular 1000 valores de las distribuciones posteriores P1 y P2, y calcular intervalos de probabilidad del 90.

Para P1, se usa el conjugado Beta-Binomial, dado que la distribución inicial es una Beta y la verosimilitud está dada por una Binomial:

$$\text{Likelihood: } Binomial(n,p) \Rightarrow P(x|n,p) = \binom{n}{x} p^x (1-p)^{n-x}, p \in [0,1]$$

Por lo que la posterior será de la forma:

$$\text{Posterior} \propto \text{Likelihood} \times \text{Prior}$$

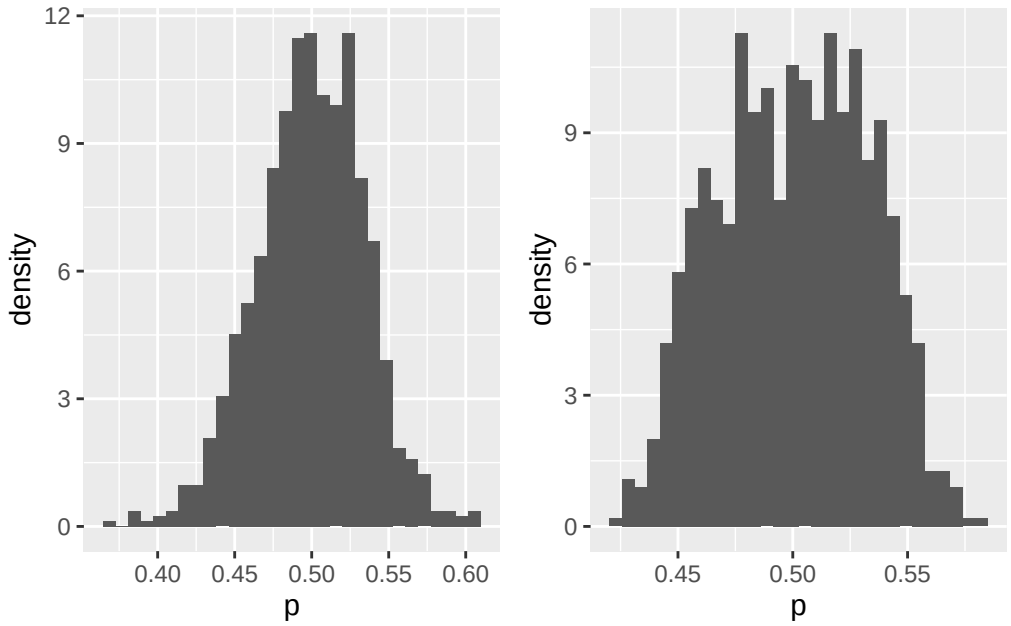
$$\text{Posterior: } Beta(\alpha + x, \beta + n - x)$$

```
n = 100 # en 100 lanzamientos
x_B = 45  # se obtienen 45 éxitos
# Nuevos parámetros
alpha_1_posB = alpha_1_ini + x_B
beta_1_posB = beta_1_ini + n - x_B

alpha_21_posB = alpha_21_ini + x_B
beta_21_posB = beta_21_ini + n - x_B
alpha_22_posB = alpha_22_ini + x_B
```

```
beta_22_posB = beta_22_ini + n - x_B
```

Haciendo la simulación:

```
P1_posB <-tibble (p = rbeta(1000, alpha_1_posB, beta_1_posB))
P2_posB <-tibble (p =0.9*rbeta(1000, alpha_21_posB, beta_21_posB)+0.1*rbeta(1000, alpha_22

g3 <- ggplot(P1_posB) +
  geom_histogram(aes(x = p, y = ..density..), bins = 30)
g4 <- ggplot(P2_posB) +
  geom_histogram(aes(x = p, y = ..density..), bins = 30)
g1+g2
```



Calculando intervalos de probabilidad:

```
quantile(P1_posB$p , c(0.05,0.95))
```

```
       5%        95%
0.4389260 0.5291968
```

```r
quantile(P2_posB$p , c(0.05,0.95))
```

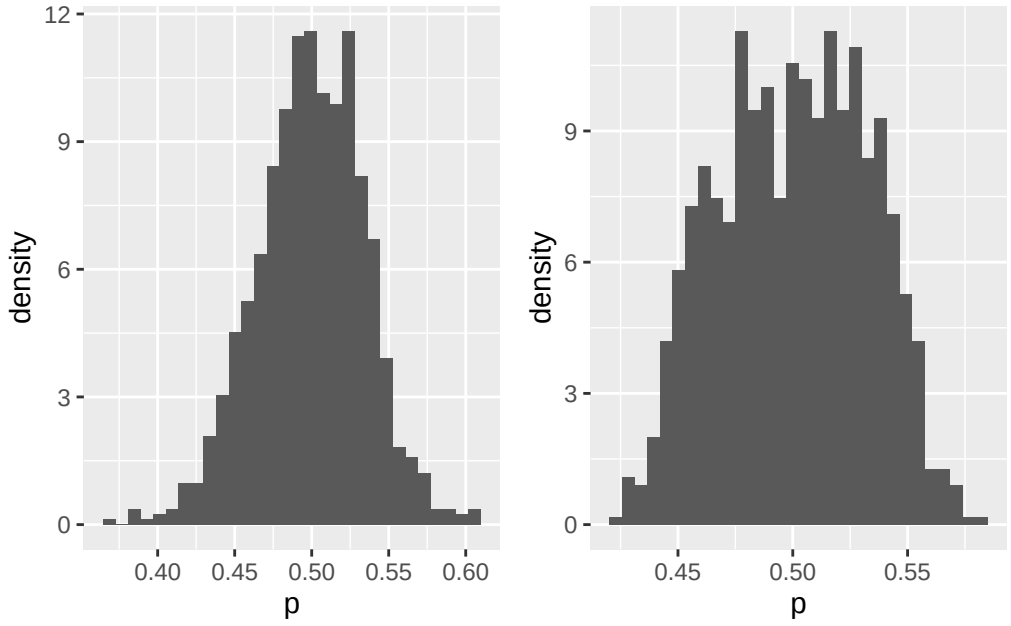
```
      5%       95%
0.4658571 0.5143838
```

c. Supongan que sólo observan 30 soles de los 100 lanzamientos. Nuevamente simular 1000 valores de las dos posteriores y calcular intervalos de probabilidad del 90.

```r
n = 100 # en 100 lanzamientos
x_C = 30  # se obtienen 45 éxitos
# Nuevos parámetros
alpha_1_posC = alpha_1_ini + x_C
beta_1_posC = beta_1_ini + n - x_C

alpha_21_posC = alpha_21_ini + x_C
beta_21_posC = beta_21_ini + n - x_C
alpha_22_posC = alpha_22_ini + x_C
beta_22_posC = beta_22_ini + n - x_C
```

Haciendo la simulación:

```r
P1_posC <-tibble (p = rbeta(1000, alpha_1_posC, beta_1_posC))
P2_posC <-tibble (p =0.9*rbeta(1000, alpha_21_posC, beta_21_posC)+0.1*rbeta(1000, alpha_22

g3 <- ggplot(P1_posC) +
  geom_histogram(aes(x = p, y = ..density..), bins = 30)
g4 <- ggplot(P2_posC) +
  geom_histogram(aes(x = p, y = ..density..), bins = 30)
g1+g2
```



Calculando intervalos de probabilidad:

```
quantile(P1_posC$p , c(0.05,0.95))
```

```
      5%       95%
0.3844803 0.4826971
```

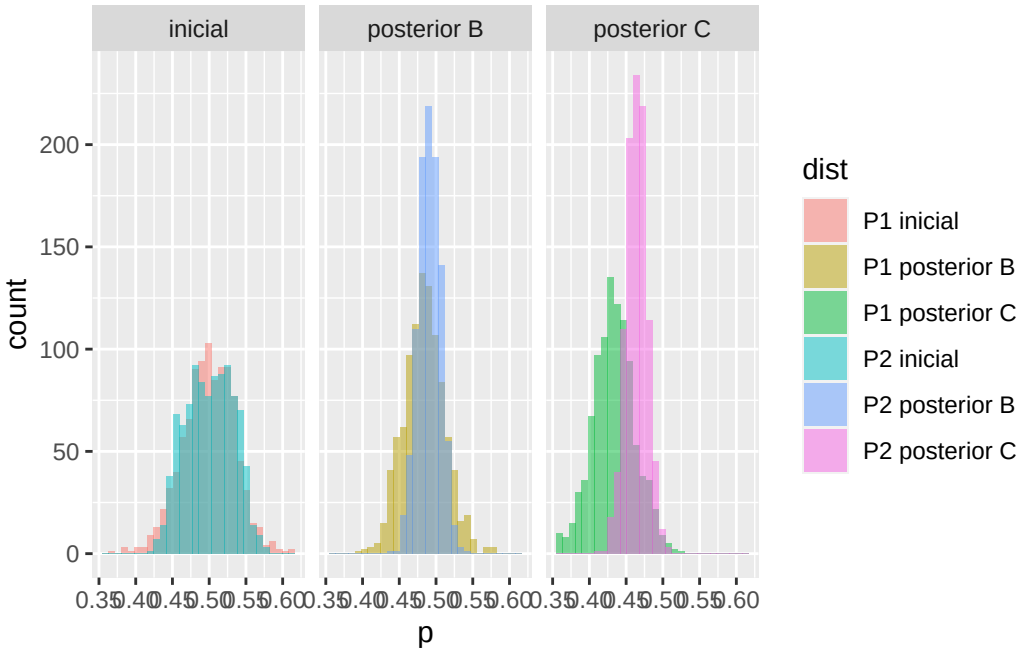
```
quantile(P2_posC$p , c(0.05,0.95))
```

```
      5%       95%
0.4398884 0.4869881
```

d. Viendo los resultados de (b) y (c), comentar sobre la robustez de la inferencia con respecto a la elección de la densidad inicial en cada caso.

```
P1_inicial <- P1_inicial %>% mutate(dist = "P1 inicial", t = "inicial")
P1_posB <- P1_posB %>% mutate(dist = "P1 posterior B", t = "posterior B")
P1_posC <- P1_posC %>% mutate(dist = "P1 posterior C", t = "posterior C")

P2_inicial <- P2_inicial %>% mutate(dist = "P2 inicial", t = "inicial")
```

```r
P2_posB <- P2_posB %>% mutate(dist = "P2 posterior B", t = "posterior B")
P2_posC <- P2_posC %>% mutate(dist = "P2 posterior C", t = "posterior C")

sims <- bind_rows(P1_inicial, P1_posB, P1_posC, P2_inicial, P2_posB, P2_posC)

g5 <- ggplot(sims, aes(x = p, fill = dist)) +
  geom_histogram(aes(x = p), bins = 30, alpha = 0.5, position = "identity")
g5 + facet_wrap(~t)
```



La inicial P2, tiene distribuciones posteriores B y C más angostas que para la inicial P1, e incluso con áreas de alta desidad de probabilidad distintas en el caso de una posterior C

6. **Aprendiendo de datos agrupados**. Supongan que manejan en carretera y típicamente manejan a una velocidad constante de $70 km/h$. Un día, rebasan un carro y son rebasados por 17 carros. Supongan que las velocidades son distribuídas $N(, 100)$. Si rebasan s carros y son rebasados por f,

Considerando que los carros o los rebasas o te rebasan, suponemos que sigue una distribución Binomial.

Por lo que la función de verosimilitud esta dada por:

$$\text{Likelihood: } Binomial(n,p) \Rightarrow P(p|n,x) = \binom{n}{x} p^x (1-p)^{n-x}, p \in [0,1]$$

donde $x = s$ el número de carros que te rebasan, $(n-x) = f$ es igual al número de carros que rebasas, y la proporción $p (y, ,) es proporcional la distribución acumulada de la distribución de velocidades $\mathcal{N}(\mu, \sigma^2 = 100)$. Sustituyendo obtenemos

$$\mathcal{L}(\mu) \propto \Phi(70, \mu, 100)^s (1 - \Phi(70, \mu, 100))^f$$

Y la log-verosmilitud será:

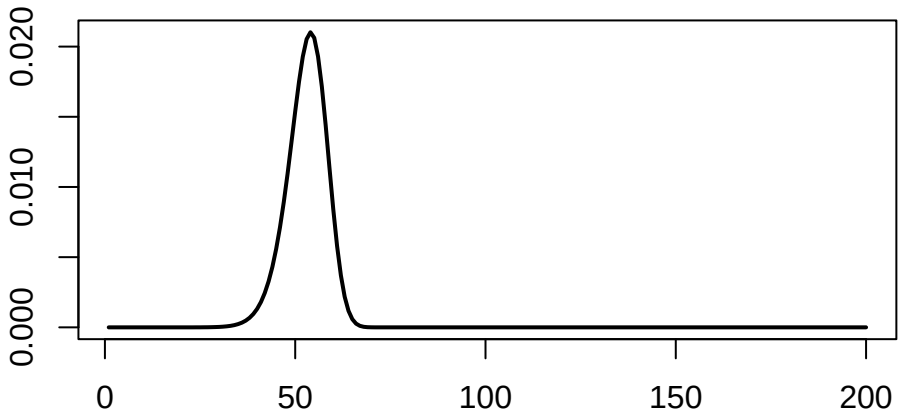
$$\log(\mathcal{L}) = \log \binom{n}{x} + x \log(p) + (n-x) \log(1-p)$$

$$\log(\mathcal{L}(\mu)) \propto s \log(\Phi(70, \mu, 100)) + f \log(1 - \Phi(70, \mu, 100))$$

a. ¿Cuál es la verosimilitud de ?

Calculando:

```r
#Parámetros
s <- 17
f <- 1
y <- 70
sigma <- 10
mu <- seq(1,200,1)

# Creamos la CDF o Phi
Phi <- pnorm(y, mu, sigma, log = FALSE)

#Hacemos la verosimilitud
likelihood <- Phi^s*(1-Phi)^f
#likelihood <- s*log(Phi) + f*log(1-Phi)

plot(mu,likelihood, type = 'l', lwd = 2, xlab = "", ylab = "")
```



Encontrando el máximo:

```
max(likelihood)
```

```
[1] 0.02102246
```

```
Phi_mu <- pnorm(y, 54, sigma, log = FALSE)
#s*log(Phi_mu) + f*log(1-Phi_mu)
Phi_mu^s*(1-Phi_mu)^f
```

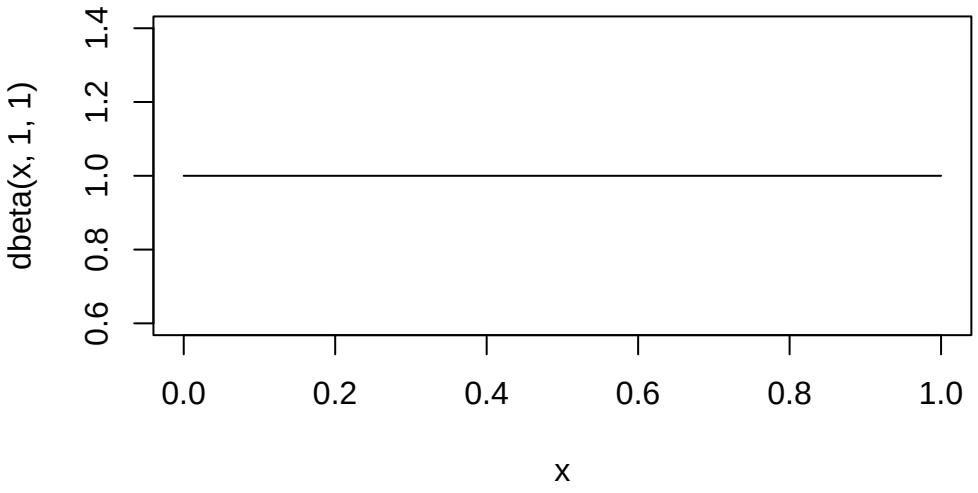
```
[1] 0.02102246
```

La velocidad promedio es 54km/h

b. Asignando una densidad inicial plana para , si $s = 1$ y $f = 17$, graficar la densidad posterior de .

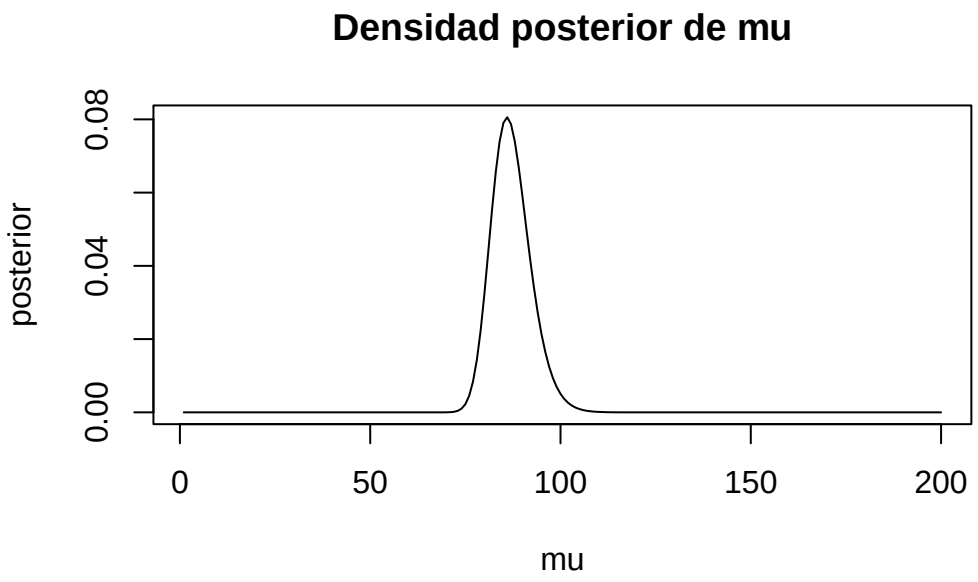
Sabiendo que podemos simular una distribución uniforme con una $beta(1, 1)$

```r
#define range
x = seq(0, 1, length=200)

#create plot of Beta distribution with shape parameters 2 and 10
plot(x, dbeta(x, 1, 1), type='l')
```



Multiplicamos la inicial por la verosimilitud para obtener la posterior

```r
#Parámetros
s <- 1
f <- 17

likelihood <- Phi^s*(1-Phi)^f
prior <- dbeta(x,1 ,1)
posterior <- prior*likelihood/sum(prior*likelihood)
plot(mu, posterior, type = 'l', ylab = "posterior", main = "Densidad posterior de mu")
```



c. Usando la densidad encontrada en (b), encontrar la media posterior de .

Hacemos una suma ponderada para obtener el promedio

```
sum(mu * posterior)
```

```
[1] 87.11109
```

d. Encontrar la probabilidad de que la velocidad promedio de los carros exceda 80 km/h.

Para esto sumamos los valores de la distrbución de 80 a 200 km/h

```
sum(posterior[80:200])
```

```
[1] 0.9468573
```

7. *Problema de Behrens-Fisher.* Supongan que se observan dos muestras normales independientes, la primera se distribuye de acuerdo a una $N(_1, \sigma_1^2)$ y la segunda de acuerdo a $N(_2, \sigma_2^2)$. Denoten la primera muestra por $x_1, ..., x_m$ y la segunda muestra por $y_1 ..., y_n$

Supongan también que los parámetros $\theta = (_1, \sigma_1^2, _2, \sigma_2^2)$ tienen la distribución inicial vaga dada por:

$$g(\theta) \propto \frac{1}{\sigma_1^2 \sigma_2^2}$$

a. Encontrar la densidad posterior. Mostrar que los vectores $(_1, \sigma_1^2)$ y $(_2, \sigma_2^2)$ tienen distribuciones posteriores independientes.

La verosimilitud (f(X| _1, _1^2)) y (f(X| _2, _2^2)) de las funciones normales está dada por:

$$f(X|\mu_1, \sigma_1^2) = \frac{1}{(2\pi\sigma_1^2)^{m/2}} \exp\left(-\frac{1}{2\sigma_1^2}\sum_{i=1}^{m}(x_i - \mu_1)^2\right)$$

y

$$f(X|\mu_2, \sigma_2^2) = \frac{1}{(2\pi\sigma_2^2)^{n/2}} \exp\left(-\frac{1}{2\sigma_2^2}\sum_{i=1}^{n}(x_i - \mu_2)^2\right)$$

Entonces la densidad posterior está dada por el producto de las verosimilitudes por la distribución inicial:

La expresión del posterior proporcional a la función de likelihood y la función de prior es:

[

$$\begin{aligned}\text{Posterior} &\propto f(X, Y|\mu_1, \sigma_1^2, \mu_2, \sigma_2^2) \cdot g(\mu_1, \sigma_1^2, \mu_2, \sigma_2^2) \\ &\propto \frac{1}{(\sqrt{2\pi\sigma_1^2})^{m/2}} \exp\left(-\frac{1}{2\sigma_1^2}\sum_{i=1}^{m}(x_i - \mu_1)^2\right) \cdot \frac{1}{(\sqrt{2\pi\sigma_2^2})^{n/2}} \exp\left(-\frac{1}{2\sigma_2^2}\sum_{i=1}^{n}(y_i - \mu_2)^2\right) \cdot \frac{1}{\sigma_1^2\sigma_2^2} \\ &\propto (\sigma_1^2)^{-m/2} \exp\left(-\frac{1}{2\sigma_1^2}\sum_{i=1}^{m}(x_i - \mu_1)^2\right) \cdot (\sigma_2^2)^{-n/2} \exp\left(-\frac{1}{2\sigma_2^2}\sum_{i=1}^{n}(y_i - \mu_2)^2\right) \cdot (\sigma_1^2\sigma_2^2)^{-1}\end{aligned}$$

]

Para demostrar que los vectores ((_1, _1^2)) y ((_2, _2^2)) tienen distribuciones posteriores independientes, debemos mostrar que la distribución posterior conjunta se puede factorizar como el producto de sus distribuciones marginales ((_1, _1^2)) y ((_2, _2^2)).

La marginal para ((_1, _1^2)) sigue una distribución normal inversa-gamma dada por:

[Posterior(_1, _1^2 | X, Y) = Normal-Inverse-Gamma(_1, _1^2)]

y análogamente para ((_2, _2^2)), su distribuciòn es:

[Posterior(_2, _2^2 | X, Y) = Normal-Inverse-Gamma(_2, _2^2)]

A continuación demostramos con simulación la siguiente expresión:

[Posterior(_1, _1^2, _2, _2^2 | X, Y) = Posterior(_1, _1^2 | X, Y) Posterior(_2, _2^2 | X, Y)]

```
library(rstan)
```

```
Loading required package: StanHeaders
```

```
rstan version 2.32.5 (Stan version 2.32.2)
```

```
For execution on a local, multicore CPU with excess RAM we recommend calling
options(mc.cores = parallel::detectCores()).
To avoid recompilation of unchanged Stan programs, we recommend calling
rstan_options(auto_write = TRUE)
For within-chain threading using `reduce_sum()` or `map_rect()` Stan functions,
change `threads_per_chain` option:
rstan_options(threads_per_chain = 1)
```

```
Attaching package: 'rstan'
```

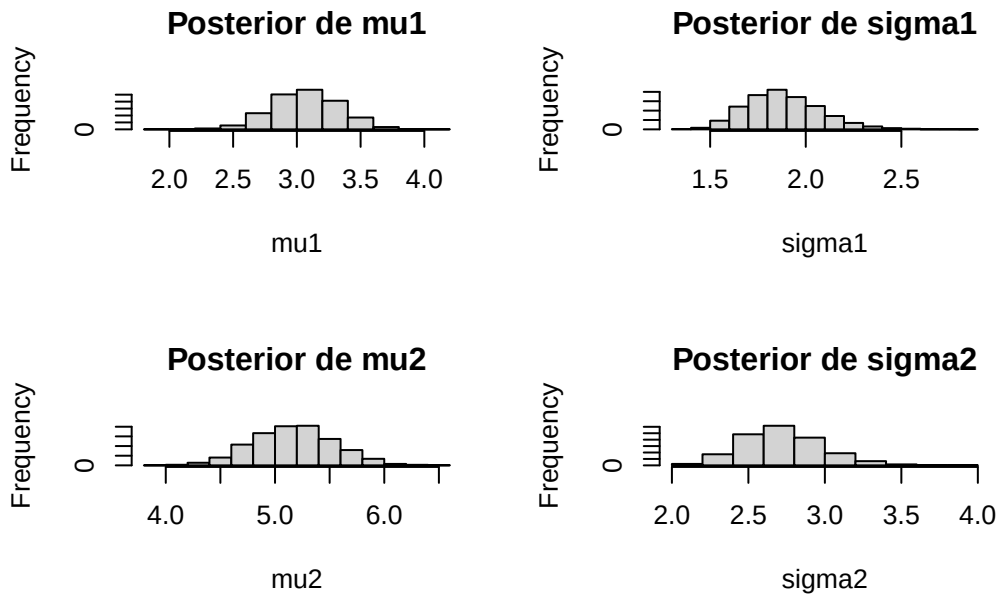
```
The following object is masked from 'package:tidyr':

    extract
```

```
options(mc.cores = parallel::detectCores())
rstan_options(auto_write = TRUE)
```

```
# Definir el modelo en Stan
stan_model <- "
data {
  int<lower=0> m;  // Tamaño de la muestra 1
  int<lower=0> n;  // Tamaño de la muestra 2
  vector[m] x;    // Datos de la muestra 1
  vector[n] y;    // Datos de la muestra 2
}

parameters {
  real mu1;        // Media de la muestra 1
```

```
  real<lower=0> sigma1_sq;  // Varianza de la muestra 1
  real mu2;         // Media de la muestra 2
  real<lower=0> sigma2_sq;  // Varianza de la muestra 2
}

model {
  // Likelihood
  x ~ normal(mu1, sqrt(sigma1_sq));
  y ~ normal(mu2, sqrt(sigma2_sq));

  // Prior
  target += -log(sigma1_sq) - log(sigma2_sq);
}

generated quantities {
  // Posterior samples
  real posterior_mu1[m];
  real<lower=0> posterior_sigma1_sq[m];
  real posterior_mu2[n];
  real<lower=0> posterior_sigma2_sq[n];

  // Simular muestras de la posterior
  for (i in 1:m) {
    posterior_mu1[i] = normal_rng(mu1, sqrt(sigma1_sq));
    posterior_sigma1_sq[i] = inv_gamma_rng(0.5, 0.5);
  }

  for (j in 1:n) {
    posterior_mu2[j] = normal_rng(mu2, sqrt(sigma2_sq));
    posterior_sigma2_sq[j] = inv_gamma_rng(0.5, 0.5);
  }
}
"

# Convertir el modelo Stan a un objeto stan
stan_model <- stan_model(model_code = stan_model)

# Datos de ejemplo
set.seed(123)
m <- 50
n <- 60
```

```r
x <- rnorm(m, mean = 3, sd = 2)
y <- rnorm(n, mean = 5, sd = 3)

# Parámetros de la cadena de Markov Monte Carlo (MCMC)
chains <- 4
iterations <- 2000

# Ejecutar MCMC
stan_data <- list(m = m, n = n, x = x, y = y)
fit <- sampling(stan_model, data = stan_data, chains = chains, iter = iterations)

# Obtener muestras posteriores simuladas
posterior_samples <- extract(fit)

# Visualizar resultados
par(mfrow = c(2, 2))

hist(posterior_samples$mu1, main = "Posterior de mu1", xlab = "mu1")
hist(sqrt(posterior_samples$sigma1_sq), main = "Posterior de sigma1", xlab = "sigma1")

hist(posterior_samples$mu2, main = "Posterior de mu2", xlab = "mu2")
hist(sqrt(posterior_samples$sigma2_sq), main = "Posterior de sigma2", xlab = "sigma2")
```



Con esto comprobamos que la distribución posterior se puede factorizar como el producto de sus distribuciones marginales y los vectores tienen distribuciones posteriores independientes.

b. Describir cómo simular la densidad posterior conjunta de θ.

Como las distribuciones son independientes, podemos simular valores de ambas distribuciones y combinarlas

c. Los siguientes datos dan la longitud de la mandíbula en mm para 10 chacales machos y 10 chacales hembras en la colección del Museo Británico. Usando simulación, encontrar la densidad posterior de la diferencia en la longitud media de las mandíbulas entre los sexos. ¿Hay suficiente evidencia para concluir que los machos tienen una longitud promedio mayor que las hembras?

Calculando:

```
Machos = c(120, 107, 110, 116, 114, 111, 113, 117, 114, 112)
Hembras = c(110, 111, 107, 108, 110, 105, 107, 106, 111, 111)

S_machos <- sum((Machos - mean(Machos))^2)
S_Hembras <- sum((Hembras - mean(Hembras))^2)

N_machos <- length(Machos)
N_hembras <- length(Hembras)

s2_machos <- S_machos/rchisq(1000,N_machos-1)
s2_hembras <- S_Hembras/rchisq(1000,N_hembras-1)

mu_machos <- rnorm(1000, mean=mean(Machos), sd=sqrt(s2_machos)/sqrt(N_machos))
mu_hembras <- rnorm(1000, mean=mean(Hembras), sd=sqrt(s2_hembras)/sqrt(N_hembras))

mu_diffs <- mu_machos - mu_hembras

intervalo <- c(0.05,0.95)
quantile(mu_diffs,intervalo)
```

```
      5%       95%
2.359780 7.369917
```

Como este intervalo es mayor a cero, concluimos que los machos tienen una longitud de mandíbula mayor que las hembras.

8. *Estimando los parámetros de una densidad Poisson/Gamma.* Supongamos que $y_1, ..., y_n$ es una muestra aleatoria de una densidad Poisson/Gamma:

$$f(y|a,b) = \frac{\Gamma(y+a)}{\Gamma(a)y!}\frac{b^a}{(b+1)^{y+a}}$$

donde $a \geq 0$, $b \geq 0$. Esta densidad es un modelo apropiado para conteos que muestran más dispersión que la que predice un modelo Poisson. Supongamos que (a, b) tiene asignada la inicial no informativa proporcional a $1/(ab)^2$. Si transformamos a los parámetros $\theta_1 = \log(a)$ y $\theta_2 = \log(b)$, la densidad posterior es proporcional a

$$g(\theta_1, \theta_2) \propto \prod_{i=1}^{n} \frac{\Gamma(yi+a}{\Gamma(a)y_i!}\frac{b^a}{(b+1)^{yi+a}}$$

donde $a = \exp(\theta_1)$ y $b = \exp(\theta_2)$. Usa este marco para modelar los datos obtenidos por Gilchrist (1984), en los que una serie de 33 trampas de insectos fueron puestas sobre varias dunas de arena y se registra el número de diferentes insectos atrapados sobre un tiempo fijo. El número de insectos en las trampas se muestran a continuación:

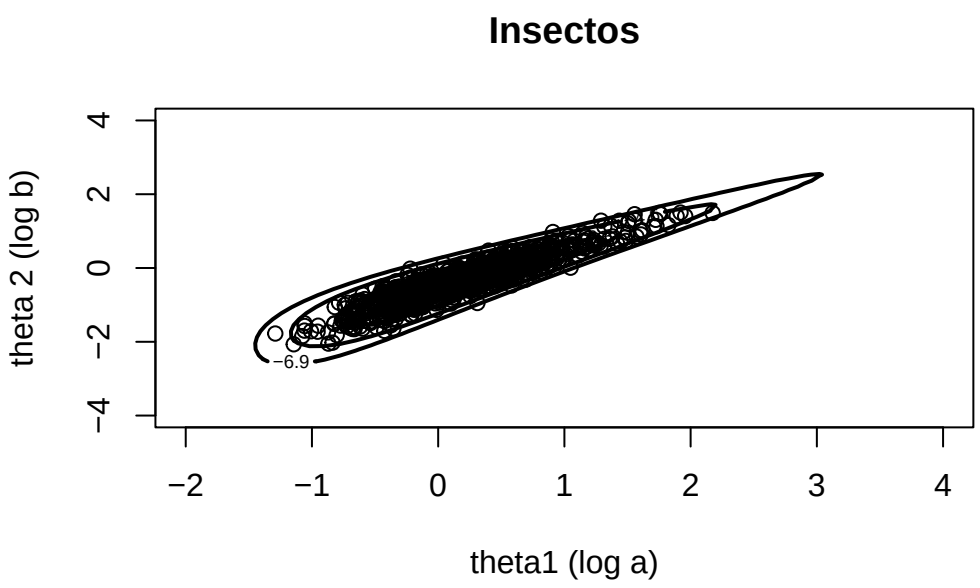
```
library(LearnBayes)

insectos = c(2,5,0,2,3,1,3,4,3,0,3,
             2,1,1,0,6,0,0,3,0,1,1,
             5,0,1,2,0,0,2,1,1,1,0)
```

Calculando la densidad posterior sobre una retícula, simular 1000 extracciones de la densidad conjunta posterior de (θ_1, θ_2). De la muestra simulada, encontrar intervalos estimados de 90 para los parámetros a y b.

Calculando:

```
yshhh <- c(2,5,0,2,3,1,3,4,3,0,3,
        2,1,1,0,6,0,0,3,0,1,1,
        5,0,1,2,0,0,2,1,1,1,0)

## De LearnBayes se usará mycontour y simcontour para responder el ejercicio


## Funciones para posterior

gamma_func <- function(y,a,b) {
  return( gamma(y+a) *
          (b ^ a) *
```

```
          (1/( gamma(a) * factorial(y) )) *
          (1/( (b+1) ^ (y+a) ))
  )
}

poisson_gamma_post <- function(theta,insectos) {
  a <- exp(theta[1])
  b <- exp(theta[2])

  return(log( (1/(a*b)) * prod(gamma_func(insectos,a,b)) ))
}


## Grid para gráfica contour

grid <- c(-2,4,-4,4)

mycontour(poisson_gamma_post,grid,insectos,xlab="theta1 (log a)", ylab="theta 2 (log b)",
sims <- simcontour(poisson_gamma_post,grid,insectos,1000)
points(sims$x,sims$y)
```



```r
as <- exp(sims$x)
bs <- exp(sims$y)
qs <- c(.05,.95)
quantile(as,qs)
```

```
       5%       95%
0.5383676 3.0357105
```

```r
quantile(bs,qs)
```

```
       5%       95%
0.2662942 1.8728760
```