

# Final-2023

Mariano Villafuerte - 156057

Mario Medina - 156940

## 1. Pruebas de hipótesis

**1.1** De acuerdo a una encuesta en EUA, 26% de los residentes adultos de Illinois han terminado la preparatoria. Un investigador sospecha que este porcentaje es menor en un condado particular del estado. Obtiene una muestra aleatoria de dicho condado y encuentra que 69 de 310 personas en la muestra han completado la preparatoria. Estos resultados soportan su hipótesis?

### Respuesta:

Podemos tomar 2 enfoques, a continuación explicamos el porqué

- **Prueba con estadístico Z:** dado que hablamos de proporciones sabemos cuál es el error estándar de una proporción, podremos calcular el estadístico Z, dada la naturaleza de la prueba también puede definirse como una prueba de Wald
- **Enfoque bayesiano:** el 26% nos ayuda a definir una a priori y con los datos podemos generar una posterior. No es un cálculo de prueba de hipótesis tal cual pero podemos obtener intervalos de credibilidad que nos ayuden a determinar si realmente es significativamente menor.

Empezamos con la **prueba del estadístico Z**, nuestra prueba de hipótesis la podemos definir como (1 cola)

$$H_0 : \hat{\theta} = 0.26$$

$$H_1 : \hat{\theta} < 0.26$$

Y el estadístico se vería de la siguiente forma. Sabemos que  $\hat{\theta} = \frac{69}{310}$

$$Z = \frac{\hat{\theta} - 0.26}{\sqrt{\frac{0.26(1-0.26)}{310}}} = -1.502016$$

El valor-p considerando que es de una cola sería, en específico la izquierda

$$p - value = P(Z < z)$$

el cálculo se ve de la siguiente manera.

```
numerador = (69/310)-0.26
denominador = sqrt((0.26*0.74)/310)
p_value <- pnorm(numerador/denominador)
```

```
[1] "El valor p asociado a esta prueba es: 0.07"
```

**La conclusión es que no es significativo al 95% de confianza.** Debido a que es mayor al valor crítico de 5%, por lo que no hay suficiente evidencia para rechazar la hipótesis nula.

**Enfoque bayesiano:** El problema trata de la estimación de una proporción, llamémosle  $\theta$  donde  $\theta$  es la proporción de adultos que terminaron la preparatoria en el condado específico de Illinois. Podemos asumir una a priori  $P(\theta)$  que siga la información inicial que nos dice que ese porcentaje dentro de Illinois es de aproximadamente 26%, entonces usaremos una **Beta** que después de prueba y error tiene los parámetros (4,11) que tiene de media 0.26 sin estar muy concentrada.

```
set.seed(156057)
sim_inicial <- tibble(theta = stats::rbeta(10000,4,11))
```

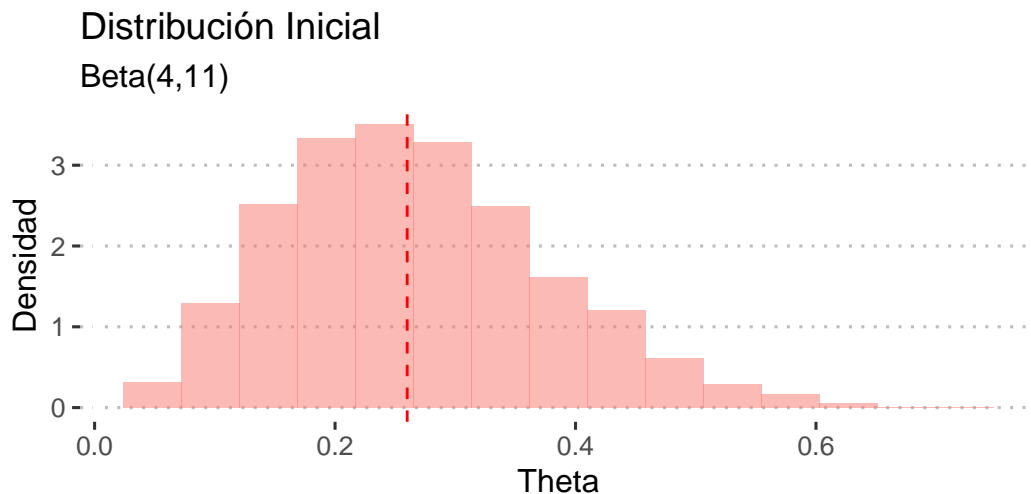


Figure 1: Distribución a priori: Beta (4,11)

Los datos del condado nos dicen que de 310 individuos únicamente tenemos 69 con preparatoria concluida (éxitos) entonces nuestra posterior queda de la siguiente manera

$$P(\theta|X) \propto P(X|\theta)P(\theta)P(\theta|X) \propto \theta^{69+3}(1-\theta)^{241+10}P(\theta|X) \propto \theta^{72}(1-\theta)^{251}$$

Obtenemos los siguientes histogramas.

```
set.seed(156057)
sim_inicial <- sim_inicial %>% mutate(dist = "inicial")
sim_posterior <- tibble(theta = rbeta(10000, 73, 252)) %>%
  mutate(dist = "posterior")

sims <- bind_rows(sim_inicial, sim_posterior)
```

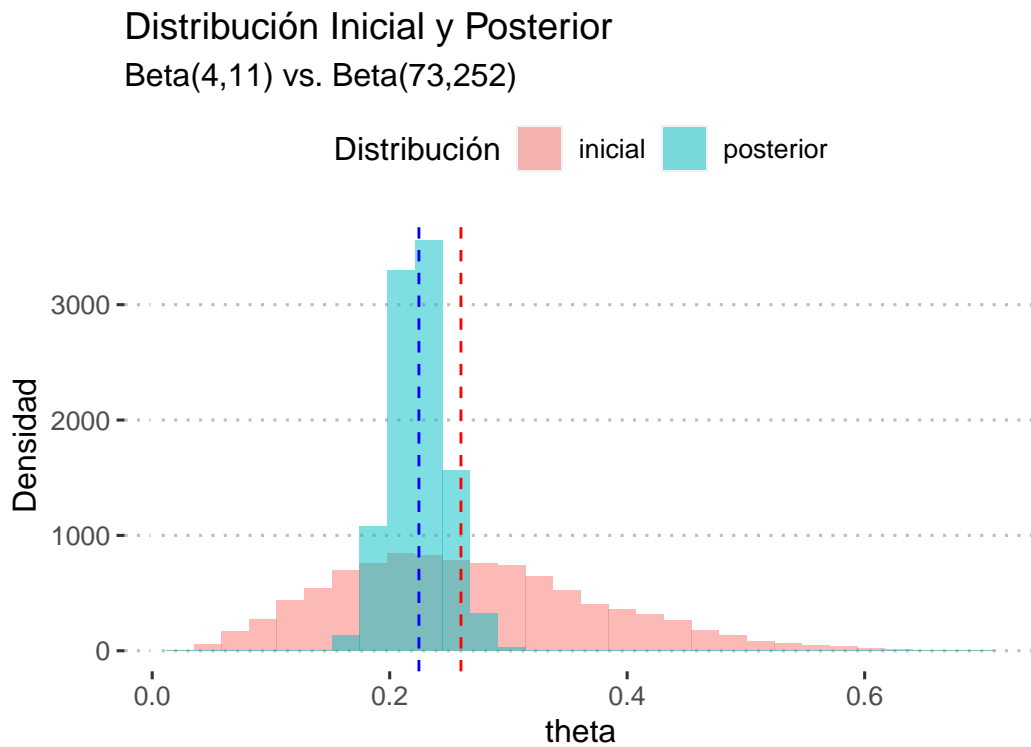


Figure 2: Inicial vs. Posterior

Un **estimador puntual** es la media de la distribuciones, que al ser una Beta se obtiene como  $\frac{a}{a+b}$

- Dist. Inicial:  $\frac{4}{4+11} = 0.267$
- Dist. Posterior:  $\frac{73}{73+252} = 0.2246$
- Máxima verosimilitud:  $\frac{69}{310} = 0.2226$

Asímismo podemos obtener intervalos de credibilidad fácilmente debido a nuestra distribución “conocida”

```
set.seed(156057)
inferior <- round(qbeta(0.025, shape1 = 73, shape2 = 252), 2)
superior <- round(qbeta(0.975, shape1 = 73, shape2 = 252), 2)
```

Table 1: Intervalo dist. posterior

Inferior	Superior
0.18	0.27

Dado la información anterior el **intervalo de confianza al 95% de la posterior sí incluye el valor del 26%. Por lo que todavía no es del todo “aceptable”** que la proporción del condado sea significativamente menor al del estado de Illinois La conclusión de este enfoque es el mismo que en el primero.

**1.2** Mendel criaba chícharos de semillas lisas amarillas y de semillas corrugadas verdes. Éstas daban lugar a 4 tipos de descendientes: amarillas lisas, amarillas corrugadas, verdes lisas y verdes corrugadas. El número de cada una es multinomial con parámetro  $p = (p_1, p_2, p_3, p_4)$ . De acuerdo a su teoría de herencia este vector de probabilidades es:

$$p = (9/16, 3/16, 3/16, 1/16)$$

A lo largo de  $n = 556$  experimentos observó  $x = (315, 101, 108, 32)$ . Utiliza la prueba de cociente de verosimilitudes para probar  $H_0 : p = p_0$  contra  $H_0 : p \neq p_0$ .

### Respuesta

La distribución multinomial para este problema la definimos de la siguiente manera

$$p(x_1, x_2, x_3, x_4 | p_1, p_2, p_3, p_4) = \frac{n!}{x_1! x_2! x_3! x_4!} p_1^{x_1} p_2^{x_2} p_3^{x_3} p_4^{x_4}$$

La primera parte será constante al momento de sacar la log-verosimilitud, por lo que la parte de interes nos queda de la siguiente forma.

$$l(p_1, p_2, p_3, p_4) = \sum_{i=1}^4 x_i \log(p_i)$$

También sabemos que el estimador de máxima verosimilitud para cada  $p_i$  es  $\frac{x_i}{n}$

Para la prueba de hipótesis de cociente de verosimilitud necesitamos calcular la log-verosimilitud bajo la hipótesis nula  $p$  y la log verosimilitud utilizando los estimadores de máxima verosimilitud. Lo haremos bajo simulación

Definimos

$$\lambda = 2[l(\hat{p}) - l(p_0)]$$

```
set.seed(156057)
experimentos <- 556
observaciones <- c(315, 101, 108, 32)
prob_nulas <- c(9/16, 3/16, 3/16, 1/16)
simul_nula <- rmultinom(15000, experimentos, prob_nulas)

lambda <- function(n, x, p = prob_nulas){
  # Estimadores MV
  p1_mv <- x[1]/n
  p2_mv <- x[2]/n
```

```

p3_mv <- x[3]/n
p4_mv <- x[4]/n
# log verosimilitud bajo mv
log_p_mv <- x[1]*log(p1_mv)+x[2]*log(p2_mv)+x[3]*log(p3_mv)+x[4]*log(p4_mv)
# log verosimilitud bajo nula
log_p_nula <- x[1]*log(p[1])+x[2]*log(p[2])+x[3]*log(p[3])+x[4]*log(p[4])
lambda <- 2*(log_p_mv - log_p_nula)
lambda
}

lambda_obs <- lambda(experimentos, observaciones, prob_nulas)

new_df <- data.frame(simul_nula)
sims_tbl <- data.frame(sim_x = I(as.list(new_df)))
sims_tbl <- sims_tbl %>%
  mutate(lambda = map_dbl(sim_x,
                           ~lambda(experimentos,
                                     .x,
                                     prob_nulas)))

```

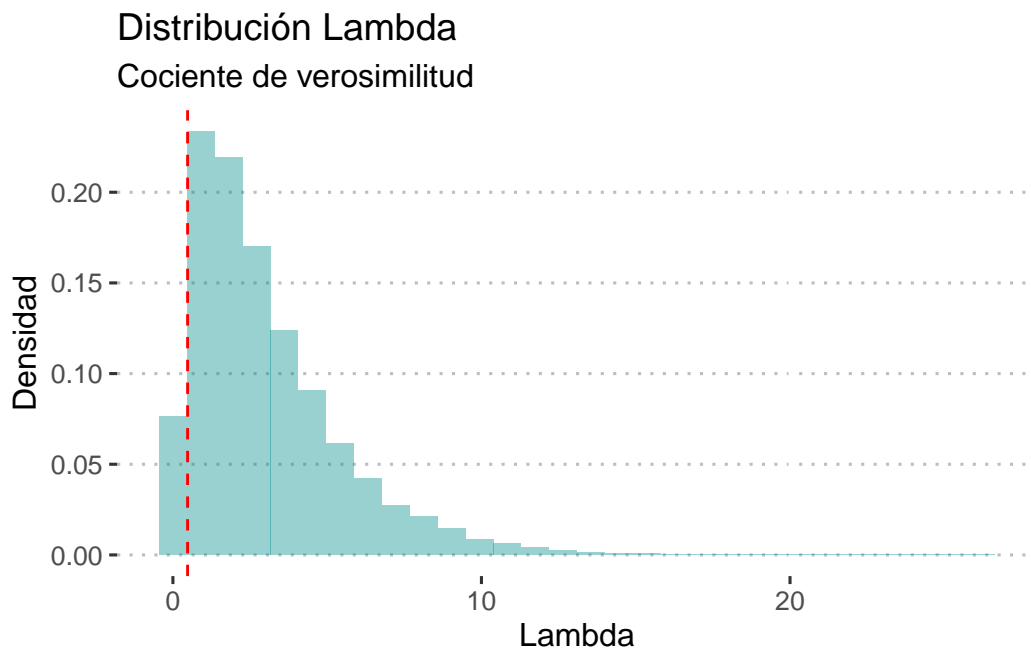


Figure 3: Distribución Lambda

Calculamos nuestro valor p.

```
valor_p <- mean(sims_tbl$lambda >= lambda_obs)
```

```
[1] "El valor p asociado a esta prueba es: 0.93"
```

Por lo que **no encontramos evidencia en contra de la hipótesis nula**. Hace sentido ya que desde la Lambda observada el valor es cercano a 0, por lo que la nula tiene bastante verosimilitud respecto a lo que los datos indican.

**1.3.** Sean  $X_1, \dots, X_n \sim \text{Poisson}(\lambda)$ ,

\* Sea  $\lambda_0 > 0$ . ¿Cuál es la prueba Wald para  $H_0 : \lambda = \lambda_0, H_1 : \lambda \neq \lambda_0$ ?

\* Si  $\lambda_0 = 1$ ,  $n = 20$  y  $\alpha = 0.05$ . Simula  $X_1, \dots, X_n \sim \text{Poisson}(\lambda_0)$  y realiza la prueba Wald, repite 1000 veces y registra el porcentaje de veces que rechazas  $H_0$ , qué tan cerca te queda el error del tipo 1 de 0.05?

### Respuesta

Para el primer inciso, hemos visto varias veces que el mejor estimador para  $\lambda$  de una Poisson será la media. Asimismo hemos visto resultados asintóticos donde la media tiene normalidad asintótica dicho esto podemos declarar que

$$W = \frac{\hat{\lambda} - \lambda}{\hat{e}\hat{e}} \sim N(0, 1)$$

Y el **valor-p** asociado para la hipótesis nula de  $H_0 : \lambda = \lambda_0, H_1 : \lambda \neq \lambda_0$  será

$$\text{valor} - p \approx P(|Z| > |w|) = 2(1 - \Phi(|w|))$$

Ahora si asumimos que  $\lambda_0 = 1$ ,  $n = 20$  y  $\alpha = 0.05$ . Podemos simular los datos de la siguiente manera. Un resultado importante es que como  $\lambda$  es estimado con la media, el  $\hat{e}\hat{e}$  puede ser calculado como el error estándar de la media con  $\frac{s}{\sqrt{n}}$

```
set.seed(156057)
n <- 20
lambda0 <- 1
datos_wald <- rpois(n, lambda0)
p_values <- c()
for (i in 1:1000) {
  datos_wald <- rpois(n, lambda0)
  w_test <- (mean(datos_wald) - lambda0) / (sd(datos_wald) / sqrt(n))
  p_values[i] <- 2 * (1 - pnorm(abs(w_test)))
}
resumen_wald <- data.frame(
  p_val = p_values
)
resume_wald <- resumen_wald %>%
  dplyr::mutate(rechazo = ifelse(p_val < 0.05, 1, 0),
               distancia = abs(0.05 - p_val),
               point = seq(1, 1000, 1))
```



**Rechazo el 7% de las veces.** El *Error del tipo I* es la probabilidad de rechazar hipótesis nula de  $H_0$  cuando es cierta, es decir que únicamente debemos calcular la distancia de aquellas simulaciones donde rechazamos

```
ggplot(resume_wald %>% filter(rechazo == 1),
  aes(x = point, xend = point,
    y = 0, yend = distancia,
    color = factor(rechazo))) +
geom_segment(size = 1) +
labs(title = "Distancia al punto crítico",
  x = "Estimación",
  y = "distancia a 0.05") +
ggpubr::theme_pubclean(base_size = 12)
```

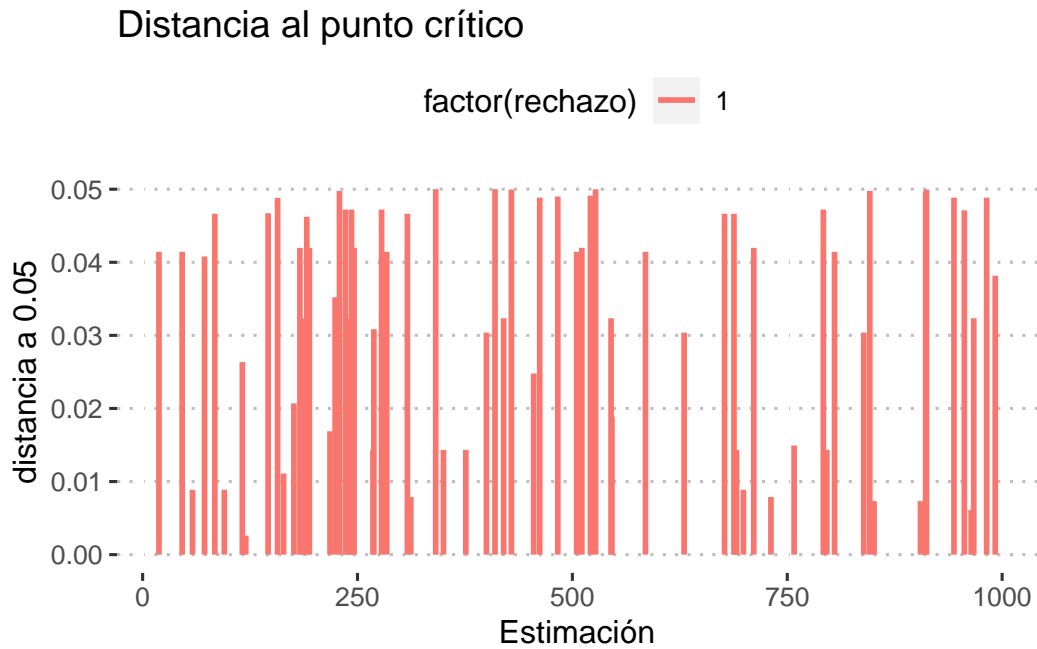


Figure 4: Distancia entre el error del tipo I y 5%

## 2. Relación entre bootstrap e inferencia bayesiana

Consideremos el caso en que tenemos una única observación  $x$  proveniente de una distribución normal

$$x \sim N(\theta, 1)$$

Supongamos ahora que elegimos una distribución inicial Normal.

$$\theta \sim N(0, \tau)$$

dando lugar a la distribución posterior (como vimos en la tarea)

$$\theta|x \sim N\left(\frac{x}{1 + 1/\tau}, \frac{1}{1 + 1/\tau}\right)$$

Ahora, entre mayor  $\tau$ , más se concentra la posterior en el estimador de máxima verosimilitud  $\hat{\theta} = x$ . En el límite, cuando  $\tau \rightarrow \infty$  obtenemos una inicial no-informativa (constante) y la distribución posterior

$$\theta|x \sim N(x, 1)$$

Esta posterior coincide con la distribución de bootstrap paramétrico en que generamos valores  $x^*$  de  $N(x, 1)$ , donde  $x$  es el estimador de máxima verosimilitud. Lo anterior se cumple debido a que utilizamos un ejemplo Normal pero también se cumple proximadamente en otros casos, lo que conlleva a una correspondencia entre el bootstrap paramétrico y la inferencia bayesiana. En este caso, la distribución bootstrap representa (aproximadamente) una distribución posterior no-informativa del parámetro de interés. Mediante la perturbación en los datos el bootstrap aproxima el efecto bayesiano de perturbar los parámetros con la ventaja de ser más simple de implementar (en muchos casos).

- Los detalles se pueden leer en \_\_The Elements of Statistical Learning\_\_ de Hastie y Tibshirani.

Comparemos los métodos en otro problema con el fin de apreciar la similitud en los procedimientos:

Supongamos  $x_1, \dots, x_n \sim N(0, \sigma^2)$ , es decir, los datos provienen de una distribución con media cero y varianza desconocida.

En los puntos 2.1 y 2.2 buscamos hacer inferencia del parámetro  $\sigma^2$ .

## 2.1 Bootstrap paramétrico.

\* Escribe la función de log-verosimilitud y calcula el estimador de máxima verosimilitud para  $\sigma^2$ . Supongamos que observamos los datos ‘x’ (en la carpeta datos), ¿Cuál es tu estimación de la varianza?

**R:** Sabemos que  $\mu = 0$ , entonces nos quedamos con la parte de la distribución que considera el término de  $\sigma^2$ . Tenemos 150 datos obtenidos del archivo *x.RData*

```
#load(paste0("C:/Users/mario/OneDrive/Documentos/fundamentos-final/data/x.RData"))  
  
load("data/x.RData")
```

$$\begin{aligned}f(x|\mu, \sigma^2) &= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \\f(x|0, \sigma^2) &= \frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \\L(0, \sigma^2|x) &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi}} (\sigma^2)^{-\frac{1}{2}} e^{-\frac{x_i^2}{2\sigma^2}} \propto (\sigma^2)^{-\frac{n}{2}} e^{-\frac{\sum x_i^2}{2\sigma^2}} \\l(0, \sigma^2|x) &= \frac{-n}{2} \log(\sigma^2) - \frac{\sum x_i^2}{2} (\sigma^2)^{-1}\end{aligned}$$

Tenemos  $n = 150$  y  $\sum^{150} x^2 = 19693.64$  podemos hacerlo de forma manual y calcular la derivada e igualar a 0.

$$\frac{\partial l}{\partial \sigma^2} = \frac{-150}{2} \frac{1}{\sigma^2} + \frac{19693.64}{2} \frac{1}{(\sigma^2)^2} = 0$$

Se puede resolver asumiendo  $x = \frac{1}{\sigma^2}$  y resolvemos por la “chicharronera”

```
set.seed(156057)  
# Coefficients  
a <- 9846.821  
b <- -75  
c <- 0  
  
# Quadratic formula  
discriminant <- b^2 - 4 * a * c  
x1 <- (-b + sqrt(discriminant)) / (2 * a)  
x2 <- (-b - sqrt(discriminant)) / (2 * a)  
  
# Reciprocal to get sigma^2
```

```
sigma_squared_1 <- 1 / x1
sigma_squared_2 <- 1 / x2
```

Lo que nos da un valor  $\sigma_{mv}^2 = 131.291$  . Podemos verlo graficamente y replicarlo con métodos de optimización numérica

```
set.seed(156057)
log_p <- function(pars){
  (-150/2)*log(pars[1]) - (19693.64/2)*((1)/pars[1])
}
solucion <- optim(c(0.5), log_p,
  control = list(fnscale = -1, maxit = 10000),
  method = "Nelder-Mead")

print(paste0("Comprobamos convergencia: ",solucion$convergence))
```

```
[1] "Comprobamos convergencia: 0"
```

```
est_mv <- tibble(parametro = c("varianza"), estimador = solucion$par) %>%
  column_to_rownames(var = "parametro")
est_mv
```

```
      estimador
varianza    131.65
```

Graficamos:

```
dat_verosim <- tibble(x = seq(5,300, 0.01)) %>% mutate(log_prob = map_dbl(x, log_p))
ggplot(dat_verosim, aes(x = x, y = log_prob)) + geom_line() +
  geom_vline(xintercept = 131.65, color = "red") +
  labs(title = "Optimización") +
  xlab("Varianza (sigma^2)") + ylab("Log verosimilitud") +
  theme_pubclean(base_size = 12)
```

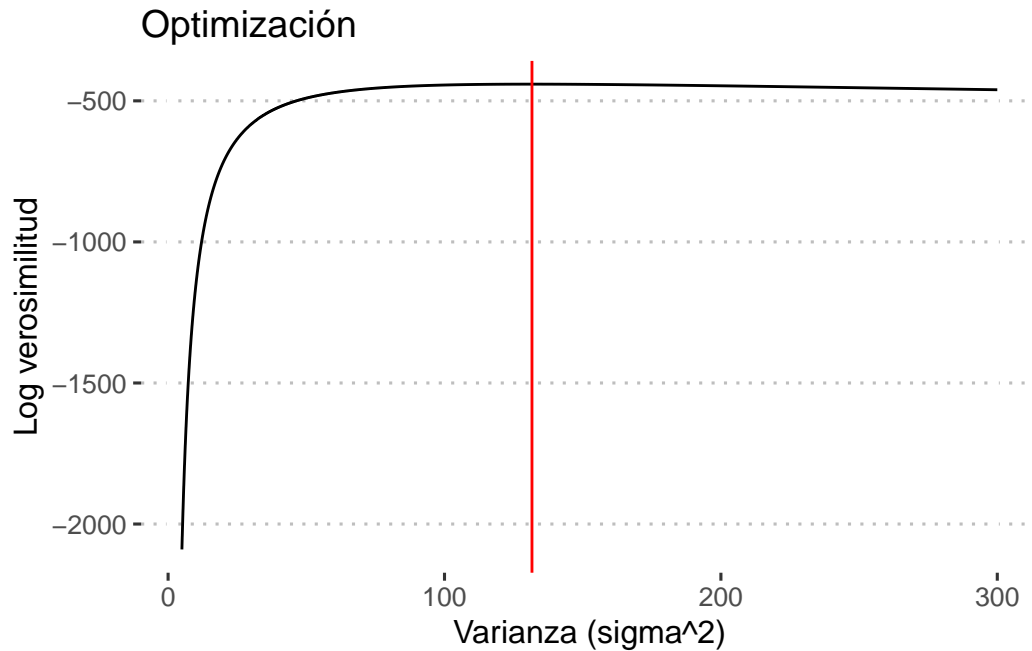


Figure 5: Optimización varianza

\* Aproxima el error estándar de la estimación usando `__bootstrap paramétrico__` y realiza un histograma de las replicaciones bootstrap.

**R:** Creamos nuestro flujo de generador de muestra (utilizando el parámetro de máxima verosimilitud), calculamos la log-verosimilitud y optimizamos

```
set.seed(156057)

est_mle <- 131.65
n <- 150

rep_boot <- function(rep, log_p, n, est_mle){
  muestra_bootstrap <- rnorm(n, 0, sqrt(est_mle))
  log_p <- function(pars){
    (-n/2)*log(pars[1]) - (sum(muestra_bootstrap^2)/2)*((1)/pars[1])
  }
  solucion <- optim(c(0.5), log_p,
                    control = list(fnscale = -1, maxit = 10000),
                    method = "Nelder-Mead")
  try(if(solucion$convergence != 0) stop("No se alcanzó convergencia."))
}
```

```

  tibble(parametro = c("varianza"), estimador_boot = solucion$par)
}

reps_boot <- map_dfr(1:15000, ~ rep_boot(.x, log_p, n = length(x),
                                         est_mle), rep = ".id")

ggplot(reps_boot, aes(x = estimador_boot)) +
  geom_histogram(aes(x = estimador_boot, y = ..density..), bins = 30,
                fill = "#F8766D",
                alpha=0.5) +
  geom_vline(xintercept = est_mle, color = "red", linetype = "dashed") +
  labs(title = "Replicaciones bootstrap", subtitle = "MLE: Varianza") +
  xlab("Varianza (Sigma^2)") + ylab("Densidad") +
  ggpubr::theme_pubclean(base_size = 12)

```

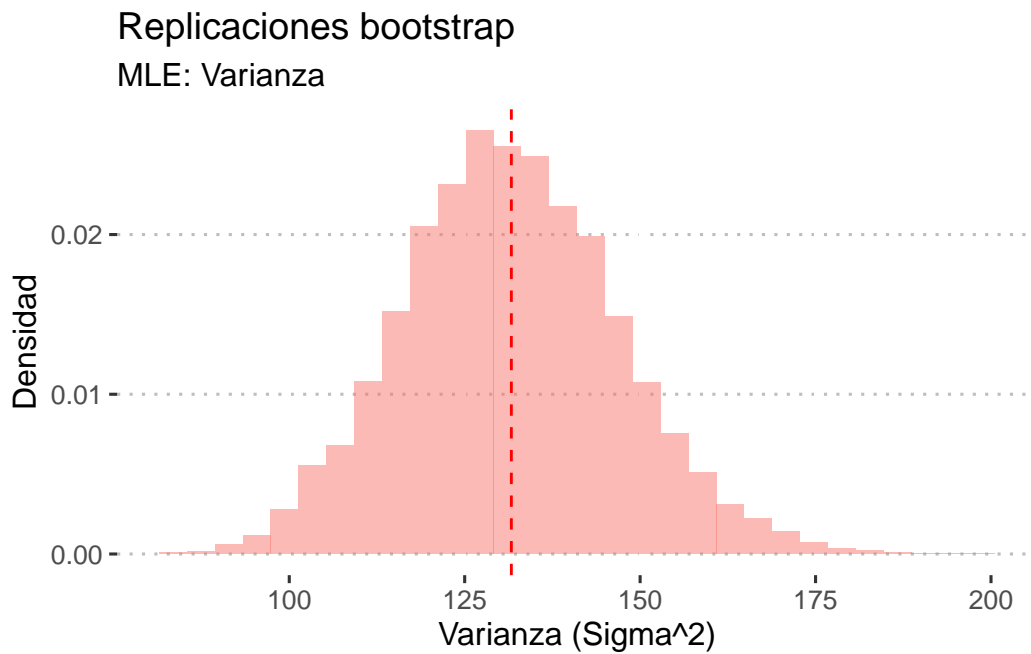


Figure 6: Histograma replications bootstrap

Ahora podemos **calcular el error estándar de nuestra estimación**

```

set.seed(156057)
error_est <- reps_boot %>% group_by(parametro) %>%

```

```
summarise(ee_boot = sd(estimador_boot))
error_est
```

```
# A tibble: 1 x 2
  parametro ee_boot
  <chr>      <dbl>
1 varianza    15.3
```

Resumiendo. Nuestro  $\sigma_{MLE}^2 = 131.65$  y su  $\hat{e}e = 15.34$

## 2.2 Análisis bayesiano

\* Continuamos con el problema de hacer inferencia de  $\sigma^2$ . Comienza especificando una inicial Gamma Inversa, justifica tu elección de los parámetros de la distribución inicial y grafica la función de densidad.

**R:** Empezamos definiendo una Gamma Inversa, los parámetros al no tener mayor contexto del problema serán de un valor bajo mostrando que es una a priori con poca información. Asimismo buscando aprovechar las colas pesadas de la distribución ya que no tenemos certeza de la cantidad de varianza del problema e.g  $\alpha = 0.05, \beta = 2$

$$f(\sigma^2) : \frac{\beta^\alpha}{\Gamma(\alpha)} * \frac{1}{(\sigma^2)^{\alpha+1}} * e^{-\frac{\beta}{\sigma^2}}$$

```
set.seed(156057)
# Calculate the corresponding alpha and beta
alpha <- 0.05 # Adjust based on your preference
beta <- 2

# Generate a range of sigma^2 values
sigma2_values <- seq(0.01, 150, by = 0.01)

# Calculate the probability densities using the Inverse Gamma density function
density_values <- actuar::dinvgamma(sigma2_values, shape = alpha, rate = beta)

# Create a data frame for ggplot2
df <- data.frame(sigma2 = sigma2_values, density = density_values)

ggplot(df, aes(x = sigma2, y = density)) +
  geom_line() +
  labs(title = "Función de densidad para la gamma inversa",
```

```

    subtitle = "InvGamma(0.05,2)",
    x = expression(sigma^2),
    y = "Densidad") +
ggpubr::theme_pubclean(base_size = 12)

```

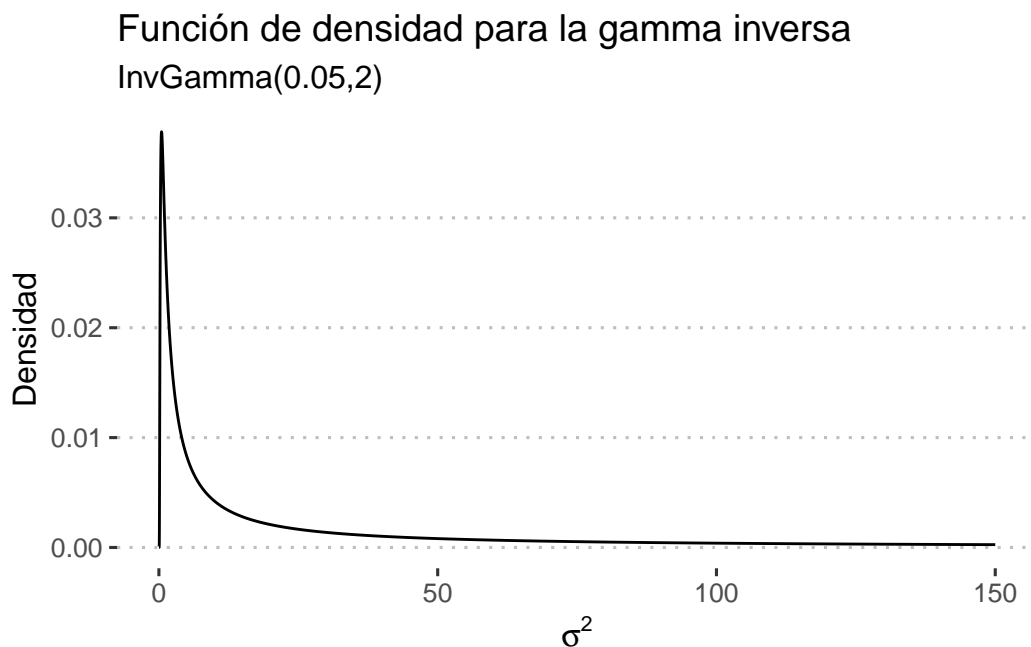


Figure 7: Distribución de inversa gamma

\* Calcula analíticamente la distribución posterior.

**R:** Sabemos que la posterior es el producto de los núcleos de la verosimilitud y de la apriori por lo que tenemos lo siguiente.

- Conocemos “n” y la suma de  $x^2$
- Conocemos  $\alpha, \beta$

$$\begin{aligned}
 P(\sigma^2|x) &= P(x|\sigma^2)P(\sigma^2) \\
 P(\sigma^2|x) &\propto \left( (\sigma^2)^{-\frac{n}{2}} e^{-\frac{\sum^n x_i^2}{2\sigma^2}} \right) \left( (\sigma^2)^{-(\alpha+1)} e^{-\frac{\beta}{\sigma^2}} \right) \\
 P(\sigma^2|x) &\propto (\sigma^2)^{-76.05} e^{-\frac{9848.82}{\sigma^2}} \\
 P(\sigma^2|x) &\sim \text{InvGamma}(75.05, 9848.82)
 \end{aligned}$$



\* Realiza un histograma de simulaciones de la distribución posterior y calcula el error estándar de la distribución.

**R:**

```
set.seed(156057)
alpha_post <- 75.05
beta_post <- 9848.82

post_samples <- 1 / rgamma(15000, shape = alpha_post, rate = beta_post)
df_posterior <- data.frame(post_samples)
a <- ggplot(df_posterior, aes(x = post_samples)) +
  geom_histogram(aes(x = post_samples, y = ..density..), bins = 30,
    fill = "#F8766D",
    alpha=0.5) +
  geom_vline(xintercept = (9848.82/74.05), color = "red", linetype = "dashed") +
  annotate("text", x = (9848.82/74.05), y = Inf, label = "E[sigma^2]: 133.00",
    vjust = 1, hjust = 0.5, colour = "red")+
  labs(title = "Simulaciones de posterior",
    subtitle = "InvGamma(75.05,9848.82)") +
  xlab("Varianza (Sigma^2)") + ylab("Densidad") +
  xlim(80,250)+
  ggpubr::theme_pubclean(base_size = 8)

b <- ggplot(reps_boot, aes(x = estimador_boot)) +
  geom_histogram(aes(x = estimador_boot, y = ..density..), bins = 30,
    fill = "#F8766D",
    alpha=0.5) +
  geom_vline(xintercept = est_mle, color = "red", linetype = "dashed") +
  annotate("text", x = est_mle, y = Inf, label = "MLE: 131.65",
    vjust = 1, hjust = 0.5, colour = "red")+
  labs(title = "Replicaciones bootstrap", subtitle = "MLE: Varianza") +
  xlab("Varianza (Sigma^2)") + ylab("Densidad") +
  xlim(80,250)+
  ggpubr::theme_pubclean(base_size = 8)

a+b
```

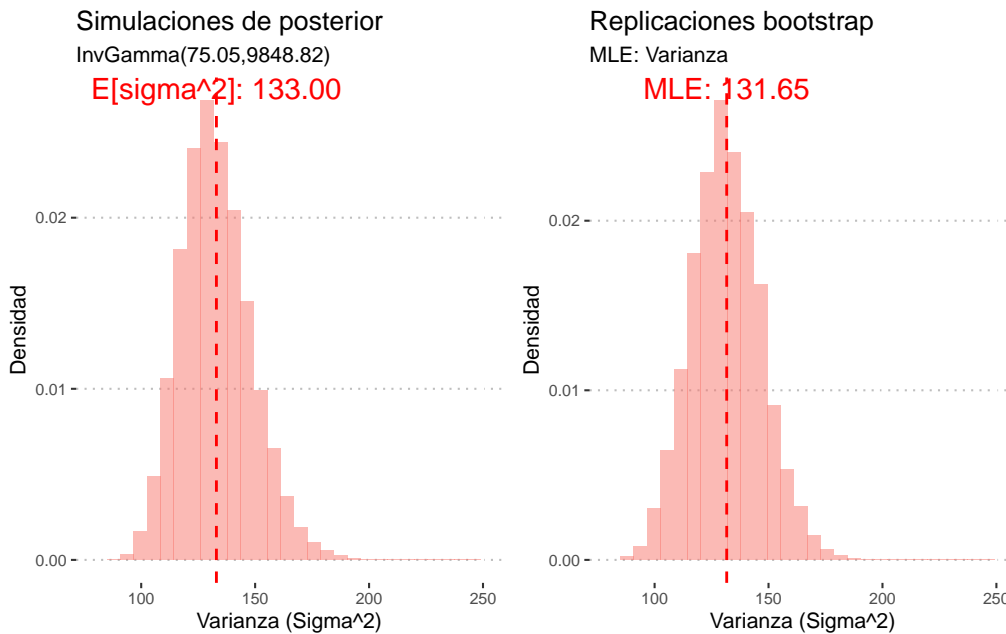


Figure 8: Comparativo Bayesiana vs. MLE

Calculamos el error estándar

```
set.seed(156057)
error_est_bayes <- df_posterior %>%
  dplyr::mutate(parametro = "varianza") %>%
  dplyr::group_by(parametro) %>%
  summarise(ee_bayes = sd(post_samples))
error_est_bayes %>% left_join(error_est)
```

Joining with `by = join\_by(parametro)`

```
# A tibble: 1 x 3
  parametro ee_bayes ee_boot
  <chr>      <dbl>   <dbl>
1 varianza    15.5    15.3
```

\* ¿Cómo se comparan tus resultados con los de bootstrap paramétrico?

**R:** En la gráfica de arriba como en la tabla viene el comparativo de los estimadores. Ponemos de nuevo el resumen

- *Bayesiana*: Calculamos el valor esperado:  $E[\sigma^2] = \frac{\beta}{\alpha-1} = 133.00$
- *Bootstrap paramétrico*: Calculamos el estimador por medio de máxima verosimilitud (i.e derivando igualando a 0)  $\sigma_{MV}^2 = 131.65$

Y los errores estándar obtenido por medio de simulaciones.

- *Bayesiana*: Distribución posterior  $InvGamma(75.05, 9848.82)$  :  $\hat{e}e = 15.459$
- *Bootstrap paramétrico*: Distribución  $Normal(0, \sigma_{MV}^2)$  :  $\hat{e}e = 15.34$

Corroboramos la correspondencia

**2.3** Supongamos que ahora buscamos hacer **inferencia del parámetro**  $\tau = \log(\sigma)$ , ¿cuál es el estimador de máxima verosimilitud?

\* Utiliza bootstrap paramétrico para generar un intervalo de confianza del 95% para el parámetro  $\tau$  y realiza un histograma de las replicaciones bootstrap.

**R**: Podemos argumentar al tratarse de una transformación logarítmica bien definida (sobre valores estrictamente positivos) que por la propiedad de **Equivarianza de MLE** que...

$$\hat{\tau} = g(\hat{\sigma}^2) = \log(\sqrt{131.65}) = 2.44$$

será el **MLE** de  $\tau$

```
reps_boot <-
  reps_boot %>% dplyr::mutate(tau_boot = log(sqrt(estimador_boot)))
ggplot(reps_boot, aes(x = tau_boot)) +
  geom_histogram(aes(x = tau_boot, y = ..density..), bins = 30,
                 fill = "#F8766D",
                 alpha=0.5) +
  geom_vline(xintercept = 2.44, color = "red", linetype = "dashed") +
  labs(title = "Replicaciones bootstrap", subtitle = "MLE: Tau") +
  xlab("Tau") + ylab("Densidad") +
  ggpubr::theme_pubclean(base_size = 12)
```

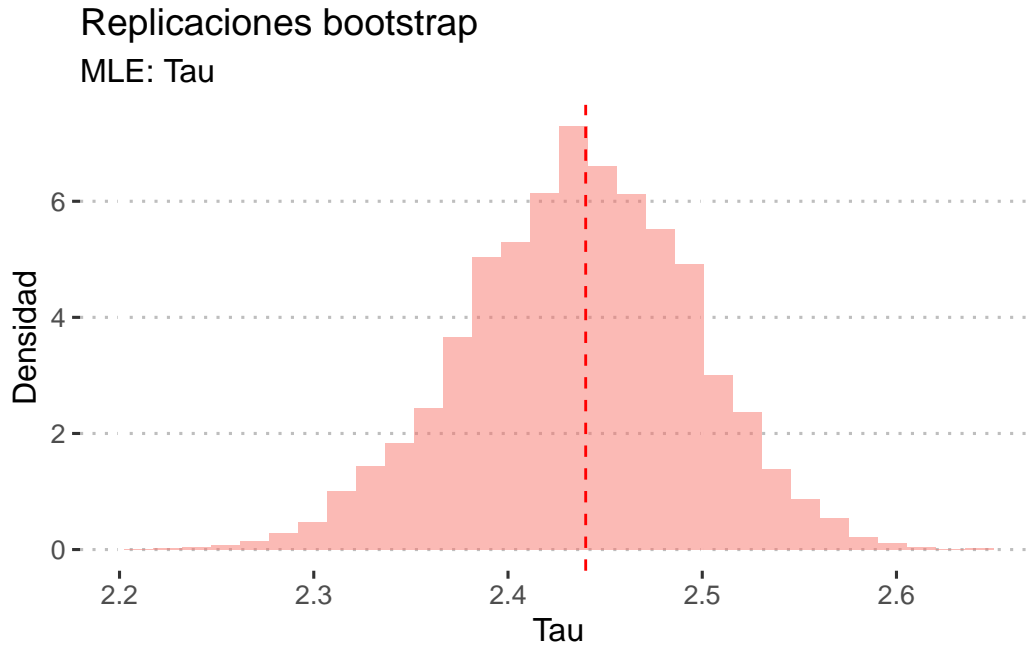


Figure 9: Histograma bootstrap: Tau

Podemos utilizar **intervalos de cuantiles** para reportar un intervalo al 95%

```
quantil_95_izq <- quantile(reps_boot$tau_boot,.025)
quantil_95_der <- quantile(reps_boot$tau_boot,.975)
print(paste0("Intervalo de confianza al 95% es : (",
             round(quantil_95_izq,3), ", ", round(quantil_95_der,3),")"))
```

```
[1] "Intervalo de confianza al 95% es : (2.317, 2.548)"
```

\* Ahora volvamos a inferencia bayesiana, calcula un intervalo de confianza para  $\tau$  y un histograma de la distribución posterior de  $\tau$ .

**R:** Dado que en Bayesiana trabajamos una vez con los datos dados es más fácil agarrar la info y hacer la transformación.

```
df_posterior <-
  df_posterior %>% dplyr::mutate(tau_bayes = log(sqrt(post_samples)))

a <- ggplot(df_posterior, aes(x = tau_bayes)) +
  geom_histogram(aes(x = tau_bayes, y = ..density..), bins = 30,
```

```

        fill = "#F8766D",
        alpha=0.5) +
geom_vline(xintercept = log(sqrt((9848.82/74.05))), color = "red", linetype = "dashed")
annotate("text", x = log(sqrt((9848.82/74.05))), y = Inf, label = "2.4452",
        vjust = 1, hjust = 0.5, colour = "red")+
labs(title = "Simulaciones de posterior",
      subtitle = "InvGamma(75.05,9848.82)") +
xlab("Tau") + ylab("Densidad") +
xlim(0,4)+
ggpubr::theme_pubclean(base_size = 8)

b <- ggplot(reps_boot, aes(x = tau_boot)) +
  geom_histogram(aes(x = tau_boot, y = ..density..), bins = 30,
    fill = "#F8766D",
    alpha=0.5) +
  geom_vline(xintercept = 2.44, color = "red", linetype = "dashed") +
  annotate("text", x = 2.44, y = Inf, label = "2.4387",
    vjust = 1, hjust = 0.5, colour = "red")+
  labs(title = "Replicaciones bootstrap", subtitle = "MLE: Tau") +
  xlab("Tau") + ylab("Densidad") +
  xlim(0,4)+
  ggpubr::theme_pubclean(base_size = 8)

a+b

```

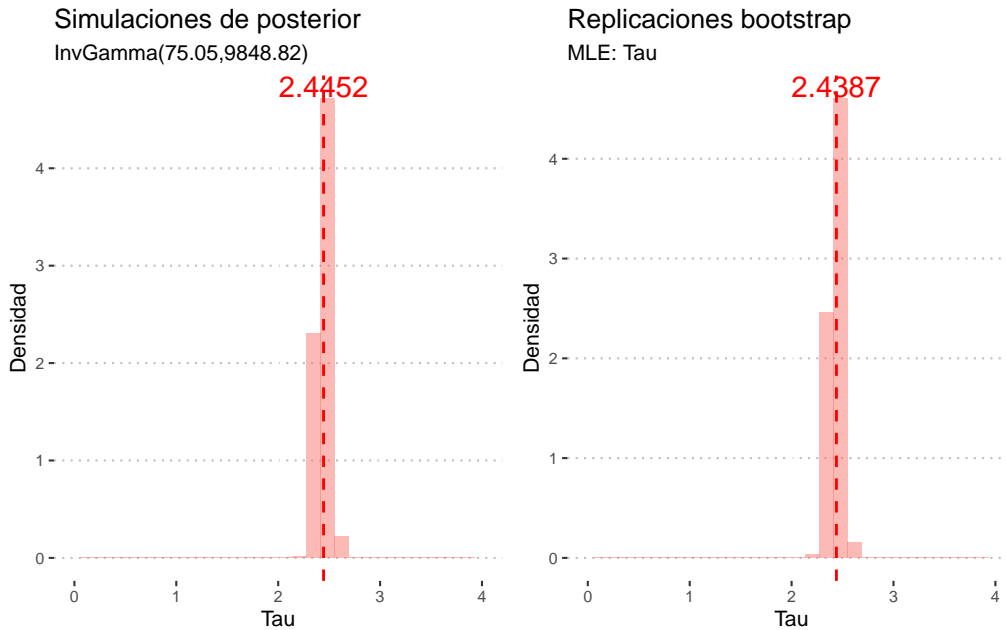


Figure 10: Comparativo Bayesiano vs. Bootstrap (Tau)

El intervalo de credibilidad para  $\tau$  es:

```
paste0("Intervalo dist. posterior: (",
       round(log(sqrt(1/qgamma(0.975, alpha_post, beta_post))),3),",",
       round(log(sqrt(1/qgamma(0.025, alpha_post, beta_post))),3),")")
```

```
[1] "Intervalo dist. posterior: (2.331,2.558)"
```

### 3. Bayesiana y regularización

Los datos `*pew_research_center_june_elect_wknd_data.dta*` tienen información de encuestas realizadas durante la campaña presidencial 2008 de EUA.

```
#poll_data <- foreign::read.dta("data/pew_research_center_june_elect_wknd_data.dta")
```

\* Estima el porcentaje de la población de cada estado (excluyendo Alaska, Hawaii, y DC) que se considera *\*very liberal\**, utilizando el estimador de máxima verosimilitud.

- Grafica en el eje *\*x\** el número de encuestas para cada estado y en el eje *\*y\** la estimación de máxima verosimilitud para *\*very liberal\**. ¿Qué observas?

- Grafica en el eje \*x\* el porcentaje de votos que obtuvo Obama en la elección para cada estado y en el eje \*y\* la estimación de máxima verosimilitud para \*very liberal\*. ¿Qué observas? (usa los datos \*2008ElectionResult.csv\*)

\* Estima el mismo porcentaje (\*very liberal\*) usando inferencia bayesiana, en particular la familia conjugada beta-binomial. Deberás estimar la proporción de manera independiente para cada estado, sin embargo, utilizarás la misma inicial a lo largo de todos:  $Beta(8, 160)$ .

- Simula de la distribución inicial y describe.
- Para dos de los estados: Idaho y Virginia, adicional a calcular la posterior usando las propiedades de la familia conjugada, utiliza Stan para hacer la inferencia, revisa los diagnósticos de convergencia y describe tus observaciones ( $\hat{R}$  y  $ESS$ ).
- Utiliza la media posterior de cada estado como estimador puntual y repite las gráficas del inciso anterior.

**\*\*Nota:\*\*** En problemas como este, donde estamos estimando un parámetro para cada grupo (estado e nuestro caso) podemos optar por un modelo jerárquico, en donde la distribución de las  $\theta_j$  no esta dada por la inicial sino que se modela con un nivel adicional, cuyos parámetros se estiman con los datos y tienen a su vez una distribución inicial:

$$y_j | \theta_j \sim \text{Binomial}(n_j, \theta_j)$$

$$\theta_j \sim \text{Beta}(\alpha, \beta)$$

$$\alpha \sim g(a_o), \beta \sim f(b_o)$$

donde  $g(a_o)$  y  $f(b_o)$  son las iniciales seleccionadas con conocimiento experto.