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 θ donde θ 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))
```

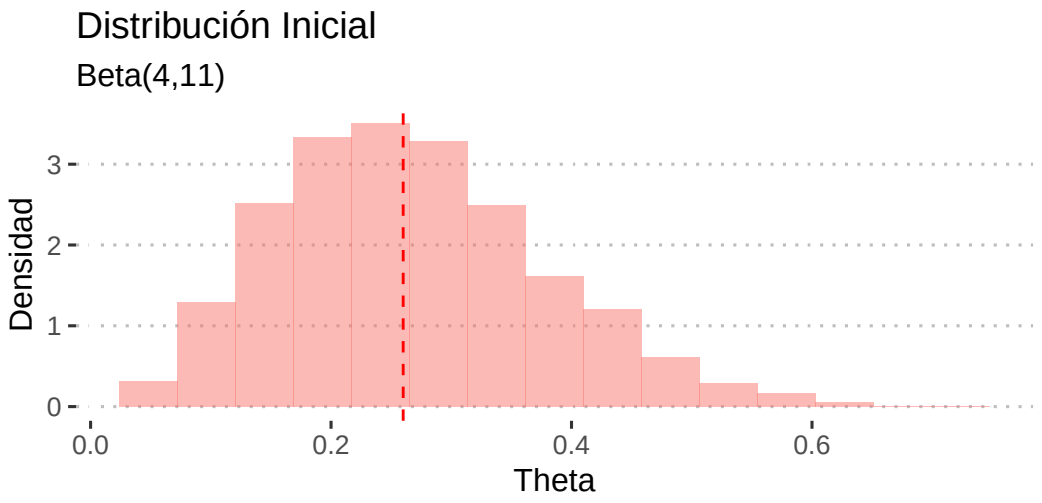


Figure 1: Distribución a priori: Beta (4,11)

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)
```

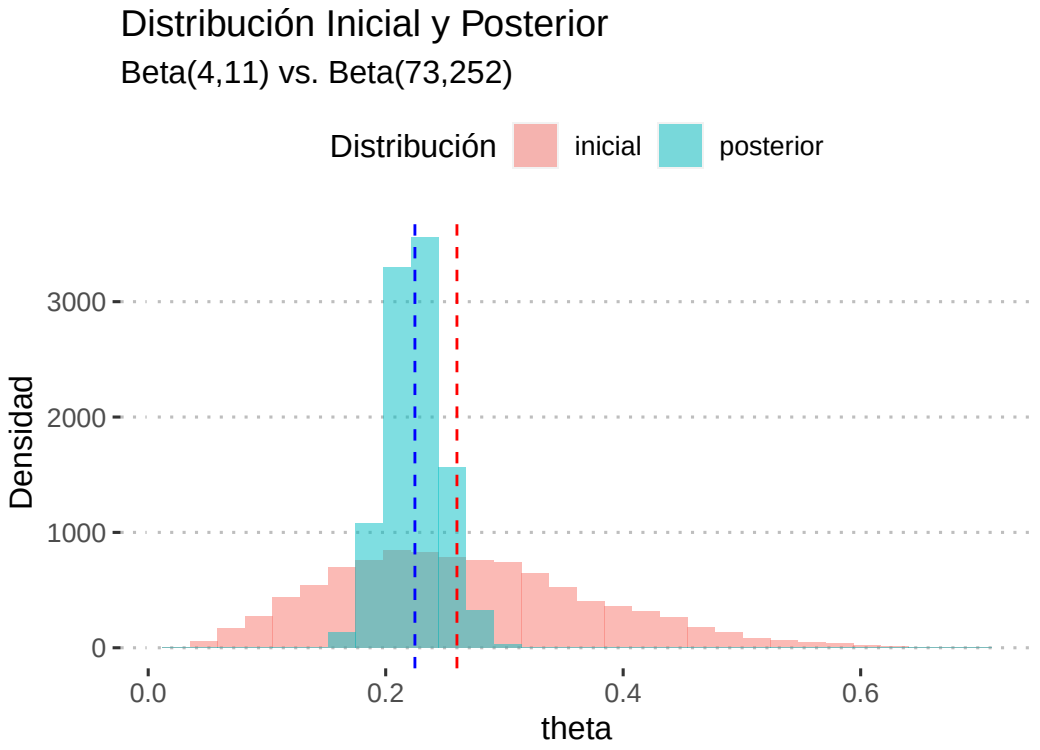


Figure 2: Inicial vs. Posterior

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: amarrillas 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 hipotesis 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)))
```

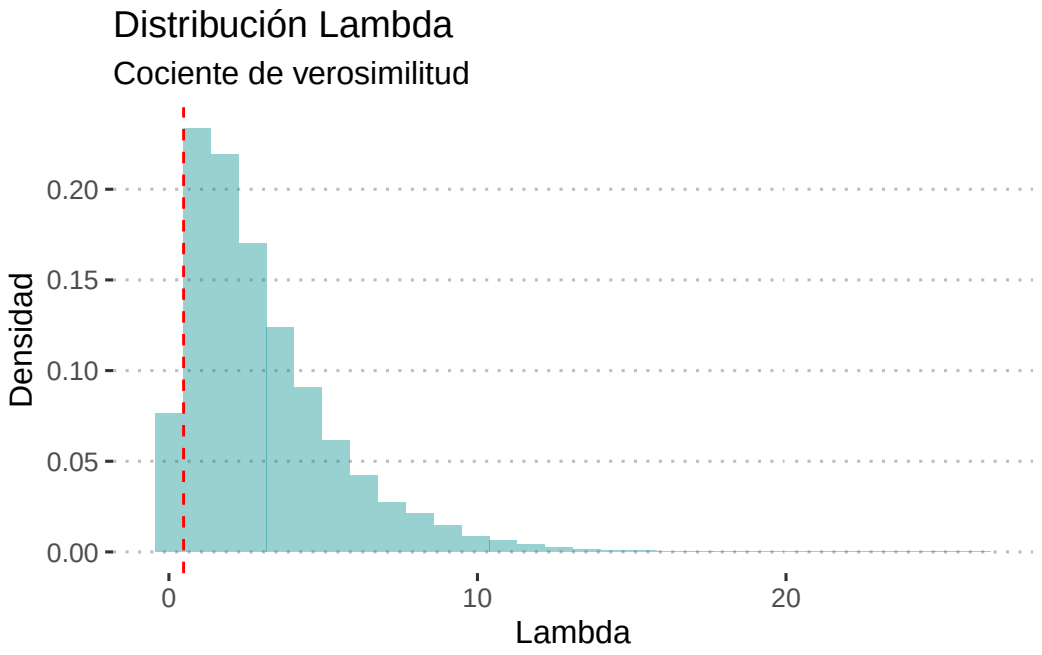


Figure 3: Distribución Lambda

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 encontra 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 $X_1, ...X_n \sim Poisson(\lambda)$,

* 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 $X_1, ...X_n \sim Poisson(\lambda_0)$ 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 λ de una Poisson será la media. Asimismo hemos visto resultados asintóticos donde la media tiene normalidad asintótica dicho esto podemos declarar que

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

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

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

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 λ es estimado con la media, el $\hat{ee}$ 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

```
resume_wald %>%
  group_by(Rechazo=rechazo) %>%
  summarise(Conteo=n(), Proporción =Conteo/1000) %>%
   kable(format='latex',
         booktabs=T,
         caption = 'Proporción de rechazos de H0 con 1000 simulaciones') %>%
  kableExtra::kable_styling(latex_options = c('hold_position'))
```

Table 2: Proporción de rechazos de H0 con 1000 simulaciones

Rechazo	Conteo	Proporción
0	930	0.93
1	70	0.07

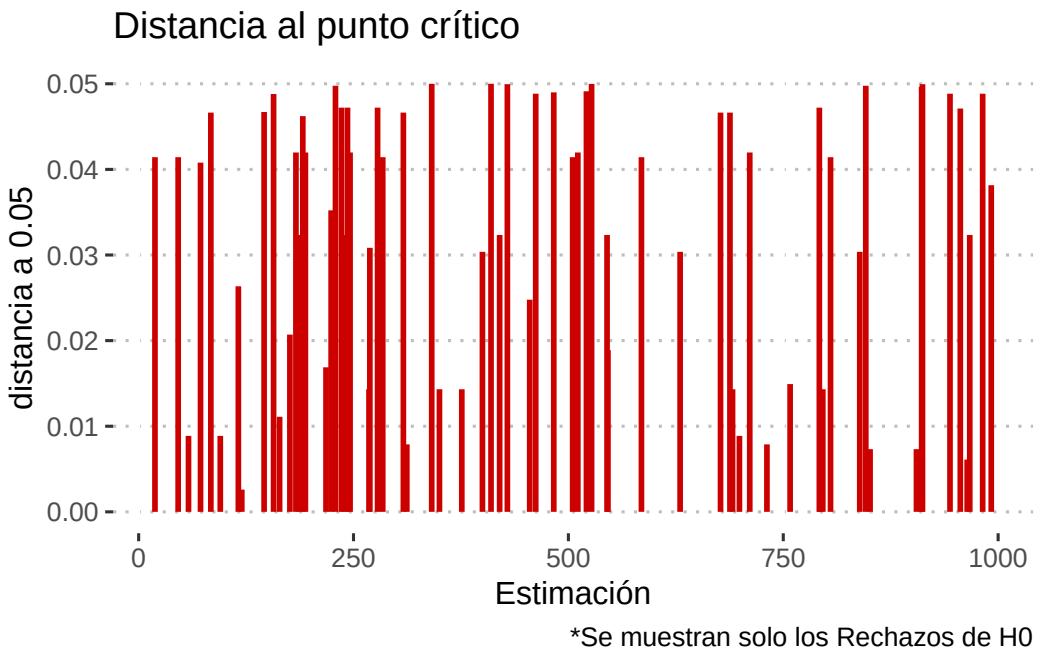


Figure 4: Distancia entre el error del tipo I y 5%

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 τ, más se concentra la posterior en el estimador de máxima verosimilitud $\hat{\theta} = x$. En el límite, cuando $\tau \to \infty$ 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-informartiva 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 $x_1, ..., x_n \sim N(0, \sigma^2)$, 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 σ^2.

2.1 Bootstrap paramétrico.

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

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

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

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"

```r
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 $\sigma^2_{mv} = 131.291$. **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")
```

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

	estimador
Varianza	131.65

Graficamos:

```
dat_verosim <- tibble(x = seq(5,300, 0.01)) %>%
  mutate(log_prob = map_dbl(x, log_p))
```

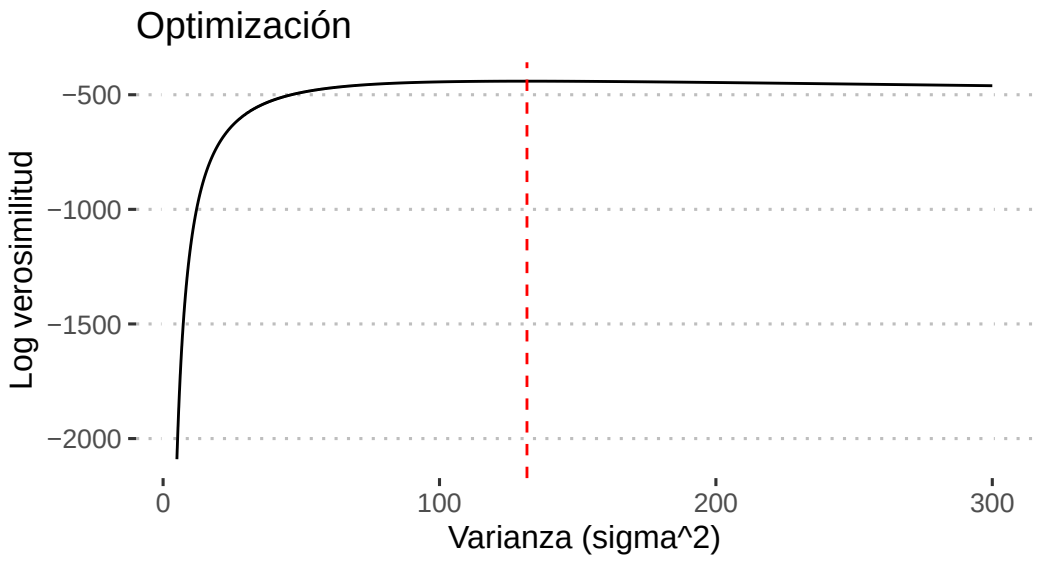
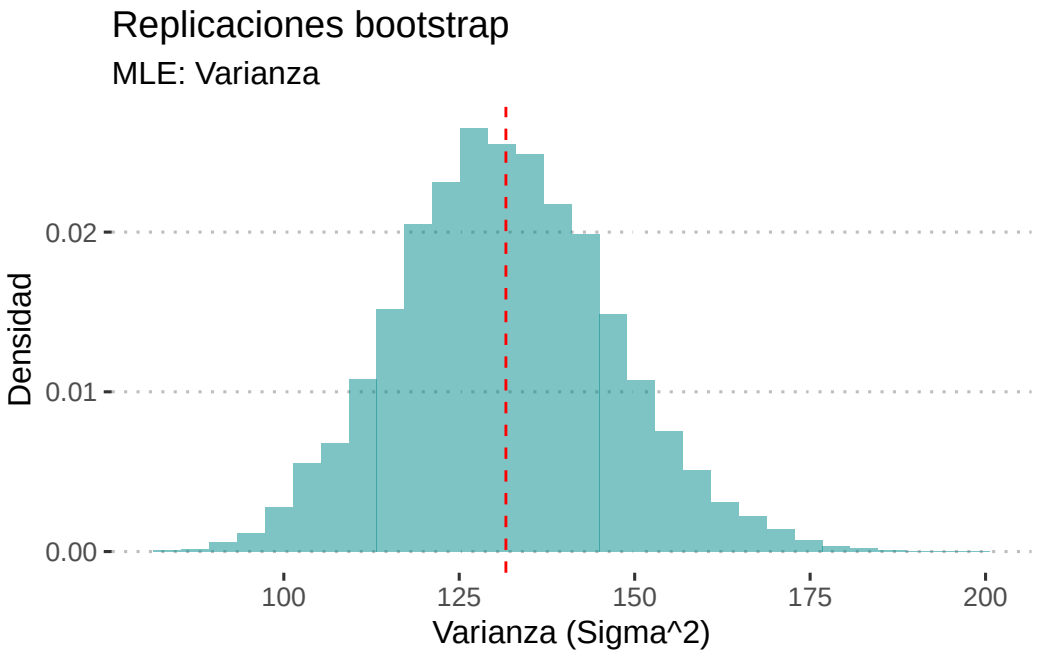


Figure 5: Optimización varianza

- Aproxima el error estándar de la estimación usando *bootstrap paramétrico* y realiza un histograma de las replicaciones bootstrap.

Respuesta: Creamos nuestro flujo de generador de muestra (utilizando el parámetro de máxima verosimilitud), cálculamos 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")
```



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))
```

parametro	ee_boot
varianza	15.33985

Resumiendo. Nuestro $\sigma^2_{MLE} = 131.65$ y su $\hat{ee} = 15.34$

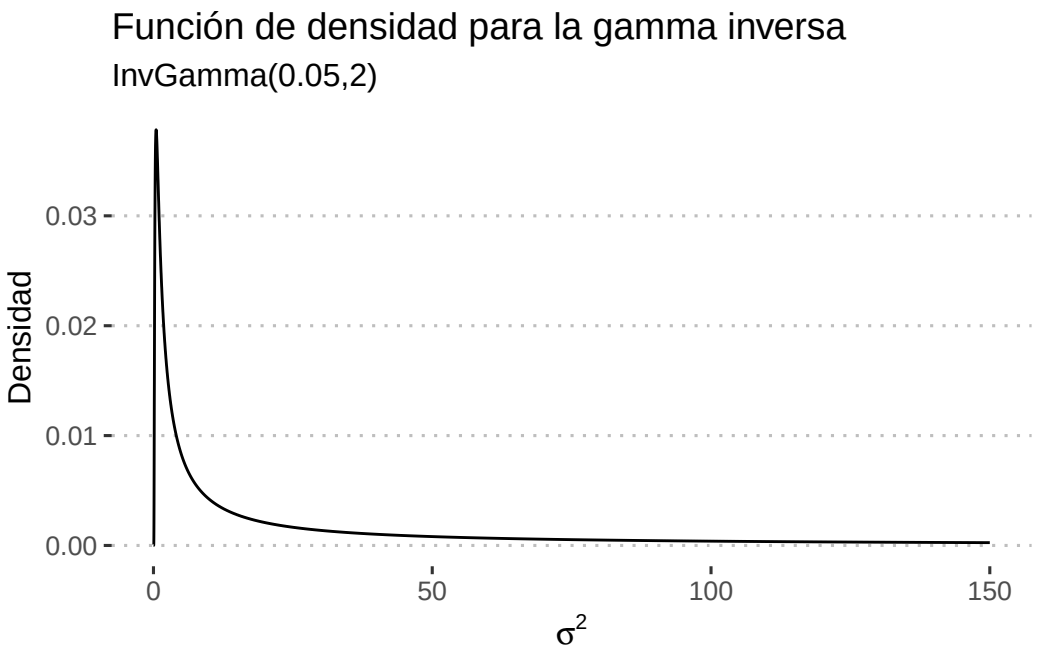
2.2 Análisis bayesiano

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

Respuesta: 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 $\alpha = 0.05, \beta = 2$

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

```r
set.seed(156057)

alpha <- 0.05
beta <- 2

# generamos distintos valores de sigma
sigma2_values <- seq(0.01, 150, by = 0.01)

# Calc densidad con inverse-gamma
density_values <- actuar::dinvgamma(sigma2_values, shape = alpha, rate = beta)

# data frame para graficar
df_sigma <- data.frame(sigma2 = sigma2_values, density = density_values)
```



- Calcula analíticamente la distribución posterior.

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

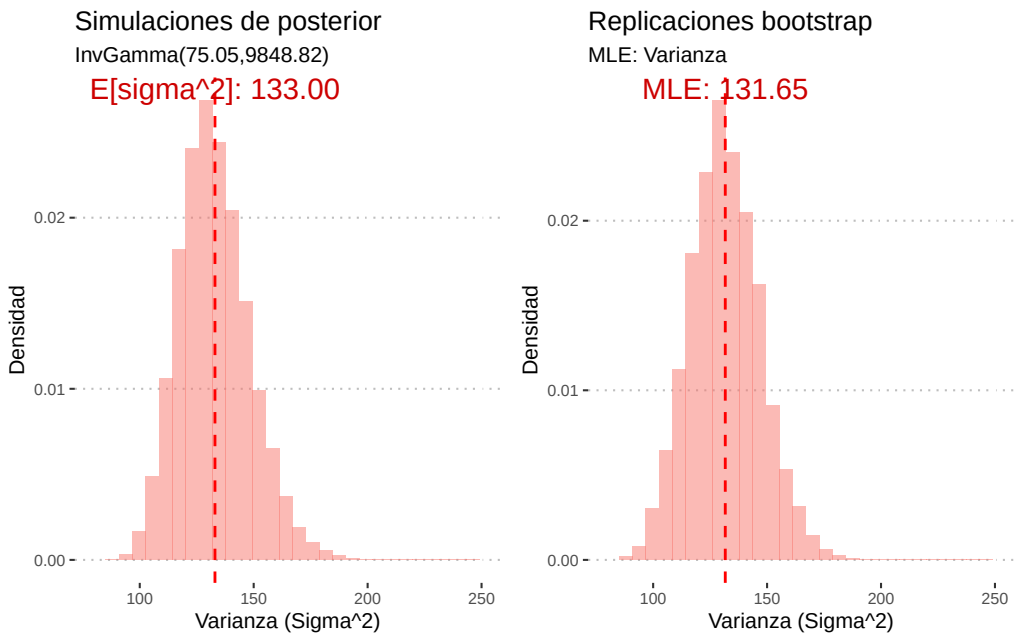
- Conocemos "n" y la suma de x^2
- Conocemos α, β

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

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

Respuesta:

```
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)
```



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))
```

parametro	ee_bayes	ee_boot
varianza	15.45913	15.33985

- ¿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) $\sigma^2_{MV} = 131.65$

Y los errores estándar obtenido por medio de simulaciones.

- *Bayesiana*: Distribución posterior $InvGamma(75.05, 9848.82)$: $\hat{ee} = 15.459$

- *Bootstrap paramétrico*: Distribución $Normal(0, \sigma^2_{MV})$: $\hat{ee} = 15.34$

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 τ y realiza un histograma de las replicaciones bootstrap.

R: Podemos argumentar al tratarse de una transformación logaritmica bien definida (sobre valores estricatamente 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 τ

```
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)
```

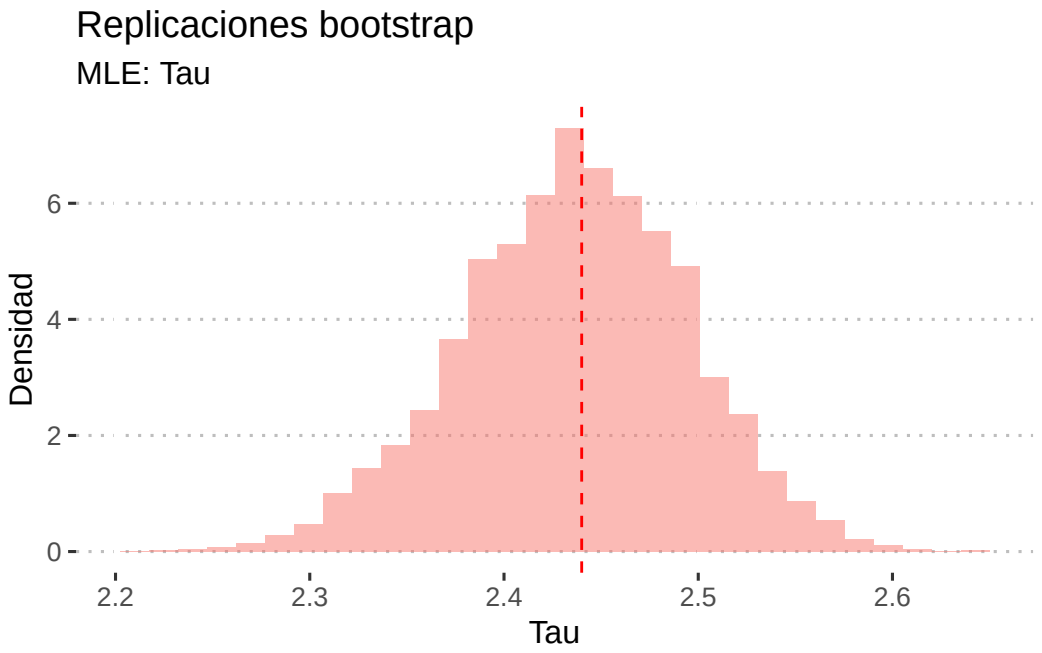


Figure 6: Histograma bootstrap: Tau

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 τ y un histograma de la distribución posterior de τ.

R: Dado que en Baysesiana 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
```

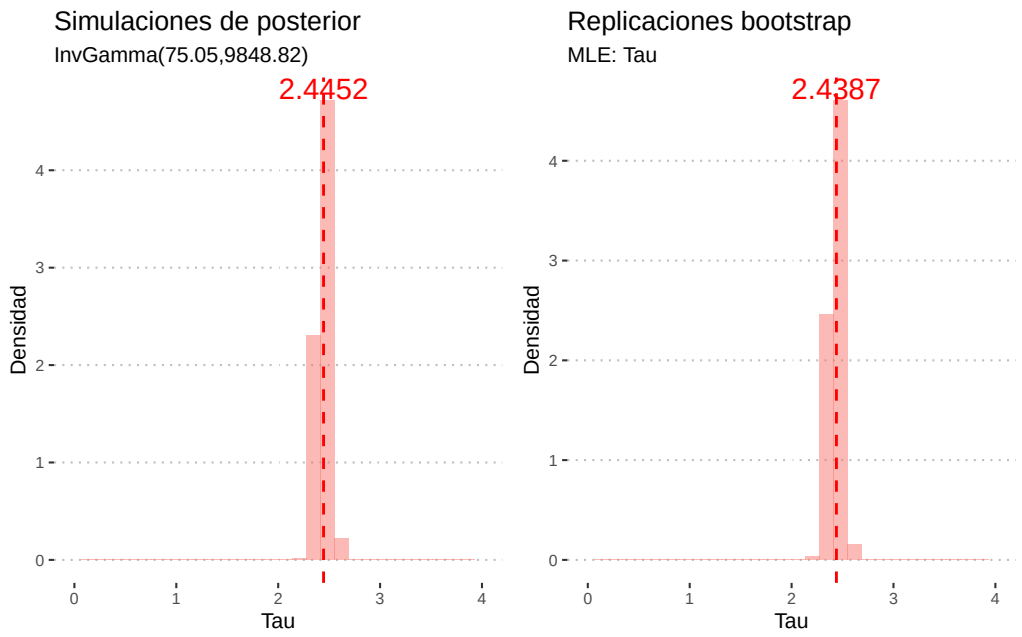


Figure 7: Comparativo Bayesiano vs. Bootstrap (Tau)

El intervalo de credibilidad para τ 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.

```
set.seed(156057)
data <- foreign::read.dta("data/pew_research_center_june_elect_wknd_data.dta")
```

NOTA: para cada respuesta se presenta solo un ejemplo del data.frame, al final como anexo se agrega la tabla para los 48 estados considerados.

A falta de más información se decidió tomar la columna *weight* como factor de expanción para las encuestas por lo que el total de población por estado será la suma de esta columna. Para el total de encuestas se considero que cada registro (renglón) era una, por lo que el total de encuestas es el conteo de registros por estado.

- Estima el porcentaje de la población de cada estado (excluyendo Alaska, Hawai, 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?

R: Excluimos los estados mencionados y utilizamos la variable *weight* como factor de expansión para realizar los calculos del porcentaje de *very liberal*. Además cada registro lo tomamos en cuenta como si fuera 1 encuesta, entonces entendemos por número de encuestas como número de registros agrupados por la variable *state*. El estimador de **máxima verosimilitud** para una proporción, *very liberal* como se ha demostrado varias veces en clase y en la pregunta 2 será $\frac{x_{very.liberal}}{n}$

```
set.seed(156057)
# Consideramos que la población total de mi estado es la suma de "weight"
very_liberal <-
  data %>%
  # Sin alaska, hawaii, washington dc
  dplyr::filter(!(state %in% c("washington dc","hawaii","alaska"))) %>%
  dplyr::group_by(state, ideo) %>%
  dplyr::summarise(poblacion = sum(weight),
                   poblacion_fix = n()) %>%
  tidyr::pivot_wider(id_cols = state, names_from = ideo, values_from = poblacion) %>%
  ungroup()
```

```
`summarise()` has grouped output by 'state'. You can override using the
`.groups` argument.
```

```
# Población total = suma de weight; # de encuestas = conteo de renglones
poblacion_encuestas <-
  data %>%
  # Sin alaska, hawaii, washington dc
  dplyr::filter(!(state %in% c("washington dc","hawaii","alaska"))) %>%
  dplyr::group_by(state) %>%
  dplyr::summarise(poblacion = sum(weight),
                   encuestas = n()) %>% ungroup()
```

Table 3: Head por estado con MLE

state	perc_very_liberal	encuestas
alabama	7.31	624
arizona	6.42	542
arkansas	1.75	307
california	6.34	2854
colorado	6.20	468
connecticut	2.99	395

```
tabla_aux <- dplyr::left_join(very_liberal, poblacion_encuestas, by = "state")

tabla_aux <- tabla_aux %>%
  dplyr::mutate(perc_very_liberal = round((`very liberal`/poblacion)*100,2))

kable(tabla_aux %>%
      dplyr::select(state, perc_very_liberal, encuestas) %>% head(),
      caption = "Head por estado con MLE",
      digits = 4,
      format = "latex",
      booktabs = T) %>%
  kable_styling(latex_options = c("striped"),
                bootstrap_options = c("striped", "hover", "condensed", "responsive"),
                full_width = F, fixed_thead = T)
```

Graficamos... Observamos una **ligera relación positiva: a mayor número de encuestas mayor el porcentaje de población que se identifican con la ideología de very liberal**. Esto es de llamar la atención, no deberían de depender nuestros resultados del número de encuestas podría haber problemas con el muestro. Sin embargo, al estar hablando de porcentajes "relativamente bajos" el encuestar podría llegarse a ver esta relación.

```
set.seed(156057)
ggplot(tabla_aux, aes(x = encuestas, y = perc_very_liberal)) +
  geom_point() +
  geom_text_repel(aes(label = state), box.padding = 0.5, size = 1.5) +
  #geom_text(aes(label = state), hjust = 1.5, vjust = 0.5, size = 1.5) +
  geom_smooth(method = "lm", se = FALSE, color = "#F8766D", linetype = "dashed") +
  labs(title = "Num. Encuestas vs. Pct Very Liberal") +
  xlab("Num Encuestas") + ylab("Pct Very Liberal (MLE)") +
```

```
    ylim(0,12) +
    ggpubr::theme_pubclean(base_size = 12)
```

```
`geom_smooth()` using formula = 'y ~ x'
```

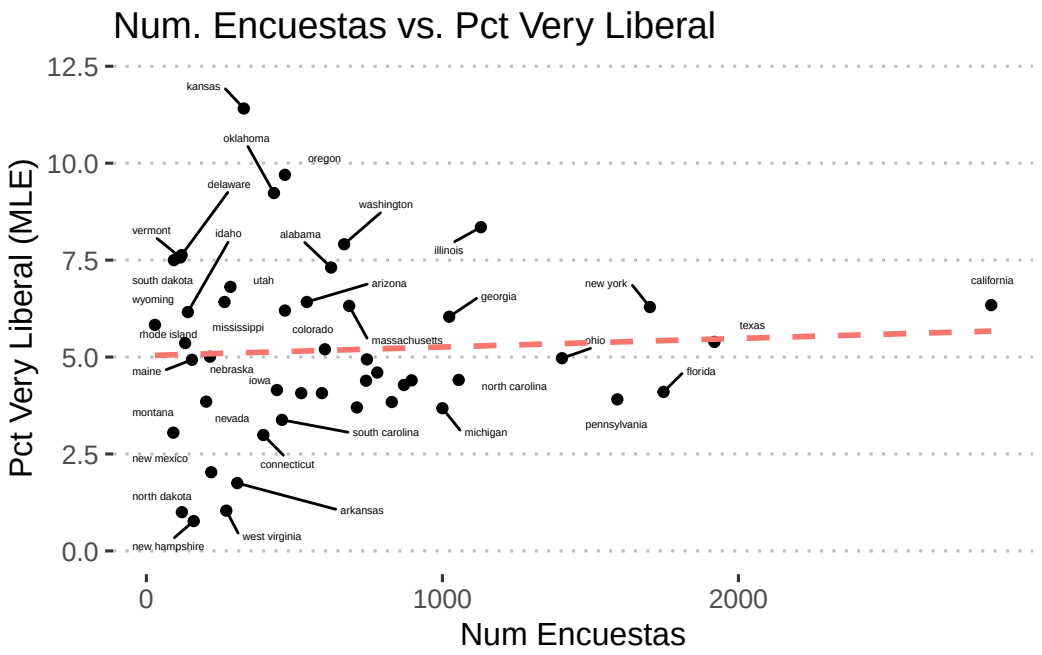


Figure 8: Pct Very Liberal vs. Num. Encuestas (archivo DTA)

- 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*)

R: Realizamos la unión de basos de datos

```
set.seed(156057)
data_xls <- data.table::fread("data/2008ElectionResult.csv") %>%
  dplyr::select(state, vote_Obama_pct)
tabla_aux <- dplyr::mutate(tabla_aux, state = toupper(state))
data_xls <- dplyr::mutate(data_xls, state = toupper(state))

tabla_aux <- left_join(tabla_aux, data_xls, by = "state")

kable(tabla_aux %>%
```

Table 4: Head por estado con MLE + Obama

state	perc_very_liberal	encuestas	vote_Obama_pct
ALABAMA	7.31	624	38.8
ARIZONA	6.42	542	45.0
ARKANSAS	1.75	307	38.8
CALIFORNIA	6.34	2854	60.9
COLORADO	6.20	468	53.5
CONNECTICUT	2.99	395	60.5

```
      dplyr::select(state, perc_very_liberal, encuestas, vote_Obama_pct) %>%
      head(),
      caption = "Head por estado con MLE + Obama",
      digits = 4,
      format = "latex",
      booktabs = T) %>%
  kable_styling(latex_options = c("striped"),
                bootstrap_options = c("striped", "hover", "condensed", "responsive"),
                full_width = F, fixed_thead = T)
```

Graficamos... Observamos una **ligera relación negativa: a mayor número de votantes de Obama menor el porcentaje de población que se identifica con la ideología de very liberal**. Esto hace sentido ya que Obama en general fue percibido como un candidato de *centro-izquierda* más que alguien que representará la ideología muy liberal

```
set.seed(156057)
ggplot(tabla_aux, aes(x = vote_Obama_pct, y = perc_very_liberal)) +
  geom_point() +
  geom_text_repel(aes(label = state), box.padding = 0.5, size = 1.2) +
  #geom_text(aes(label = state), hjust = 1.5, vjust = 0.5, size = 1.5) +
  geom_smooth(method = "lm", se = FALSE, color = "#F8766D", linetype = "dashed") +
  labs(title = "Pct Obama vs. Pct Very Liberal") +
  xlab("Pct Obama") + ylab("Pct Very Liberal (MLE)") +
  ylim(0,12) +
  ggpubr::theme_pubclean(base_size = 12)
```

```
`geom_smooth()` using formula = 'y ~ x'
```

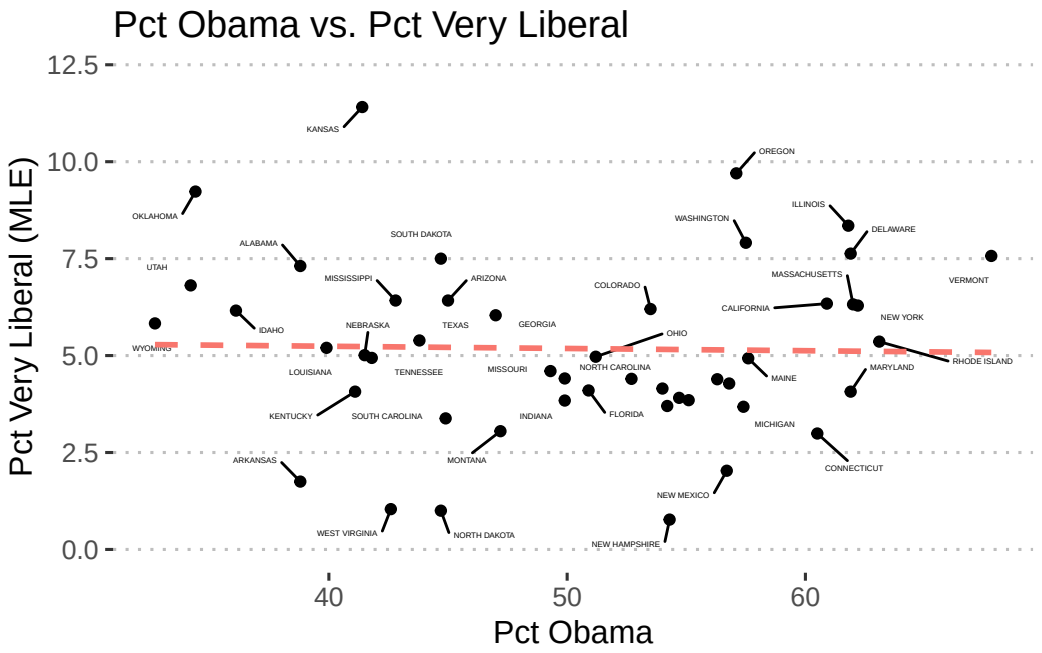


Figure 9: MLE vs. Pct de votantes por Obama

- 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: $Beta(8, 160)$.

R: Al tener un modelo beta-binomial sabemos que la **posterior tendrá una distribución Beta**. Previo a establecer una función general para la posterior realizamos simulaciones de la inicial para ver que información previa tenemos.

$$p(\theta) \propto \theta^{8-1}(1-\theta)^{160-1}$$

Como era de esperar tenemos una **media cercana al 5%. No es simétrica ya que considera los posibles escenarios que cubren las colas largas** para porcentajes más altos.

```
set.seed(156057)
sim_inicial <- tibble(theta = stats::rbeta(10000,8,160))
ggplot(sim_inicial) +
  geom_histogram(aes(x = theta, y = ..density..), bins = 15,
                 fill = "#F8766D",
                 alpha=0.5) +
  geom_vline(xintercept = (8/168), color = "red") +
```

```
annotate("text", x = (8/168), y = Inf, label = "Media",
         vjust = 1, hjust = 0.5,colour = "red")+
labs(title = "Distribución Inicial", subtitle = "Beta(8,160)") +
xlab("Theta") + ylab("Densidad") +
ggpubr::theme_pubclean(base_size = 12)
```

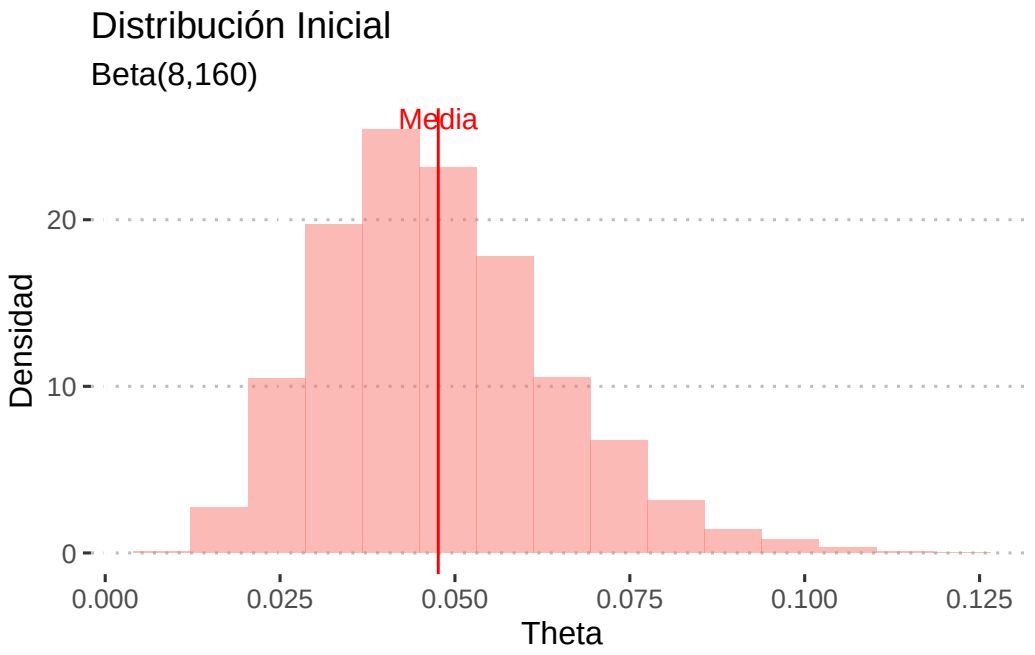


Figure 10: Distribución a priori: Beta (8,160)

La posterior para el *estado i* estará dada por:

$$P(\theta|X) \propto P(X|\theta)P(\theta)$$
$$P(\theta|X) \propto \theta^{k_i+7}(1-\theta)^{n_i+159} \sim Beta(k_i+8, n_i+160)$$

Donde:

- k_i: es el número de individuos identificados con la ideología *very liberal*
- n_i: total de la población

Y nuestro estimador puntual será la media posterior, es decir: $\frac{k_i+8}{(k_i+8)+(n_i+160)}$

Dejamos código pero la tabla se muestra hasta el final (El total de estados con sus estimadores)

Table 5: Head por estado con estimaciones (+Bayesiano)

state	perc_very_liberal	bayesiano	encuestas
ALABAMA	7.31	6.63	624
ARIZONA	6.42	5.90	542
ARKANSAS	1.75	2.30	307
CALIFORNIA	6.34	5.94	2854
COLORADO	6.20	5.72	468
CONNECTICUT	2.99	3.17	395

```
set.seed(156057)
for (i in 1:nrow(tabla_aux)) {
  alpha <- tabla_aux$`very liberal`[i] + 8
  beta <- tabla_aux$poblacion[i] + 160
  tabla_aux$bayesiano[i] <- round((alpha/(alpha+beta)*100),2)
}


kable(tabla_aux %>%
      dplyr::select(state, perc_very_liberal, bayesiano, encuestas) %>%
      head(),
      caption = "Head por estado con estimaciones (+Bayesiano)",
      digits = 4,
      format = "latex",
      booktabs = T) %>%
  kable_styling(latex_options = c("striped"),
                bootstrap_options = c("striped", "hover", "condensed", "responsive"),
                full_width = F, fixed_thead = T)
```

- 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).

R: Realizamos estimaciones vía *Stan*. Dado que es la misma inicial un mismo modelo nos funciona para ambos Estados.

```
set.seed(156057)
archivo_stan <- file.path("stan/modelo_preg3.stan")
# compilar
mod <- cmdstan_model(archivo_stan)
```

```
mod
```

```
// The input data is a vector 'y' of length 'N'.
data {
  int n; // número de individuos
  int y; // número de seguidores para very liberal
}

parameters {
  real<lower=0,upper=1> theta; // parámetro de interés
}

// The model to be estimated.
model {
  // inicial
  theta ~ beta(8, 160);
  y ~ binomial(n, theta);
}

generated quantities {
  real theta_inicial;
  theta_inicial = beta_rng(8, 160);
}
```

Pasamos datos, muestreamos y revisamos convergencia

IDAHO

```
n <- dplyr::filter(tabla_aux, state == "IDAHO")$poblacion
y <- dplyr::filter(tabla_aux, state == "IDAHO")$`very liberal`
datos_lista <- list(n = n, y = y)
ajuste <- mod$sample(
  data = datos_lista,
  seed = 156057,
  chains = 4,
  iter_warmup = 5000,
  iter_sampling = 20000,
  parallel_chains = 4,
  show_messages = F)
ajuste$cmdstan_diagnose()
```

```
Processing csv files: C:/Users/mario/AppData/Local/Temp/Rtmpg9EJ46/modelo_preg3-202312032159-

Checking sampler transitions treedepth.
Treedepth satisfactory for all transitions.

Checking sampler transitions for divergences.
No divergent transitions found.

Checking E-BFMI - sampler transitions HMC potential energy.
E-BFMI satisfactory.

Effective sample size satisfactory.

Split R-hat values satisfactory all parameters.

Processing complete, no problems detected.
```

```r
idaho <- ajuste$summary() %>%
  dplyr::mutate(state = "Idaho") %>%
  dplyr::select(state,variable, mean, sd, rhat, ess_bulk, ess_tail)
```

VIRGINIA

```r
n <- dplyr::filter(tabla_aux, state == "VIRGINIA")$poblacion
y <- dplyr::filter(tabla_aux, state == "VIRGINIA")$`very liberal`
datos_lista <- list(n = n, y = y)
ajuste <- mod$sample(
  data = datos_lista,
  seed = 156057,
  chains = 4,
  iter_warmup = 5000,
  iter_sampling = 20000,
  parallel_chains = 4,
  show_messages = F)
ajuste$cmdstan_diagnose()
```

```
Processing csv files: C:/Users/mario/AppData/Local/Temp/Rtmpg9EJ46/modelo_preg3-202312032159-

Checking sampler transitions treedepth.
Treedepth satisfactory for all transitions.
```

Table 6: Estimaciones via Stan para Idaho y Virginia

state	variable	mean	sd	rhat	ess_bulk	ess_tail
Idaho	lp___	-126.9621	0.7103	1.0000	35612.33	41426.59
Idaho	theta	0.0572	0.0096	1.0000	30507.51	34989.49
Idaho	theta_inicial	0.0477	0.0165	1.0000	79460.34	79388.52
Virginia	lp___	-451.7294	0.7154	1.0001	35564.62	43272.79
Virginia	theta	0.0440	0.0041	1.0001	29322.89	37589.43
Virginia	theta_inicial	0.0477	0.0165	1.0000	80329.04	79109.27

```
Checking sampler transitions for divergences.
No divergent transitions found.

Checking E-BFMI - sampler transitions HMC potential energy.
E-BFMI satisfactory.

Effective sample size satisfactory.

Split R-hat values satisfactory all parameters.

Processing complete, no problems detected.
```

```
virginia <- ajuste$summary() %>%
  dplyr::mutate(state = "Virginia") %>%
  dplyr::select(state,variable, mean, sd, rhat, ess_bulk, ess_tail)
```

Observamos que el valor de la posterior tiene efectos diferentes entre ambos estados. Para Idaho la media posterior aumenta, pasando de 0.047 a 0.057 y para Virginia disminuye ligeramente de 0.047 a 0.043. Los valores de *rhat* muestran un buen ajuste

```
stan_resumen <- rbind(idaho, virginia)
kable(stan_resumen,
      caption = "Estimaciones via Stan para Idaho y Virginia",
      digits = 4,
      format = "latex",
      booktabs = T) %>%
  kable_styling(latex_options = c("striped"),
                bootstrap_options = c("striped", "hover", "condensed", "responsive"),
                full_width = F, fixed_thead = T)
```

- Utiliza la media posterior de cada estado como estimador puntual y repite las gráficas del inciso anterior.

R: Realizamos las mismas gráficas; sin embargo, ahora utilizamos la media posterior como estimadores puntuales.

La relación encontrada se mantiene respecto a la relación entre número de encuestas y el porcentaje estimado de la población con ideología *very liberal*

```
set.seed(156057)
tabla_aux_plot <-
  tabla_aux %>%
  dplyr::select(state, perc_very_liberal, vote_Obama_pct, bayesiano, encuestas) %>%
  dplyr::rename(MLE=perc_very_liberal) %>%
  tidyr::pivot_longer(
    cols = c(MLE, bayesiano),
    names_to = "Metodologia",
    values_to = "value"
  )
ggplot(tabla_aux_plot, aes(x = encuestas, y = value, color = Metodologia)) +
  geom_point() +
  geom_text_repel(aes(label = state), box.padding = 0.5, size = 1.5) +
  #geom_text(aes(label = state), hjust = 1.5, vjust = 0.5, size = 1.5) +
  #geom_smooth(method = "lm", se = FALSE, color = Metodologia) +
  geom_smooth(method = "lm", se = FALSE,
              aes(group = Metodologia, color = Metodologia),
              linetype = "dashed") +
  labs(title = "Num. Encuestas vs. Pct Very Liberal", subtitle = "MLE vs. Bayesiano") +
  xlab("Num. Encuestas") + ylab("Pct Very Liberal") +
  ylim(0,12) +
  ggpubr::theme_pubclean(base_size = 12)
```

```
`geom_smooth()` using formula = 'y ~ x'
```

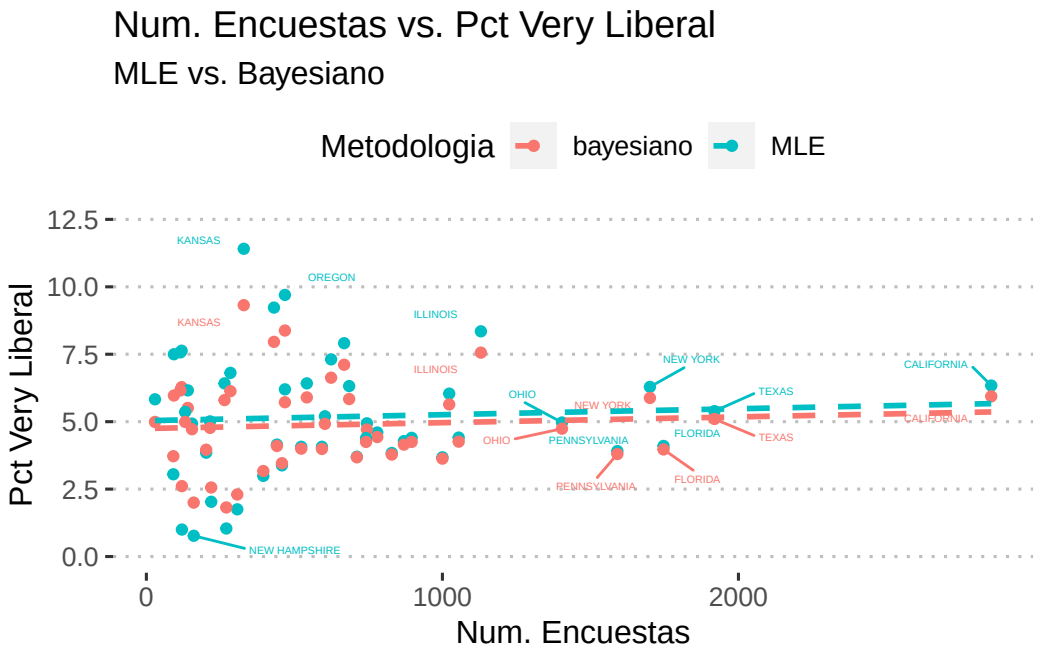


Figure 11: Bayes-MLE y Num. Encuestas

La relación encontrada se mantiene respecto a la relación entre porcentaje de población votante por Obama y el porcentaje estimado de la población con ideología *very liberal*. Un factor a considerar es que en esta gráfica es mucho más notorio que la estimación bayesiana en probabilidades bajas obtenidas por MLE son un poco mayores y probabilidades relativamente más altas por MLE ahora son un poco menores.

```
ggplot(tabla_aux_plot, aes(x = vote_Obama_pct, y = value, color = Metodologia)) +
  geom_point() +
  geom_text_repel(aes(label = state), box.padding = 0.5, size = 1.5) +
  #geom_text(aes(label = state), hjust = 1.5, vjust = 0.5, size = 1.5) +
  #geom_smooth(method = "lm", se = FALSE, color = Metodologia) +
  geom_smooth(method = "lm", se = FALSE,
              aes(group = Metodologia, color = Metodologia),
              linetype = "dashed") +
  labs(title = "Num Encuestas vs. Pct Very Liberal", subtitle = "MLE vs. Bayesiano") +
  xlab("Pct Obama") + ylab("Pct Very Liberal") +
  ylim(0,12) +
  ggpubr::theme_pubclean(base_size = 12)
```

```
`geom_smooth()` using formula = 'y ~ x'
```

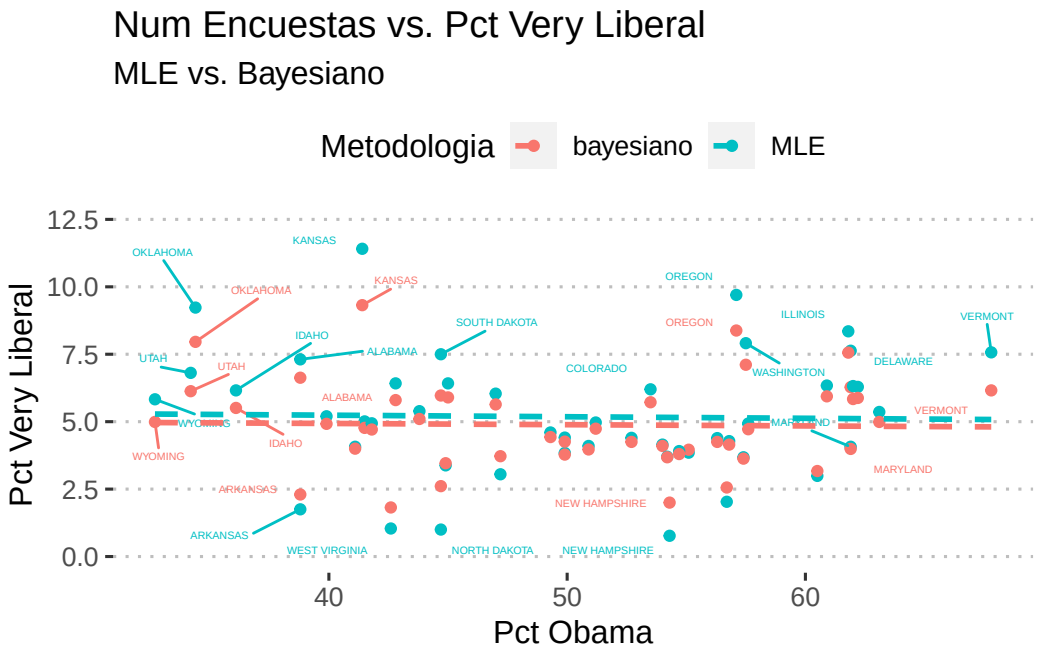


Figure 12: Bayes vs. MLE y Pct Obama

Apéndice

Tabla con todas las estimaciones para todos los estados.

```
kable(tabla_aux %>%
        dplyr::select(state, perc_very_liberal, bayesiano) %>%
        dplyr::rename(MLE = perc_very_liberal, Bayesiano = bayesiano),
      caption = "Estimaciones por estado",
      digits = 4,
      format = "latex",
      booktabs = T) %>%
  kable_styling(latex_options = c("striped"),
                bootstrap_options = c("striped", "hover", "condensed", "responsive"),
                full_width = F, fixed_thead = T)
```

Table 7: Estimaciones por estado

state	MLE	Bayesiano
ALABAMA	7.31	6.63
ARIZONA	6.42	5.90
ARKANSAS	1.75	2.30
CALIFORNIA	6.34	5.94
COLORADO	6.20	5.72
CONNECTICUT	2.99	3.17
DELAWARE	7.63	6.28
FLORIDA	4.10	3.97
GEORGIA	6.04	5.64
IDAHO	6.16	5.51
ILLINOIS	8.35	7.56
INDIANA	3.84	3.78
IOWA	4.15	4.10
KANSAS	11.41	9.32
KENTUCKY	4.07	4.00
LOUISIANA	5.20	4.92
MAINE	4.93	4.72
MARYLAND	4.07	3.99
MASSACHUSETTS	6.32	5.84
MICHIGAN	3.68	3.63
MINNESOTA	3.70	3.68
MISSISSIPPI	6.42	5.80
MISSOURI	4.60	4.43
MONTANA	3.05	3.72
NEBRASKA	5.01	4.77
NEVADA	3.85	3.96
NEW HAMPSHIRE	0.77	2.00
NEW JERSEY	4.28	4.15
NEW MEXICO	2.03	2.56
NEW YORK	6.29	5.88
NORTH CAROLINA	4.41	4.26
NORTH DAKOTA	1.00	2.61
OHIO	4.97	4.74
OKLAHOMA	9.23	7.96
OREGON	9.70	8.38
PENNSYLVANIA	3.91	3.80
RHODE ISLAND	5.36	4.99
SOUTH CAROLINA	3.38	3.46
SOUTH DAKOTA	7.50	5.97
TENNESSEE	4.94	4.71
TEXAS	5.39	5.10
UTAH	6.81	6.13
VERMONT	7.57	6.16
VIRGINIA	4.40	4.25
WASHINGTON	7.91	7.11
WEST VIRGINIA	1.04	1.82