

# Examen Final

*Dalia Camacho y Gabriela Vargas*

*December 4, 2018*

```
library(tidyverse)
library(knitr)
```

## 3. MCMC

Siguiendo con el conteo rápido de Guanajuato, calcularás intervalos de confianza usando el modelo propuesto en @mendoza2016.

Los autores proponen ajustar un modelo de manera independiente para cada candidato en cada estrato:

- Verosimilitud

$$X_{ij}^k | \theta_{ij}, \tau_{ij} \sim N\left(n_i^k \theta_{ij}, \frac{\tau_{ij}}{n_i^k}\right)$$

para  $k = 1, \dots, c_i$ ,  $i = 1, \dots, N$ ,  $j = 1, \dots, J$

- Iniciales

$$p(\theta_{i,j}, \tau_{ij}) \propto \tau_{ij}^{-1} I(\tau_{ij} > 0) I(0 < \theta_{i,j} < 1)$$

- Posterior

$$p(\theta_{ij}, \tau_{ij} | X_{ij}) \sim N\left(\theta_{ij} \left| \frac{\sum_{k=1}^{c_i} x_{ij}^k}{\sum_{k=1}^{c_i} n_i^k}, \tau_{ij} \sum_{k=1}^{c_i} n_i^k \right| I(0 < \theta_{ij} < 1) \times Ga\left(\tau_{ij} \left| \frac{c_i - 1}{2}, \frac{1}{2} \left[ \sum_{k=1}^{c_i} \frac{(x_{ij}^k)^2}{n_i^k} - \frac{(\sum_{k=1}^{c_i} x_{ij}^k)^2}{\sum_{k=1}^{c_i} n_i^k} \right] \right| \right)$$

donde:

- $X_{ij}$  número de personas que favorecen al candidato  $j$  en el estrato  $i$ .
- $X_{ij}^k$  número de personas que favorecen al candidato  $j$  en la casilla  $k$  del estrato  $i$ .
- $n_i^k$  tamaño de la lista nominal en la  $k$ -ésima casilla del  $i$ -ésimo estrato.
- $\tau_{ij}/n_i^k$  es la precisión para cada candidato.
- $\theta_{ij}$  es la proporción de las personas en la lista nominal del estrato  $i$  que favorecen al  $j$ -ésimo partido.
- $c_i$  número de casillas del  $i$ -ésimo estrato en la muestra.

Los detalles del modelo los puedes encontrar en el artículo.

Implementa el modelo y estima los resultados electorales de Guanajuato con la muestra:

```
gto_muestra <- read.csv("data/muestra_gto_2012.csv", stringsAsFactors = FALSE)

# Eliminamos el estrato con una sola casilla para evitar que la posterior sea impropia
gto_muestra <- gto_muestra %>% filter(distrito_fed_17!=20)
```

```

# Obtener nik
nik <- gto_muestra %>% group_by(casilla_id, distrito_fed_17) %>%
  summarise("nik"=sum(ln_total))
nik <- nik %>% ungroup() %>%
  mutate("casilla_id"=as.numeric(as.factor(casilla_id))) %>%
  mutate("distrito_fed_17"=as.numeric(as.factor(distrito_fed_17)))

nik <- nik %>% spread(key =casilla_id, value = nik, fill = 0) %>% select(-"distrito_fed_17") %>% as.ma

gto_muestraVotos <- gto_muestra %>% gather(key="partido", value = "votos",
  pri_pvem, pan_na, prd, pt, mc, otros)

# Número de estratos
Ni <- length(unique(gto_muestra$distrito_fed_17))

# Número de candidatos
Nj <- length(unique(gto_muestraVotos$partido))

# Número de casillas
Nk <- length(unique(gto_muestraVotos$casilla_id))

# Número de casillas por estrato
ci <- gto_muestra %>% group_by(distrito_fed_17) %>% summarise("ci"=n()) %>%
  select("ci") %>% unlist()

# Datos observados
Xijk <- gto_muestraVotos %>% group_by(partido, distrito_fed_17, casilla_id) %>%
  summarise("Xijk"=sum(votos)) %>%
  ungroup() %>%
  mutate("partido"=as.numeric(as.factor(partido))) %>%
  mutate("casilla_id"=as.numeric(as.factor(casilla_id))) %>%
  mutate("distrito_fed_17"=as.numeric(as.factor(distrito_fed_17)))

num_votos_estrato <- gto_muestraVotos %>% group_by(distrito_fed_17) %>%
  summarise(total= sum(total)) %>% ungroup() %>%
  select(-"distrito_fed_17") %>% as.matrix()

# Inicial no informativa para theta if(thetaij >0 & thetaij<1)

sumatorias <- function(i,j, nik, Xijk){

  # Vector de Xs para estrato i y candidato j
  Xs <- Xijk %>% filter(distrito_fed_17==i & partido==j) %>% select("Xijk") %>% unlist()

  # Vector de tamaño de la lista nominal en la casilla k
  ns <- nik[i,]
  ns <- ns[which(ns>0)]
  ci <- length(ns)

  # Precalcular sumas de la distribución
  sum1 <- sum(Xs)/sum(ns)
  sum2 <- sum(ns)
  sum3 <- sum(Xs^2/ns)
  sum4 <- sum(Xs)^2/sum(ns)

```

```

sums<-c(ci,sum1,sum2,sum3,sum4)

return(sums)
}

```

Simulaciones de distribución posterior

1. Para obtener  $\tau_{ij}$  generamos variables aleatorias de una distribución Gamma.
2. Condicional al valor de  $\tau_{ij}$ , generamos una variable aleatoria con distribución normal para  $\theta_{ij}$ .
3. Si  $0 < \theta_{ij} < 1$ , consideramos la dupla  $(\theta_{ij}, \tau_{ij})$ . En caso contrario, la dupla se descarta.

```

sim_params <- function(i,j, ln, votos, nsims){

  thetaij1 <- runif(1,0,1)
  tauij1 <- runif(1,0,50)
  aux <- sumatorias(i,j,ln,votos)
  a <- (aux[1]-1)/2
  b <- 1/2*(aux[4]-aux[5])
  media <- aux[2]
  thetas <- c()
  taus <- c()
  i <- 0

  while (i < nsims) {
    tauij.2 <- rgamma(1,a,b)
    sd <- sqrt(1/(tauij.2*aux[3]))
    thetaij.2 <- rnorm(1,media,sd)

    if(thetaij.2>0 & thetaij.2<1){
      thetaij1 <- thetaij.2
      tauij1 <- tauij.2
      i <- i + 1

      thetas <- c(thetas, thetaij1)
      taus <- c(taus, tauij1)
    }
  }

  return(list(Mean_Theta=mean(thetas), Thetasij=thetas, Tausij=taus))
}

```

Simulamos las thetas por candidato en el i-ésimo estrato.

```

SimEstrCand <- data.frame(matrix(ncol=7,nrow=0))
names(SimEstrCand) <- c("Estrato","Candidato","Mean_Theta","Thetasij","Tausij", "ni", "sims")

for (i in 1:Ni){
  for(j in 1:Nj){
    s <- sim_params(i, j, nik, Xijk, 1000)
    aux <- data.frame("Estrato"=rep(i,1000), "Candidato"=rep(j,1000),
                      "Mean_Theta"=rep(s$Mean_Theta,1000), "Thetasij"=s$Thetasij,
                      "Tausij" =s$Tausij, "ni"=rep(sum(nik[i,]),1000), "sims"=1:1000)
    SimEstrCand <- rbind(SimEstrCand,aux)
  }
}

```

```

}
}

thetajs <- SimEstrCand %>% group_by(Estrato, Candidato) %>%
  summarise(Mean_Theta= mean(Mean_Theta), ni=unique(ni)) %>%
  ungroup() %>% group_by(Candidato) %>%
  summarise(thetaj = sum(Mean_Theta*ni/sum(ni)))

lambdajs <- thetajs %>%ungroup()%>%
  mutate(lambdajs=thetaj/sum(thetaj))
lambdajs

```

```

## # A tibble: 6 x 3
##   Candidato thetaj lambdajs
##   <int>     <dbl>     <dbl>
## 1         1 0.00515  0.00852
## 2         2 0.0217   0.0360
## 3         3 0.290     0.479
## 4         4 0.0298   0.0494
## 5         5 0.251     0.415
## 6         6 0.00708  0.0117

```

Reporta estimaciones puntuales (media posterior) e intervalos del 95% de credibilidad para cada candidato

```

thetajsIC <- SimEstrCand %>% group_by(Estrato, Candidato, sims) %>%
  summarise(Mean_Theta= Thetasij, ni=unique(ni)) %>%
  ungroup() %>% group_by(Candidato, sims) %>%
  summarise(thetaj = sum(Mean_Theta*ni/sum(ni)))

lambdajsIC <- thetajsIC %>%ungroup()%>%
  group_by(sims) %>%
  mutate(lambdajs=thetaj/sum(thetaj)) %>% ungroup() %>%
  group_by(Candidato) %>% summarise(MeanLambda=mean(lambdajs),
                                     Lambda025 = quantile(lambdajs,0.025),
                                     Lambda975 = quantile(lambdajs, 0.975))

lambdajsIC

```

```

## # A tibble: 6 x 4
##   Candidato MeanLambda Lambda025 Lambda975
##   <int>     <dbl>     <dbl>     <dbl>
## 1         1  0.00852  0.00660  0.0105
## 2         2  0.0360   0.0340  0.0382
## 3         3  0.479     0.470   0.489
## 4         4  0.0494   0.0456  0.0528
## 5         5  0.415     0.406   0.424
## 6         6  0.0117   0.0107  0.0128

```

```

lambdajsIC %>% mutate(Candidato= unique(gto_muestraVotos$partido)[order(unique(gto_muestraVotos$partido))])
  mutate("IC"=paste0(formatC(MeanLambda, digits = 3, format = "f"), " ( ",
                             formatC(Lambda025, digits = 3, format = "f"), ", ",
                             formatC(Lambda975, digits = 3, format = "f"), ") ")) %>%
  select(c("Candidato", "IC")) %>% kable()

```

Candidato	IC
mc	0.009 ( 0.007, 0.011)
otros	0.036 ( 0.034, 0.038)

Candidato	IC
pan_na	0.479 ( 0.470, 0.489)
prd	0.049 ( 0.046, 0.053)
pri_pvem	0.415 ( 0.406, 0.424)
pt	0.012 ( 0.011, 0.013)