

# handout4

Jorge Fábrega

2023-09-10

## Teoría espacial del voto

### *Concepto principal*

\* Primeras ideas: Hotelling (1929)

+ Modelo de Hotelling: Estudió la ubicación óptima de una tienda en una “ciudad lineal”.

+ Conclusión: la ubicación óptima es la mediana de la ciudad.

- Anthony Downs (1957): “An Economic Theory of Democracy”.
- Extiende el modelo de Hotelling al ámbito político.
- Los votantes se distribuyen en una dimensión de elección (ej. carga tributaria para redistribución).
- A esa dimensión de elección se le puede atribuir una interpretación ideológica
- Poole y Rosenthal (2000): Congress: A Political-economic History of Roll Call Voting
- Fuerte evidencia empírica a favor de la unidimensionalidad

### *Sobre el concepto de ideología*

La ideología puede definirse pero no puede observarse directamente. Se puede decir que la acción X o la opinión Y son consistentes o propias de una ideología Z, pero la ideología propiamente tal es una abstracción que busca sintetizar a un conjunto más o menos armónico de creencias, valores y reglas o normas que un individuo o un grupo posee sobre las formas de organizar la vida en común. Existen numerosos estudios y miradas que se aproximan al concepto de ideología y hay muchos test conocidos para clasificar a personas según sus opiniones en una escala de izquierda a derecha o en mapas bidimensionales como el Political Compass.

### *La aproximación empírica de la teoría espacial del voto*

La aproximación empírica de la teoría espacial del voto es agnóstica respecto de qué es y qué no es un pensamiento o una acción “de izquierda” o “de derecha” o “liberal” o “conservadora”, etc... Simplemente, busca modos de estimar a partir de comportamiento una variable latente que explique (o sea consistente) con dicho comportamiento observado. Por ello, todo análisis empírico desde la teoría espacial del voto debe ser contrastado con conocimiento experto (o general) en alguna dimensión sobre la cual existe suficiente consenso valorativo y que forme parte del material con el que se hizo la estimación. Por ejemplo, una vez estimada la ideología de un grupo de personas que opinaron sobre temas económicos y regulatorios que indagaba sobre el rol del Estado y el Mercado, debería constatar que aquellos que quedaron ubicados más a la izquierda según la estimación expresaron mayor valor o importancia a acciones de regulación o producción estatal y aquellos que quedaron ubicados más a la derecha se inclinaron por posturas más proclives a mercados autorregulados. Si ello no sucede y si revirtiendo el orden de las estimaciones no se corrige, entonces, es muy probable que la estimación no esté generando resultados confiables.

La aplicación empírica de la teoría espacial del voto parte de las siguientes premisas:

+ Las preferencias políticas pueden representarse en un “espacio ideológico” abstracto. + Los individuos

votan/opinan/se comportan de forma tal que eligen al candidato/el tema/la situación política más cercana a su posición o creencia. Y, por ende, los que votan/piensan/actúan de forma más parecida estarán más cerca ideológicamente entre sí y alejados de los que votan/piensan/actúan diferente frente a los mismos estímulos.

## Veamos un ejemplo: El Match

- El match constituyente de La Tercera
- ¿Cuál es el problema con el Match Constituyente en su versión original? [Link](#)
- El match electoral (parlamentario) [Link](#)

## Formalización de las ideas anteriores

### *Puntos Ideales y Funciones de Utilidad*

Punto Ideal: Resultado más preferido por el votante.

Función de Utilidad: Cantidad de satisfacción que un votante obtiene de diferentes políticas.

+ Ejemplos:  $U_i = -|x - z_i|$  o  $U_i = -(x - z_i)^2$

### *Tipos de Funciones de Utilidad*

Cuadrática: Más común, permite descomposición media-varianza.

Normal (Gaussiana): Mejor aproxima las funciones de utilidad de los legisladores (evidencia para USA).

Lineal (Distancia Absoluta): Más simple pero menos utilizada.

## NOMINATE

El desarrollo del procedimiento de escalamiento NOMINATE por Keith T. Poole y Howard Rosenthal marcó un hito, proporcionando estimaciones fiables de las posiciones ideológicas de los miembros del Congreso a lo largo de la historia de EE. UU. Este avance impulsó un interés considerable en la estimación de puntos ideales y técnicas de escalamiento.

El método NOMINATE (Nominal Three-Step Estimation) se basa en un modelo probabilístico para predecir cómo un miembro de un cuerpo colegiado votará en una votación particular. Aunque hay varias versiones del método, veamos el modelo básico.

### *Modelo Formal simplificado*

Supongamos que tenemos  $n$  legisladores y  $m$  votaciones. Cada legislador  $i$  tiene un punto ideal  $X_i$  en un espacio  $k$ -dimensional (generalmente  $k = 1$  o  $k = 2$ ). Además, cada votación  $j$  se representa mediante un vector  $Y_j$  en el mismo espacio  $k$ -dimensional y un parámetro de corte  $\beta_j$  que es perpendicular al vector  $Y_j$  (y que sirve para definir si se vota a favor o en contra).

La probabilidad  $P_{ij}$  de que el legislador  $i$  vote “sí” en la votación  $j$  se modela como:

$$P_{ij} = \frac{1}{1 + \exp(-\alpha_j d_{ij})}$$

donde  $\alpha_j$  es un parámetro que captura la discriminación de la votación  $j$ , y  $d_{ij}$  es la distancia euclidiana entre el punto ideal del legislador  $X_i$  y el punto de corte  $\beta_j$  para la votación  $j$ .

$$d_{ij} = \|X_i - \beta_j\|$$

## Estimación en el modelo simple

Los parámetros  $X_i, Y_j, \beta_j$ , y  $\alpha_j$  se estiman mediante la maximización de la función de verosimilitud, que se basa en las votaciones observadas. La función de verosimilitud  $L$  para todos los legisladores y todas las votaciones es:

$$L = \prod_{i=1}^n \prod_{j=1}^m P_{ij}^{V_{ij}} (1 - P_{ij})^{(1-V_{ij})}$$

donde  $V_{ij}$  es 1 si el legislador  $i$  votó “sí” en la votación  $j$  y 0 en caso contrario.

El objetivo es encontrar los parámetros que maximizan esta función de verosimilitud. Esto se hace generalmente mediante algoritmos de optimización numérica. Una vez estimados los parámetros, se pueden utilizar para analizar las posiciones ideológicas de los legisladores, predecir votaciones futuras, y estudiar la estructura subyacente de un cuerpo legislativo. Veamos un ejemplo simplificado del proceso que hace NOMINATE.

## Ejemplo

```
library(dplyr)
library(ggplot2)

n <- 10 # votantes

# Generamos datos ficticios de un cuerpo colegiado de n personas que
# han votado 50 veces
set.seed(123)
puntos_ideals <- seq(-10, 10, length.out = n)
vote_positions <- sample(-10:10, 50, replace = TRUE)
votacion <- expand.grid(Person_ID = 1:10, Vote_ID = 1:50)
votacion$Vote_Result <- ifelse(abs(puntos_ideals[votacion$Person_ID] -
                                vote_positions[votacion$Vote_ID]) <= 2,
                                1,
                                0)

# Creamos un dataframe para almacenar los resultados de las iteraciones
ideals_df <- data.frame(Iteration = integer(),
                        Person_ID = integer(),
                        Ideal = numeric())

# Función para calcular la prob. de un voto dado un punto ideal y una
# posición de votación
calc_prob <- function(ideal, position) {
  1 / (1 + exp(-abs(ideal - position)))
}

# Aquí vamos a hacer f iteraciones
f <- 3 # rango de variación inicial para simular iteraciones para ir viendo el
#proceso de convergencia

# Iteración 1
# 1. Inicialización: Estimación inicial de los puntos ideales y parámetros
ideals_est <- puntos_ideals
positions_est <- vote_positions

# 2. Cálculo de Probabilidades
votacion$Prob <- mapply(calc_prob,
                        ideals_est[votacion$Person_ID],
                        positions_est[votacion$Vote_ID])
```

```

# Almacenar los resultados de la iteración 1
ideals_df <- rbind(ideals_df,
                  data.frame(Iteration = 1,
                             Person_ID = 1:n,
                             Ideal = ideals_est))

# 3. Aquí debería ir la Maximización de la Verosimilitud de NOMINATE, pero
# hagamos una versión simplificada en la que incrementamos (sin justificación
# alguna) en 0.1 los puntos ideales y la posición estimada.
ideals_est <- ideals_est + runif(1, min=-f, max=f)
positions_est <- positions_est + + runif(1, min=-0.1, max=0.1)

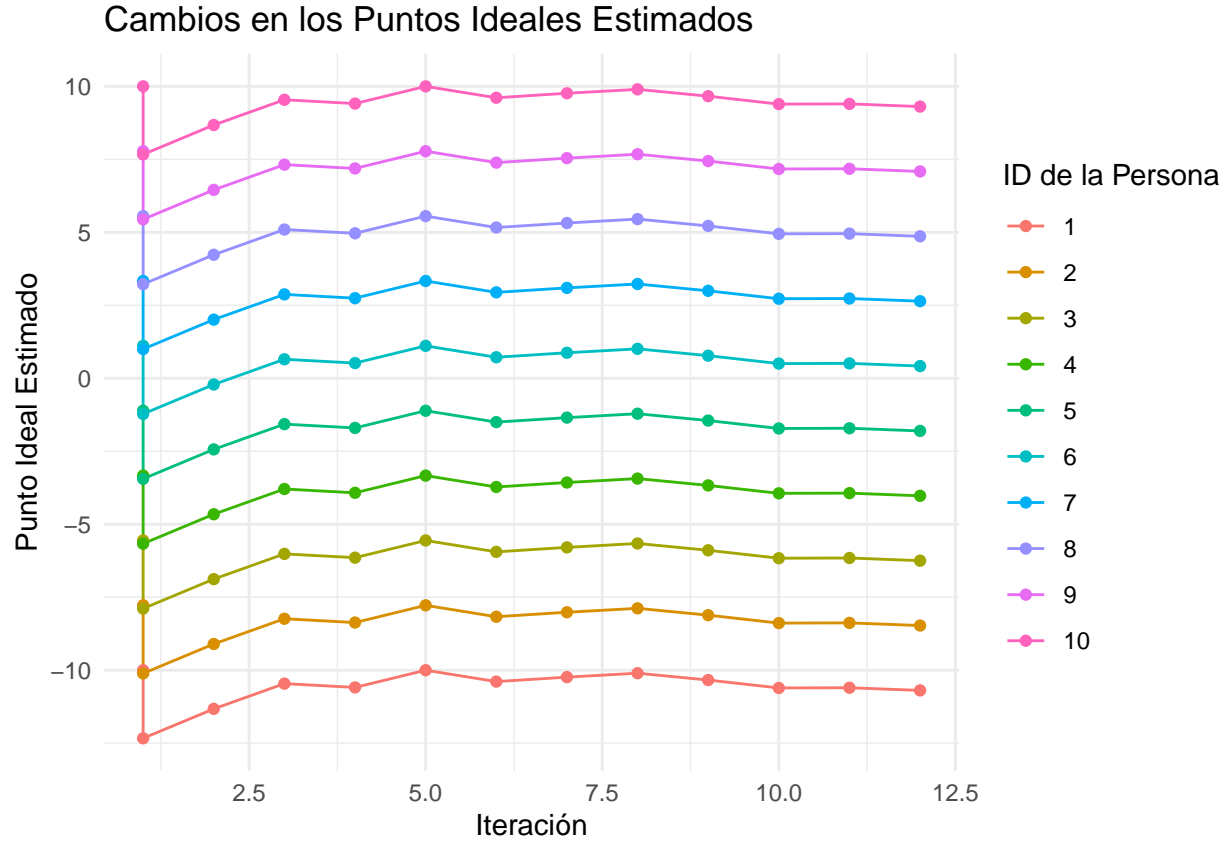
it <- 12 # número de iteraciones

for(j in 1:it){
# 1. Cálculo de Probabilidades considerando que ideals_est ha variado
votacion$Prob <- mapply(calc_prob,
                        ideals_est[votacion$Person_ID],
                        positions_est[votacion$Vote_ID])
ideals_df <- rbind(ideals_df,
                  data.frame(Iteration = j,
                             Person_ID = 1:n,
                             Ideal = ideals_est))

# 2. Simulación de la Maximización de Verosimilitud
ideals_est <- ideals_est + runif(1, min=-f, max=f)/j
positions_est <- positions_est + runif(1, min=-0.1, max=0.1)/j
}

# Gráfico resumen
ggplot(ideals_df, aes(x = Iteration, y = Ideal, group = Person_ID,
                    color = as.factor(Person_ID))) +
  geom_line() +
  geom_point() +
  labs(title = "Cambios en los Puntos Ideales Estimados",
       x = "Iteración",
       y = "Punto Ideal Estimado",
       color = "ID de la Persona") +
  theme_minimal()

```



Una versión más completa de NOMINATE es W-NOMINATE. La que puede resumirse así:

Sea  $i = 1, 2, \dots, n$  un cuerpo colegiado que decide sobre una materia  $k$  donde  $d = \{\text{Aprueba}, \text{Rechaza}\}$ :

$$U_{ikd} = u_{ikd} + \epsilon_{ikd}$$

Entonces:

$$U_{ik}(\text{Aprueba}) > U_{ik}(\text{Rechaza}) : u_{ika} - u_{ikr} > \epsilon_{ika} - \epsilon_{ikr}$$

Y lo mismo para cada tema que se vote. El problema empírico es que la distribución de los errores es desconocida. Por lo tanto, debe estimarse. Al respecto, WNOMINATE asume una distribución normal de los errores para cada  $i$  en cada materia  $1, 2, \dots, k \in s$  y cada votación  $1, 2, \dots, j \in q$  de forma tal que:

$$U_{ijy} = \beta e^{\left[ \frac{\sum_{k=1}^s w_k^2 d_{ijyk}^2}{2} \right]} + \epsilon_{ijy}$$

Donde  $d_{ijyk}^2 = \sum_{k=1}^s (x_{ij} - z_{jyk})^2$

Ideal del individuo  $i$  en la votación  $j$       Ubicación de votar A favor en tema  $k$       Saliencia del tema      Peso de la parte determinística

## WNOMINATE aplicado

```
# Cargar el paquete 'here' para manejar rutas de archivos de manera más fácil
require(here)
# Establecer la ruta actual como la ruta de trabajo
aqui <- here()

# Cargar librerías necesarias para la manipulación de datos y análisis
library(foreign)
library(gdata)

# Cargar librerías para análisis paramétrico y no paramétrico de datos
# en formatos rollcall
library(wnominate)

# Leer datos de un archivo CSV que contiene votaciones de las Naciones Unidas
UN <- read.csv(paste0(aqui, "/un.csv"), header=FALSE, strip.white=TRUE)

# Preparar la base de datos
# Extraer la columna de países y atributos
pais <- UN[,1]
atributos <- matrix(UN[,2], length(UN[,2]), 1)
colnames(atributos) <- "party"

# Eliminar las columnas de país y atributos para quedarse solo con los votos
UN <- UN[, -c(1,2)]

# Crear un objeto de clase rollcall para el análisis con wnominate
rc <- rollcall(UN,
               yea=c(1,2,3),
               nay=c(4,5,6),
               missing=c(7,8,9),
               notInLegis=0,
               legis.names=pais,
               legis.data=atributos,
               desc="UN 31 to 33")

# Ejecutar el análisis wnominate
result <- wnominate(rc, dims=2, polarity=c(1,1))
```

```
##
## Preparing to run W-NOMINATE...
##
## Checking data...
##
## All members meet minimum vote requirements.
##
## Votes dropped:
## ... 18 of 237 total votes dropped.
##
## Running W-NOMINATE...
##
## Getting bill parameters...
## Getting legislator coordinates...
```

```

## Starting estimation of Beta...
## Getting bill parameters...
## Getting legislator coordinates...
## Starting estimation of Beta...
## Getting bill parameters...
## Getting legislator coordinates...
## Getting bill parameters...
## Getting legislator coordinates...
## Estimating weights...
## Getting bill parameters...
## Getting legislator coordinates...
## Estimating weights...
## Getting bill parameters...
## Getting legislator coordinates...
##
##
## W-NOMINATE estimation completed successfully.
## W-NOMINATE took 65.59 seconds to execute.

# Mostrar un resumen de los resultados
summary(result)

##
##
## SUMMARY OF W-NOMINATE OBJECT
## -----
##
## Number of Legislators:      59 (0 legislators deleted)
## Number of Votes:           219 (18 votes deleted)
## Number of Dimensions:       2
## Predicted Yeas:             4702 of 5039 (93.3%) predictions correct
## Predicted Nays:             4121 of 4488 (91.8%) predictions correct
## Correct Classification:      89.49% 92.61%
## APRE:                       0.573 0.7
## GMP:                        0.783 0.841
##
##
## The first 10 legislator estimates are:

##           coord1D coord2D
## United States  0.931  0.365
## Canada         0.931  0.366
## Cuba           0.519 -0.391
## Haiti          0.362 -0.129
## Dominican Rep  0.796 -0.228
## Mexico         0.459  0.026
## Guatemala      0.382  0.365
## Honduras       0.587 -0.269
## El Salvador    0.888 -0.460
## Nicaragua      0.876 -0.301

# Explorar los componentes del objeto 'result'
names(result)

## [1] "legislators" "rollcalls" "dimensions" "eigenvalues" "beta"
## [6] "weights" "fits"

```

```
head(result$legislators)
```

```
##           party correctYea wrongYea wrongNay correctNay      GMP      CC
## United States Other        83        5        6       106 0.8625017 0.9450000
## Canada        Other        79        4        4       97 0.8697127 0.9565217
## Cuba          Other        99        8        5       59 0.8575070 0.9239766
## Haiti         Other        80        4        7       53 0.8137493 0.9236111
## Dominican Rep Other        78        9        7       76 0.8304488 0.9058824
## Mexico        Other        91        6        7       53 0.8201158 0.9171975
##           coord1D      coord2D selD se2D corr.1
## United States 0.9305151 0.36471203 0 0 0
## Canada        0.9308084 0.36550757 0 0 0
## Cuba          0.5190305 -0.39070553 0 0 0
## Haiti         0.3616978 -0.12925637 0 0 0
## Dominican Rep 0.7958211 -0.22819380 0 0 0
## Mexico        0.4587053 0.02637642 0 0 0
```

```
head(result$rollcalls)
```

```
##      correctYea wrongYea wrongNay correctNay      GMP      PRE      spread1D
## 1           23         6         6       12 0.6268082 0.3333333 -0.05225485
## 2            7         0         2       34 0.8323205 0.7777778 0.22332399
## 3           21        10         7       12 0.5542633 0.2272727 0.12107838
## 4            8         6         4       21 0.6194052 0.1666667 -0.10000002
## 5           20         3         5       15 0.7068619 0.5555556 -0.08745581
## 6           38         1         0        0 0.8431301 0.0000000 1.00000000
##      spread2D midpoint1D midpoint2D
## 1 2.8446059 -0.93767309 0.156832755
## 2 -0.9884471 -0.36141780 0.133013800
## 3 0.2350686 0.75770509 -0.076091960
## 4 0.4361386 0.61595130 -0.098933913
## 5 0.8235936 -0.05485914 0.005830734
## 6 0.0000000 1.00000000 0.000000000
```

```
result$dimensions
```

```
## [1] 2
```

```
head(result$eigenvalues)
```

```
## [1] 3.5807960 1.0402914 0.5464752 0.2699804 0.2011858 0.1322650
```

```
result$beta
```

```
## [1] 17.12951
## attr(,"Csingle")
## [1] TRUE
```

```
result$weights
```

```
## [1] 0.6420091 0.5386842
## attr(,"Csingle")
## [1] TRUE
```

```
result$fits
```

```
## correctclass1D correctclass2D      apre1D      apre2D      gmp1D
##      89.4930191      92.6104813    0.5734981    0.7000426    0.7829756
```



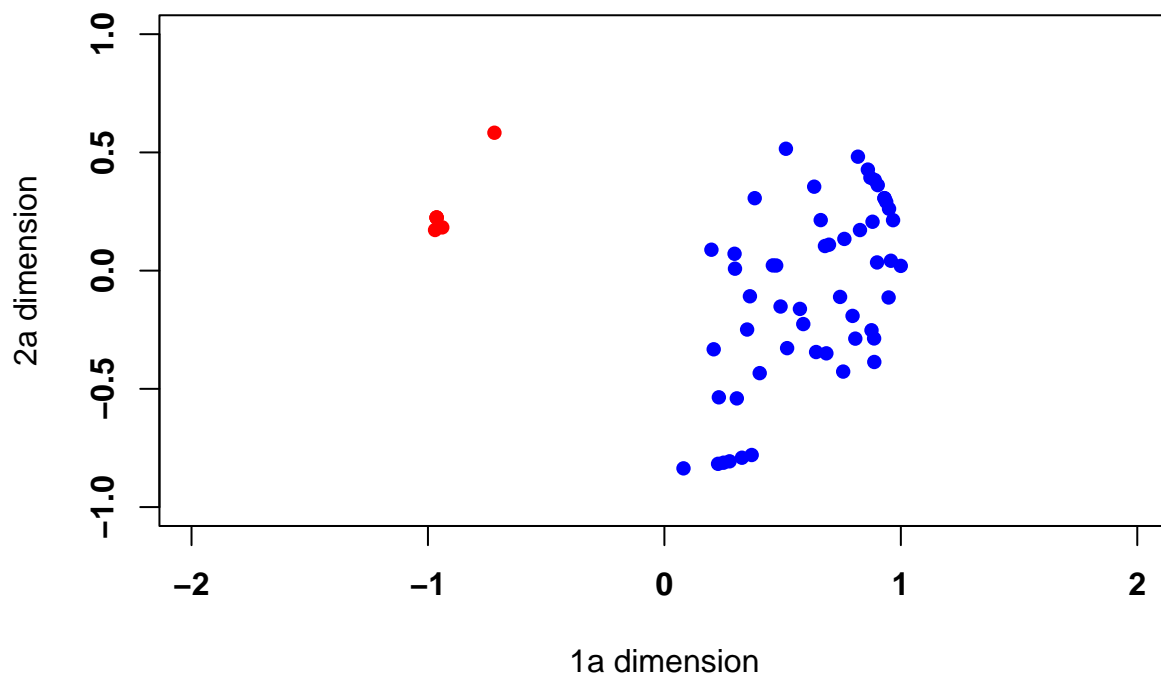
```
##          gmp2D
##      0.8405045

# Calcular el peso relativo de la segunda dimensión
WEIGHT = (result$weights[2]) / (result$weights[1])

# Extraer las coordenadas de las dimensiones
X1 <- result$legislators$coord1D
X2 <- (result$legislators$coord2D) * WEIGHT # (su peso es menor)

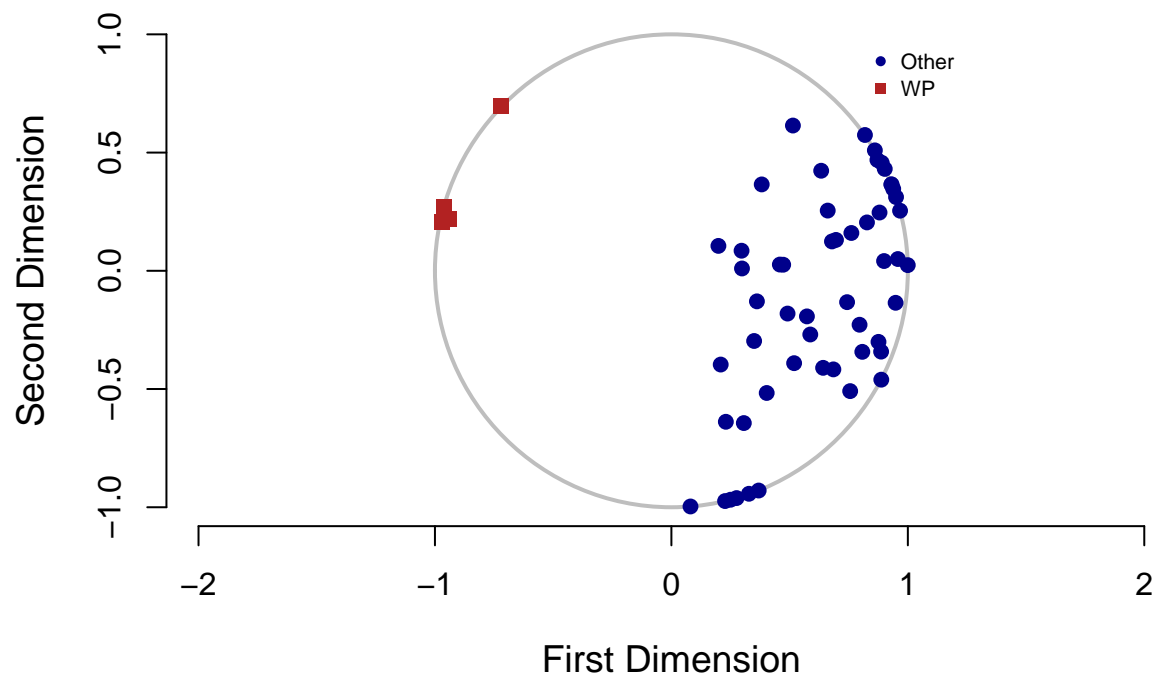
# Crear un gráfico de dispersión para visualizar las dimensiones
plot(X1, X2, type="n", asp=1,
      xlab="1a dimension",
      ylab="2a dimension",
      xlim=c(-1.0, 1.0), ylim=c(-1.0, 1.0), font=2, cex=1.2)
mtext("Naciones Unidas: 31 - 33", side=3, line=1.50, cex=1.2, font=2)
points(X1[result$legislators$party == "Other"],
       X2[result$legislators$party == "Other"], pch=16, col="blue", font=2)
points(X1[result$legislators$party == "WP"],
       X2[result$legislators$party == "WP"], pch=16, col="red", font=2)
```

### Naciones Unidas: 31 – 33



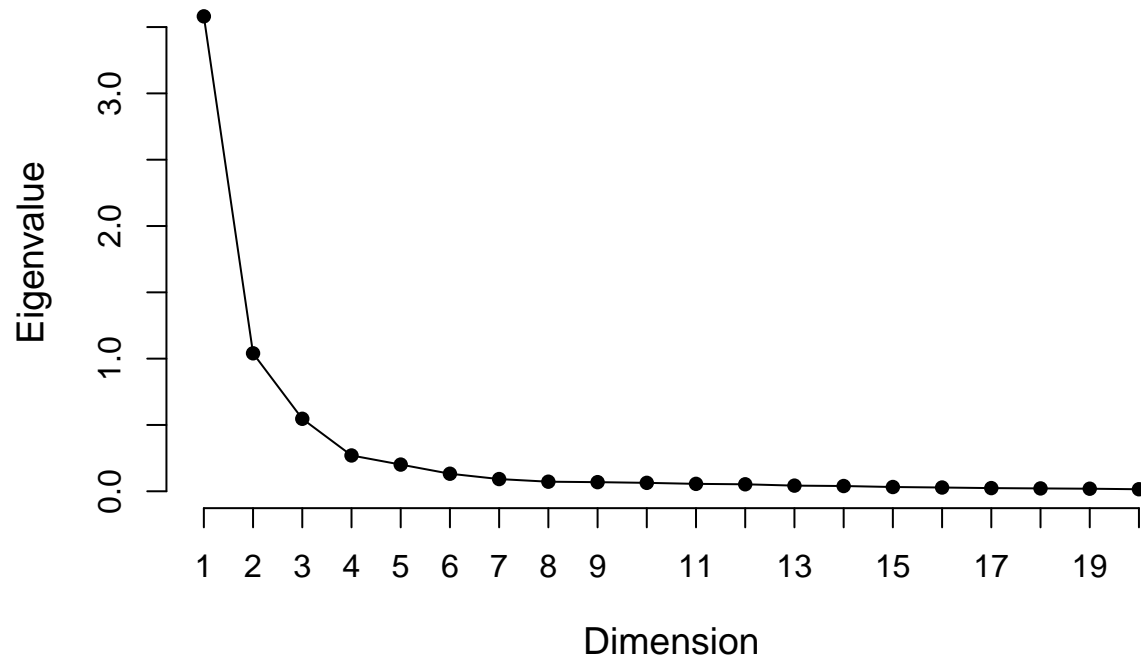
```
# Gráficos para visualizar los resultados de unominate
plot.coords(result) # Gráfico de coordenadas separado por grupos
```

## W-NOMINATE Coordinates

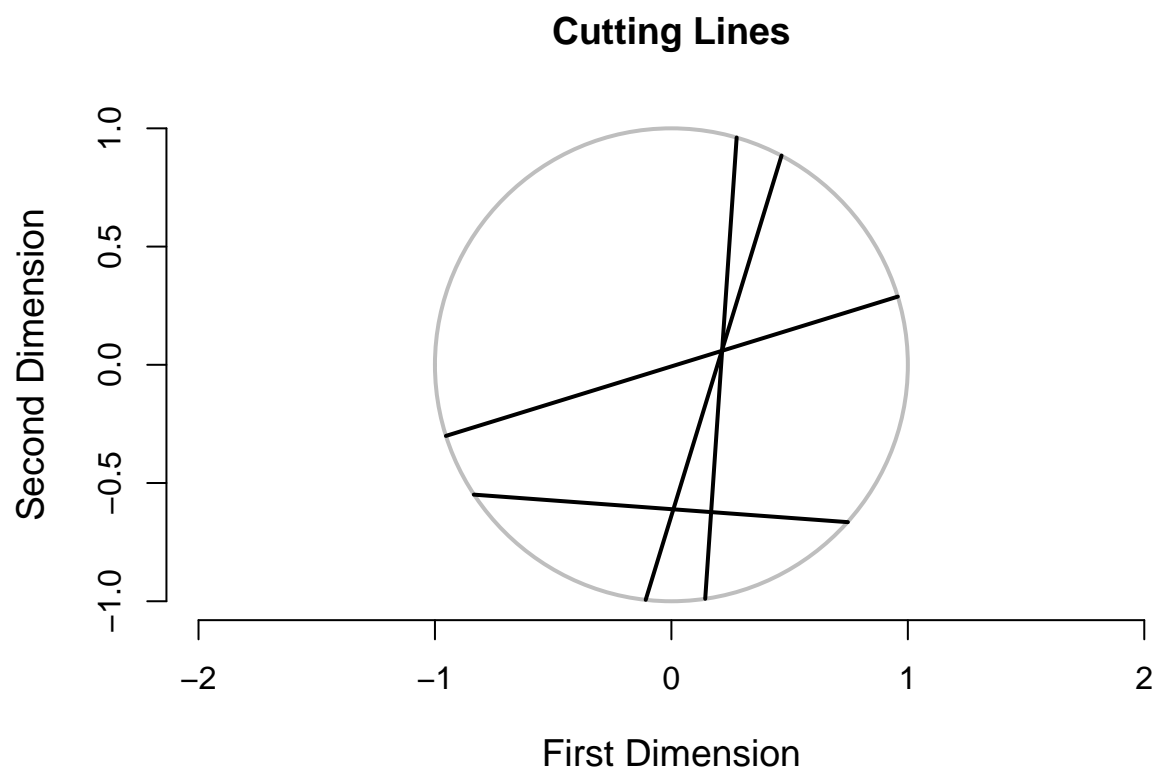


```
plot.screes(result) # Gráfico para mostrar la importancia de cada eigenvalue
```

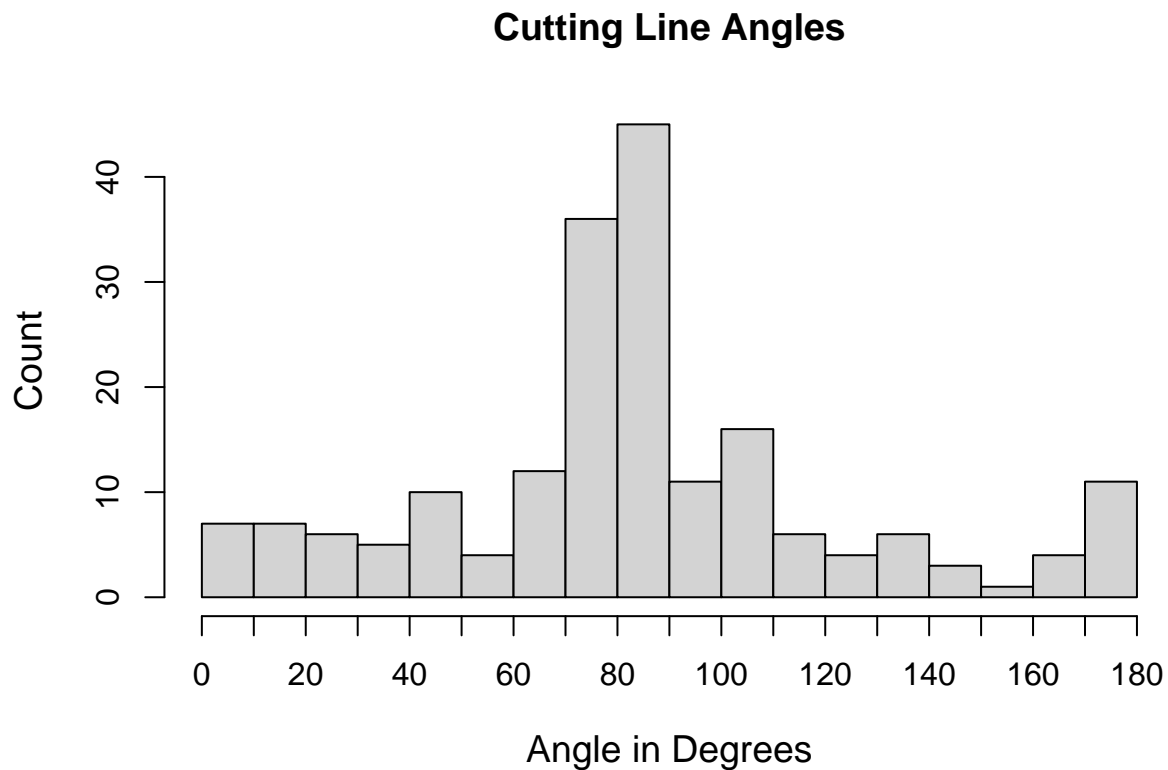
## Scree Plot



```
plot.cutlines(result, lines=5) # Gráfico de líneas de corte para cada proyecto
```



```
## NULL  
plot.angles(result) # Histograma de ángulos de corte
```



## Ahora, aplicamos el mismo análisis para datos del Senado de Chile

Aquí tomaremos algunos datos de votaciones en el período 2014-2016 del Senado de Chile.

```
# Leer los datos del archivo CSV
senado <- read.csv(paste0(aqui,"/votos2014_2016_procesado.csv"), sep=";")
# Extraer los nombres de los legisladores
nombres <- senado[,1]
# Eliminar la columna de nombres para quedarse solo con los votos
senado <- senado[,2:NCOL(senado)]

# Crear un objeto de clase rollcall para el análisis con wnominate
rc_senado <- rollcall(senado,
  yea=c(1),
  nay=c(-1),
  missing=c(0),
  notInLegis=NULL,
  legis.names=nombres,
  legis.data=NULL,
  desc="Senado 2014-15")

# Ejecutar el análisis wnominate
result_senado <- wnominate(rc_senado, dims=2, polarity=c(35,35))

##
## Preparing to run W-NOMINATE...
```

```

##
## Checking data...
##
## All members meet minimum vote requirements.
##
## Votes dropped:
## ... 526 of 935 total votes dropped.
##
## Running W-NOMINATE...
##
## Getting bill parameters...
## Getting legislator coordinates...
## Starting estimation of Beta...
## Getting bill parameters...
## Getting legislator coordinates...
## Starting estimation of Beta...
## Getting bill parameters...
## Getting legislator coordinates...
## Getting bill parameters...
## Getting legislator coordinates...
## Estimating weights...
## Getting bill parameters...
## Getting legislator coordinates...
## Estimating weights...
## Getting bill parameters...
## Getting legislator coordinates...
##
## W-NOMINATE estimation completed successfully.
## W-NOMINATE took 7.58 seconds to execute.
# Mostrar un resumen de los resultados
summary(result_senado)

##
##
## SUMMARY OF W-NOMINATE OBJECT
## -----
##
## Number of Legislators:      38 (0 legislators deleted)
## Number of Votes:      409 (526 votes deleted)
## Number of Dimensions:      2
## Predicted Yeas:      6493 of 6751 (96.2%) predictions correct
## Predicted Nays:      4134 of 4410 (93.7%) predictions correct
## Correct Classification:      94.41% 95.22%
## APRE:      0.803 0.832
## GMP:      0.858 0.893
##
##
## The first 10 legislator estimates are:
##
##      coord1D coord2D
## Allamand Z., Andr\xe9s      0.649 -0.196
## Allende B., Isabel      -0.919  0.303
## Araya G., Pedro      -0.841 -0.335

```

## Bianchi C., Carlos	-0.536	-0.844
## Chahu\xe1n C., Francisco	0.651	-0.549
## Coloma C., Juan Antonio	0.984	0.179
## De Urresti L., Alfonso	-0.979	0.043
## Espina O., Alberto	0.607	-0.009
## Garc\xeda Huidobro S., Alejandro	0.844	0.247
## Garc\xeda R., Jos\xe	0.594	0.196

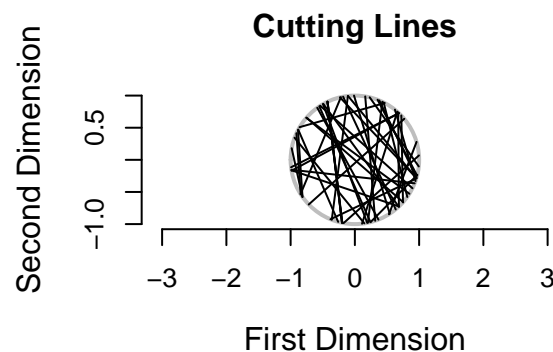
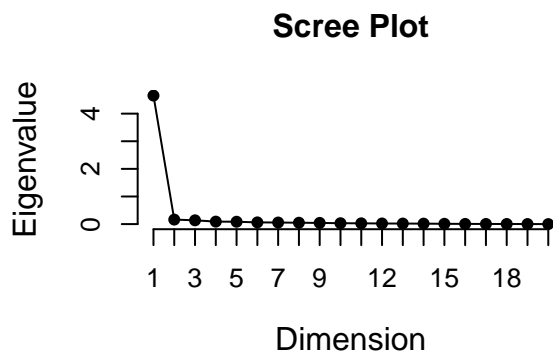
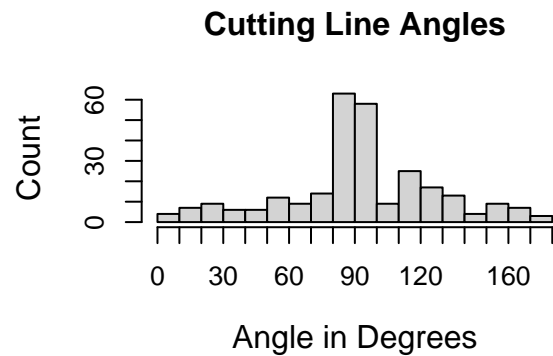
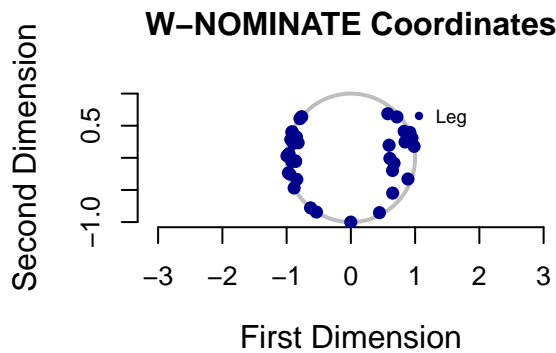
En el resultado se incluyen dos medidas de ajuste del modelo.

+ APRE (Aggregate Proportional Reduction in Error) mide la reducción proporcional en el error de clasificación en comparación con un modelo nulo que siempre predice la clase más frecuente en el conjunto de datos. APRE se calcula como:

$$APRE = 1 - \frac{\text{Error del modelo}}{\text{Errores del modelo nulo}}$$

+ GMP (Geometric Mean Probability) es la media geométrica de las probabilidades predichas ( $p_i$ ) para cada observaciones  $i$  clasificada correctamente.

$$GMP = (\prod_{i=1}^n p_i)^{1/n}$$

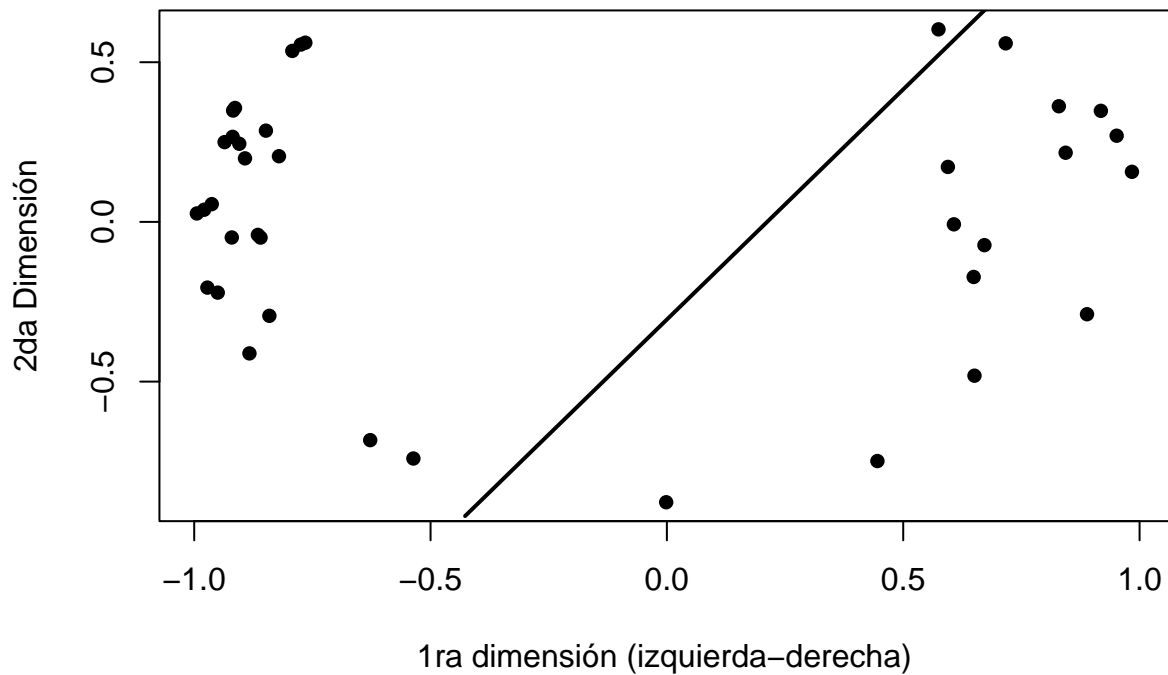


## NULL

En este caso particular una sola dimensión es significativa. Pero supongamos que tenemos dos dimensiones significativas. Cuando eso sucede, es necesario considerar el peso relativo de cada dimensión.

Con estas modificaciones podemos ahora revisar cómo se comporta el modelo para un proyecto/votación específica:

## Senado – WNOMINATE – Proyecto de ley



Como modelo frecuentista, los valores obtenidos son parámetros y, por ende, no tienen variación. Para que haya variación dentro de una aproximación frecuentista se hace bootstrapping.

```
# Bootstrap para estimar la incertidumbre en WNOMINATE
library(ellipse)
result_senado_boot <- wnominate(rc_senado,
                                ubeta=15, # Valor por defecto para el parámetro beta
                                uweights=0.5, # Valor por defecto para los pesos
                                dims=2, # Número de dimensiones
                                minvotes=10, # Núm mínimo para incluir en el análisis
                                lop=0.025, # Parámetro para el método de optimización
                                trials=10, # Número de iteraciones de bootstrap
                                # esto debería ser 10000 o más, pero es lento
                                polarity=c(1,5), # Polaridad para las dimensiones
                                verbose=FALSE) # No mostrar mensajes durante la ejec.
```

```
##
## Preparing to run W-NOMINATE...
##
## Checking data...
##
## All members meet minimum vote requirements.
##
## Votes dropped:
## ... 526 of 935 total votes dropped.
##
## Running W-NOMINATE...
```



```

##
##      Getting bill parameters...
##      Getting legislator coordinates...
##      Starting estimation of Beta...
##      Getting bill parameters...
##      Getting legislator coordinates...
##      Starting estimation of Beta...
##      Getting bill parameters...
##      Getting legislator coordinates...
##      Getting bill parameters...
##      Getting legislator coordinates...
##      Estimating weights...
##      Getting bill parameters...
##      Getting legislator coordinates...
##      Estimating weights...
##      Getting bill parameters...
##      Getting legislator coordinates...
##      Starting bootstrap iterations...
## .....      Computing standard errors...
## W-NOMINATE estimation completed successfully.
## W-NOMINATE took 730.4 seconds to execute.

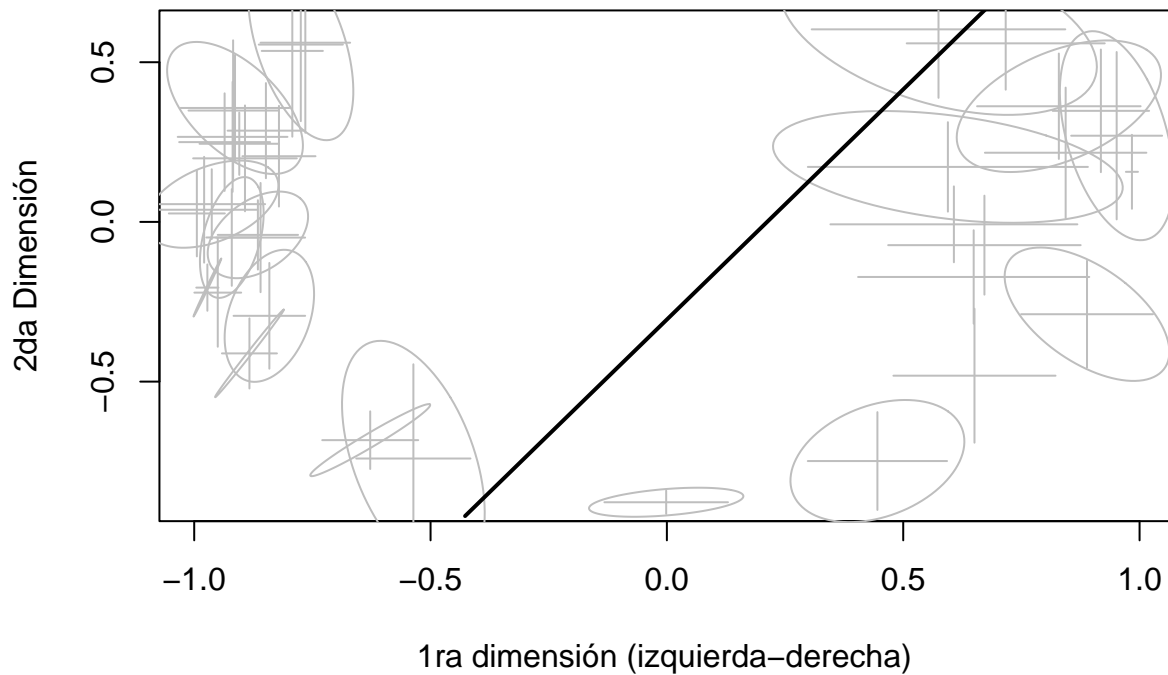
# # Extraer las desviaciones estándar de las dimensiones
std1 <- result_senado_boot$legislators$se1D
std2 <- result_senado_boot$legislators$se2D * WEIGHT
corr12 <- result_senado_boot$legislators$corr.1

# # Gráfico de las dimensiones con intervalos de confianza
plot(X1, X2,
     xlab="1ra dimensión (izquierda-derecha)",
     ylab="2da Dimensión",
     pch=16,
     main="Ejemplo - WNOMINATE",
     type="n")

# Dibujar elipses para representar los intervalos de confianza
for (i in 1:nrow(result_senado_boot$legislators)){
  if(!is.na(corr12[i])){
    lines(c(X1[i],X1[i]),
          c(X2[i]- 1.96*std2[i], X2[i] + 1.96*std2[i]),
          col="gray")
    lines(c(X1[i] - 1.96*std1[i], X1[i] + 1.96*std1[i]),
          c(X2[i], X2[i]),
          col="gray")
    if (abs(corr12[i]) > .30){
      lines(ellipse(x=corr12[i],
                    scale=c(std1[i],std2[i]),
                    centre=c(X1[i] ,X2[i])),
              col="gray")
    }
  }
}
segments(xws+N2W, yws-N1W, xws-N2W, yws+N1W, lwd=2, col="black")

```

## Ejemplo – WNOMINATE



```
polarity <- X1*N1W + X2*N2W - ws
```

---

Estimación de Ideología Política usando MCMC (Monte Carlo Markov Chain)

---

Problemas y limitaciones de usar NOMINATE:

1. Teóricos: No todos los roll calls son adecuados para todos los comportamientos políticos (p.ej., votos sinceros vs. estratégicos).
2. Prácticos: Mejorar la precisión de NOMINATE es computacionalmente costoso.
3. Estadísticos: NOMINATE estima múltiples parámetros simultáneamente, lo que debilita la base estadística del modelo.

## Enfoque Bayesiano propuesto por Jackman y otros:

- Los datos son conocidos y fijos.
- Los parámetros son variables y se estiman usando una distribución a priori.
- Este enfoque permite estimaciones más flexibles y robustas.

Nota sobre la complejidad computacional: - Aumentar el número de legisladores o dimensiones incrementa exponencialmente el número de parámetros a estimar, lo que puede llevar a estimaciones inconsistentes.

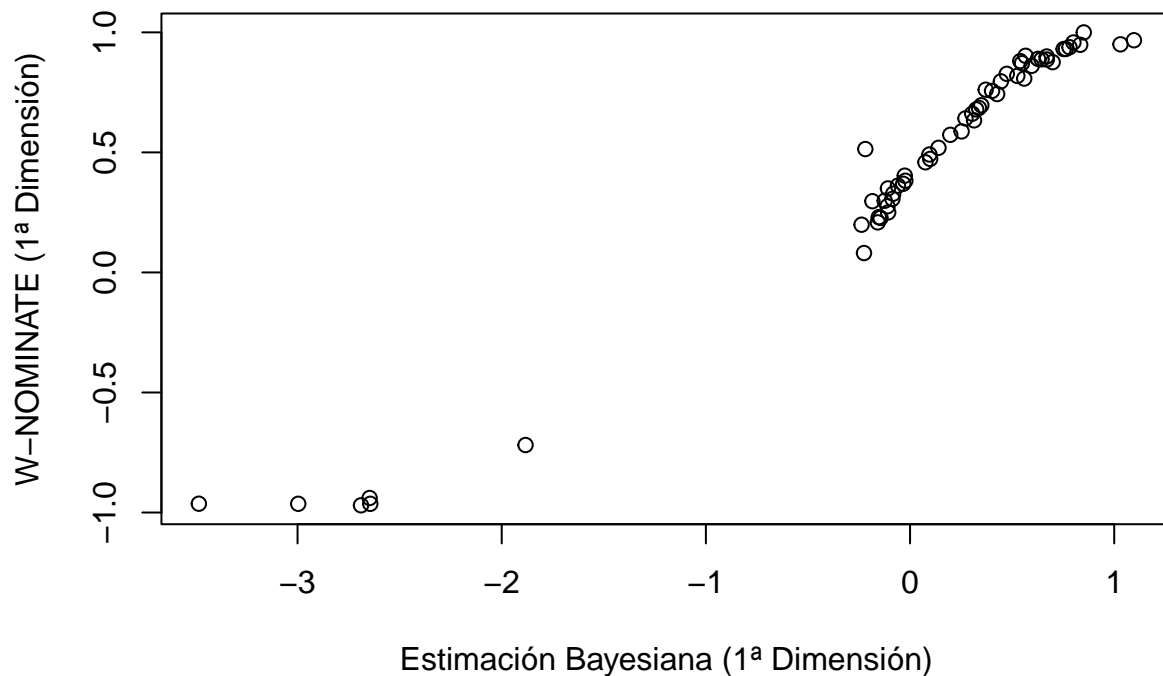
```
# Ejecutar el modelo MCMC
mcmc_e <- ideal(rc, codes = rc$codes,
  maxiter = 8000, # Número máximo de iteraciones
  burnin = 1000,  # Número de iteraciones para el periodo de "quemado"
```

```

thin = 500,      # Intervalo para guardar muestras
normalize = T)   # Normalizar los datos (ver documentación para más detalles)

## ideal: analysis of roll call data via Markov chain Monte Carlo methods.
##
## Ideal Point Estimation
##
## Number of Legislators      59
## Number of Items           223
##
##
## Starting MCMC Iterations...
# Comparar resultados entre la estimación Bayesiana y W-NOMINATE
plot(-mcmc_e$xbar[,1], result$legislators$coord1D,
     xlab="Estimación Bayesiana (1ª Dimensión)",
     ylab="W-NOMINATE (1ª Dimensión)")

```



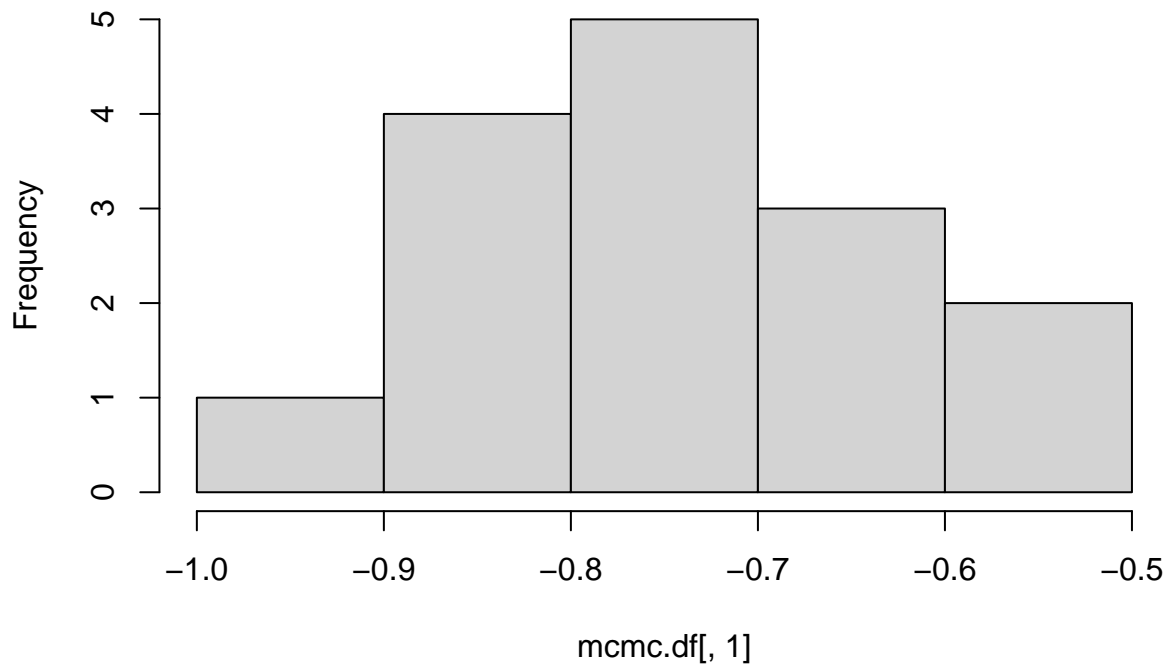
```

# Nota: Se multiplica por -1 para alinear las escalas. Esto es válido ya que
# estamos interesados en la relación, no en la magnitud.

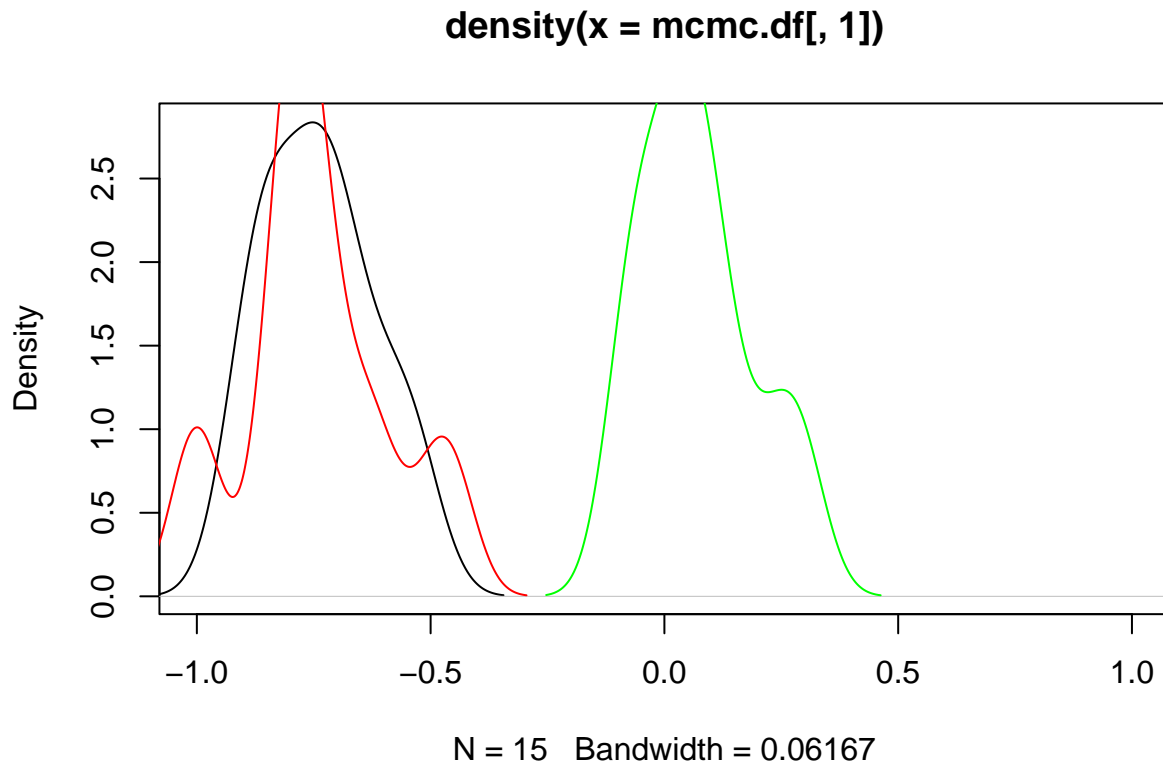
# Obtener y visualizar los márgenes de error
mcmc.df <- as.data.frame(mcmc_e$x)
hist(mcmc.df[,1]) # Histograma para la primera dimensión

```

**Histogram of mcmc.df[, 1]**



```
plot(density(mcmc.df[,1]), xlim=c(-1,1)) # Densidad para la 1era dimensión
lines(density(mcmc.df[,2]), col="red")   # Densidad para la 2da dim, en rojo
lines(density(mcmc.df[,4]), col="green") # Densidad para la 4ta dim, en verde
```



El método también se puede utilizar para variables ordinales con más de dos valores. Y por eso, sirve para hacer estimaciones a partir de escalas likert en encuestas.

```
# -----
# Estimación de Ideología Política a partir de Encuestas - tscores
# -----

# Importar la biblioteca necesaria
library(basicspace)

# Leer los datos desde un archivo CSV
base <- read.csv(paste0(aqui, "/ejemplo_basicspace.csv"))

# Tomar una muestra aleatoria de 1500 filas (necesario debido a
# limitaciones del algoritmo)
muestra <- sample(row.names(base), 1500)
base <- base[row.names(base) %in% muestra,]

# Preparar los datos para el análisis de tscores
# Convertir las columnas relevantes a numéricas
tscores <- cbind(as.numeric(base$t_allende),
                  as.numeric(base$t_pinochet),
                  as.numeric(base$t_aylwin),
                  as.numeric(base$t_freie),
                  as.numeric(base$t_lagos),
                  as.numeric(base$t_bachelet1),
                  as.numeric(base$t_pinera),
```

```

as.numeric(base$t_bachelet2))

# Asignar nombres de columnas y convertir a matriz
colnames(tscores) <- c("Allende", "Pinochet", "Aylwin", "Frei", "Lagos",
                       "Bachelet1", "Pinera", "Bachelet2")
tscores <- as.matrix(tscores)

# Ejecutar el modelo Blackbox para estimar la ideología
results_bb <- blackbox(tscores,
                      missing=c(8,9),
                      verbose=T,
                      dim=2,
                      minscale=3)

##
##
## Beginning Blackbox Scaling...8 stimuli have been provided.
##
## Blackbox estimation completed successfully.

# Visualizar las estimaciones de ideología para los individuos
d1 <- unlist(results_bb$individuals[1])
plot(density(d1, na.rm = T), main="Distribución Ideológica de los Encuestados")

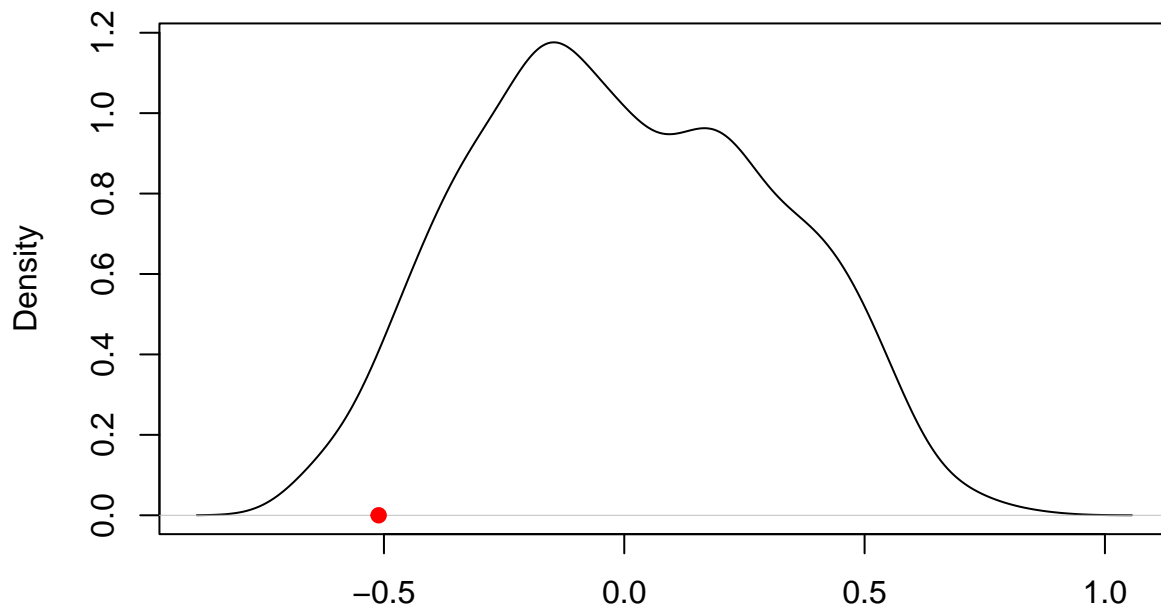
# Incorporar los resultados en el conjunto de datos original
psi.hat <- -results_bb$individuals[[1]]$c1
psi.hat.df <- as.data.frame(psi.hat)
colnames(psi.hat.df) <- c("ideologia")
base$ideologia <- unlist(psi.hat.df)

# Estimar la posición ideológica de los tscores
results_bb.t <- blackbox_transpose(tscores, dims=1, minscale = 3)
ideologia_tscores <- -results_bb.t$stimuli[[1]][2]

# Visualizar las estimaciones de ideología para los tscores
# Los puntos representan diferentes tscores y los colores pueden ser ajustados
# para representar diferentes categorías
plot(density(d1, na.rm = T), main="Distribución Ideológica de los Encuestados")
points(ideologia_tscores[1,1], 0, col="red", type='p', pch=19)

```

## Distribución Ideológica de los Encuestados



N = 1028 Bandwidth = 0.06941

```
# ... (resto de los puntos y etiquetas)
```

Hay otros métodos que son combinaciones de métodos frecuentistas y bayesianos que tienen la ventaja de ser más rápidos y aptos para bases de datos con más observaciones o t-scores. Un caso es el de expected maximization combinado con teoría de respuesta al ítem (emIRT). Veamos un ejemplo de estimación dinámica.

```
# -----  
# Estimación de Ideología Dinámica  
# -----  
  
# Importar la biblioteca necesaria  
require(emIRT)  
  
# Leer los datos desde un archivo CSV  
archivo <- paste0(aqui, "/20170127data_integrado_revisado_solovotosefectivos.csv")  
data <- read.csv(archivo)  
  
# Preparar los datos para el análisis  
votes <- data[, 2:NCOL(data)]  
votes <- as.matrix(votes)  
J <- NCOL(votes)  
  
#####  
# para estimacion dinamica necesitamos cuatro set de datos  
# data: votos,  
#       periodo de inicio (starts),  
#       periodo de termino (ends),
```

```

#      numero de fallo (bill.session)
#      numero de fallos/periodo (asumiremos cada fallo es un periodo)

startlegis <- matrix(0,nrow=NROW(votes), ncol=1)
bill.session <- seq(1:NCOL(votes))
bill.session <- bill.session - 1
bill.session <- as.matrix(bill.session)

endlegis <- matrix(NCOL(votes)-1,nrow=NROW(votes), ncol=1)

panel.data <- list(rc = votes,
                  startlegis = startlegis,
                  endlegis = endlegis,
                  bill.session = bill.session,
                  T = NCOL(votes)) # cambiar por s

# starts:
# alpha: parametro de dificultad de orden J*1
# beta: parametro de discriminacion de orden J*1
# x: matriz de N*T

alpha <- matrix(0.1,nrow=NCOL(votes), ncol=1) # 0.1 arbitrario
beta <- matrix(-0.1,nrow=NCOL(votes), ncol=1) # -0.1 arbitrario
#x <- matrix(0,nrow=NROW(votes), ncol=s)
x <- matrix(0,nrow=NROW(votes), ncol=NCOL(votes))
starts.points <- list(alpha = alpha, beta = beta, x = x)

# priors:
# x.mu0: promedio inicial para ideal points de cada legislador de orden N*1
# x.sigma0: varianza inicial para cada estimacion
# beta.mu: promedio inicial para todos los bills, de orden 2*1
# beta.sigma: promedio inicial de varianza para todos los bills, de orden 2*2
# omega2: evolucion de la varianza por legislador

#x.mu0 <- result.w$legislators$coord1D # estimacion unominate como punto de partida
x.mu0 <- matrix(0,nrow=NROW(votes),ncol=1) # siguiendo ejemplo en paquete emIRT
x.sigma0 <- matrix(1, nrow=NROW(votes),ncol=1) # arbitrariamente prior en 1
beta.mu <- matrix(NA,nrow=2,ncol=1)
beta.mu[1,1] <- 0
beta.mu[2,1] <- 0
beta.sigma <- matrix(NA,nrow=2,ncol=2)
beta.sigma[1,1] <- 1
beta.sigma[2,1] <- 0
beta.sigma[1,2] <- 0
beta.sigma[2,2] <- 1
omega2 <- matrix(0.1,nrow=NROW(votes),1) # 0.1 siguiendo ejemplo en paquete

priors.points <- list(x.mu0 = x.mu0,
                    x.sigma0 = x.sigma0,
                    beta.mu = beta.mu,
                    beta.sigma = beta.sigma,
                    omega2 = omega2)

```



```

base <- list(base.data = panel.data,
             base.cur = starts.points,
             base.priors = priors.points)

result <- dynIRT(.data = base$base.data,
                 .starts = base$base.cur,
                 .priors = base$base.priors)

##
## =====
## dynIRT: Dynamic IRT via Variational Inference
##
## Done in 150 iterations, using 1 threads.
## =====

ideology <- result$means$x
row.names(ideology) <- data[,1]

# Veamos el resultado
ej1 <- 1
ej2 <- 36
ej3 <- 5
ej4 <- 25
ej5 <- 15

plot(ideology[ej1,],
     ylim=c(-1,18),
     col="blue",
     type="l")
text(100,8,row.names(ideology)[ej1], col="blue", cex = 0.75)
lines(ideology[ej2,],
     col="red")
text(100,7,row.names(ideology)[ej2], col="red", cex = 0.75)
lines(ideology[ej3,],
     col="green")
text(100,6,row.names(ideology)[ej3], col="green", cex = 0.75)
lines(ideology[ej4,],
     col="black")
text(100,5,row.names(ideology)[ej4], col="black", cex = 0.75)
lines(ideology[ej5,],
     col="grey")
text(100,4,row.names(ideology)[ej5], col="grey", cex = 0.75)

```

