Tarea 4 - Muldimensional Scaling

Equipo 3

Mariana Godina, Lorena Malpica, Sonia Mendizábal, Victor Montoya

---

En esta tarea se presenta cada paso para obtener Multidimensional Scaling usando datos de votaciones de propuestas en el senado. Para los valores faltantes de la base se creó la siguiente función que sustituye los valores faltantes por 99.

```
NAReplace <- function(col){
  col2 <- ifelse(is.na(col), 99, col)
  return(col2)
}
```

A continuación presentamos paso por paso el análisis:

1. Se calcula la matriz de distancias euclideanas.

```
d <- senado_votaciones %>%
  dplyr::select(-1:-3) %>%
  t() %>%
  as_tibble() %>%
  mutate_all(funs(NAReplace(.))) %>%
  dist(method = "euclidean") %>%
  as.matrix()
n <- nrow(d)
```

2. Se genera la matriz centradora `kn`.

```
kn <- diag(1, n) - (1/n)*rep(1, n)*rep(1, n)
dim(d)
```

```
## [1] 132 132
```

```
dim(kn)
```

```
## [1] 132 132
```

2. Se centra la matriz de distancias al cuadrado, por filas y por columnas. Así obtenemos la matriz `b`.

$$B = \frac{-1}{2} k_n D^2 k_n$$

```
b <- (-1/2) *((kn %*% d^2) %*% kn)
```

3. Se realiza la descomposición espectral

Sabemos que:

$$B = XX^T$$

Por lo tanto teniendo B podemos obtener X usando el método de descomposición espectral. Primero se obtienen los eigenvalores y eigenvectores de B tomando k eigenvectores y eigenvalores. P es la matriz de eigenvectores y C la matriz de eigenvalores.

$$P = [V_1, V_2, |...|, V_k] \qquad \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_k \qquad C = \begin{bmatrix} \lambda_1 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & \lambda_n \end{bmatrix}$$

```
eigenval <- eigen(b)
c <-  diag(eigenval$values)
p <-  eigenval$vectors
eigenval$values
```

```
##   [1]  7.226845e+06  2.730978e+06  1.900156e+06  1.566928e+06  1.291614e+06
##   [6]  1.244604e+06  1.078727e+06  9.165104e+05  8.539099e+05  8.268353e+05
##  [11]  7.774946e+05  7.249058e+05  6.590470e+05  6.337920e+05  5.622018e+05
##  [16]  5.532300e+05  5.273531e+05  4.783418e+05  4.539559e+05  4.241998e+05
##  [21]  3.960367e+05  3.852497e+05  3.618575e+05  3.485072e+05  3.363900e+05
##  [26]  3.214531e+05  3.077733e+05  2.764432e+05  2.667671e+05  2.659467e+05
##  [31]  2.513752e+05  2.435547e+05  2.277254e+05  2.218237e+05  2.143399e+05
##  [36]  1.991546e+05  1.898131e+05  1.840904e+05  1.741554e+05  1.684318e+05
##  [41]  1.600769e+05  1.534417e+05  1.488672e+05  1.435219e+05  1.337604e+05
##  [46]  1.303711e+05  1.241707e+05  1.160334e+05  1.134752e+05  1.099232e+05
##  [51]  1.038815e+05  1.023734e+05  9.973335e+04  9.706620e+04  9.101808e+04
##  [56]  8.751108e+04  8.325846e+04  7.847590e+04  7.683615e+04  7.261919e+04
##  [61]  7.011198e+04  6.746123e+04  6.506871e+04  6.214997e+04  5.933137e+04
##  [66]  5.663586e+04  5.544762e+04  4.913508e+04  4.788945e+04  4.412667e+04
##  [71]  4.240591e+04  4.161342e+04  4.024348e+04  3.826346e+04  3.675247e+04
##  [76]  3.449255e+04  3.304252e+04  3.074631e+04  2.890125e+04  2.797667e+04
##  [81]  2.644313e+04  2.555676e+04  2.343417e+04  2.243432e+04  2.046912e+04
##  [86]  1.964462e+04  1.809327e+04  1.654995e+04  1.490537e+04  1.402482e+04
##  [91]  1.308049e+04  1.234309e+04  1.152979e+04  1.057094e+04  9.737006e+03
##  [96]  9.156323e+03  8.759276e+03  7.251314e+03  6.651788e+03  6.529864e+03
## [101]  5.754558e+03  5.309442e+03  5.058593e+03  4.345134e+03  3.818365e+03
## [106]  3.494478e+03  2.677807e+03  1.954205e+03  1.688964e+03  1.485028e+03
## [111]  1.419210e+03  1.245689e+03  9.382085e+02  4.925007e+02  3.690811e+02
## [116]  3.215008e-10  2.826709e-10  1.174402e-10  1.106042e-10  9.218453e-11
## [121]  5.238703e-11  1.878759e-11  7.829437e-12  2.755681e-12 -1.635470e-11
## [126] -5.254907e-11 -5.623487e-11 -6.950537e-11 -8.463635e-11 -1.202264e-10
## [131] -1.732819e-10 -4.247444e-10
```

4. Aproximación de componentes.

De acuerdo a la demostración vista en clase sabemos que:

$$X = PC^{1/2} = [V_1, V_2, |...|, V_n] \begin{bmatrix} \sqrt{\lambda}_1 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & \sqrt{\lambda}_n \end{bmatrix} = [\sqrt{\lambda}_1 V_1, \sqrt{\lambda}_2 V_2, |...|, \sqrt{\lambda}_n V_n]$$

Suponemos que:

$$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$$

Entonces:

$$\hat{X} = [\sqrt{\lambda}_1 V_1, \sqrt{\lambda}_2 V_2, |...|, \sqrt{\lambda}_k V_k]$$

es una aproximación de grado k de X si

$$k \leq n$$

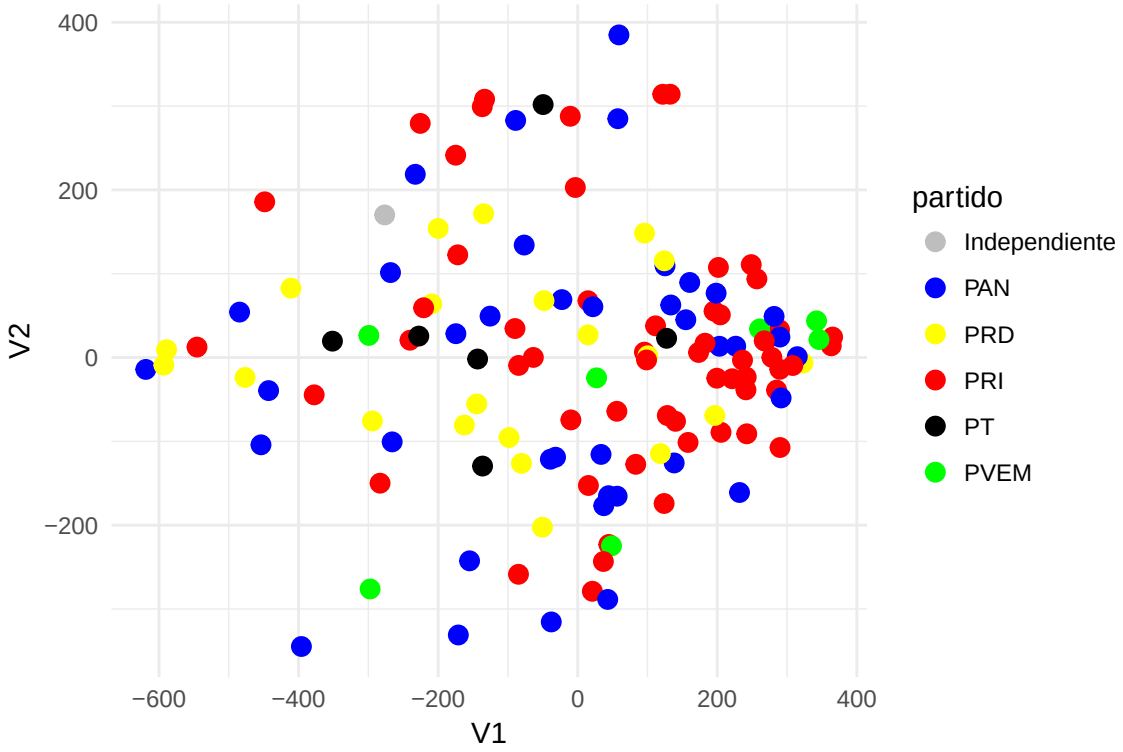
```
aprox <- p %*% sqrt( abs(c) ) %>%
  as_tibble()
aprox[1:5, 1:10]
```

```
## # A tibble: 5 × 10
##           V1         V2         V3         V4        V5           V6
##        <dbl>      <dbl>      <dbl>      <dbl>     <dbl>        <dbl>
## 1  268.75156   19.65564 -151.72688  147.37987 -84.27479  -2.90349520
## 2   44.62522 -223.01186  -12.79888 -257.22793 -37.15791   0.03383691
## 3 -174.82380  241.38841  114.61484  -52.95068  86.48357  46.66826282
## 4 -133.40766  308.19232 -131.84751   57.81849  18.22385 -28.30474364
## 5  365.32251   24.42520   18.15856  -46.22069 -56.86127  -9.02901687
## # ... with 4 more variables: V7 <dbl>, V8 <dbl>, V9 <dbl>, V10 <dbl>
```

5. Gráfica de componentes.

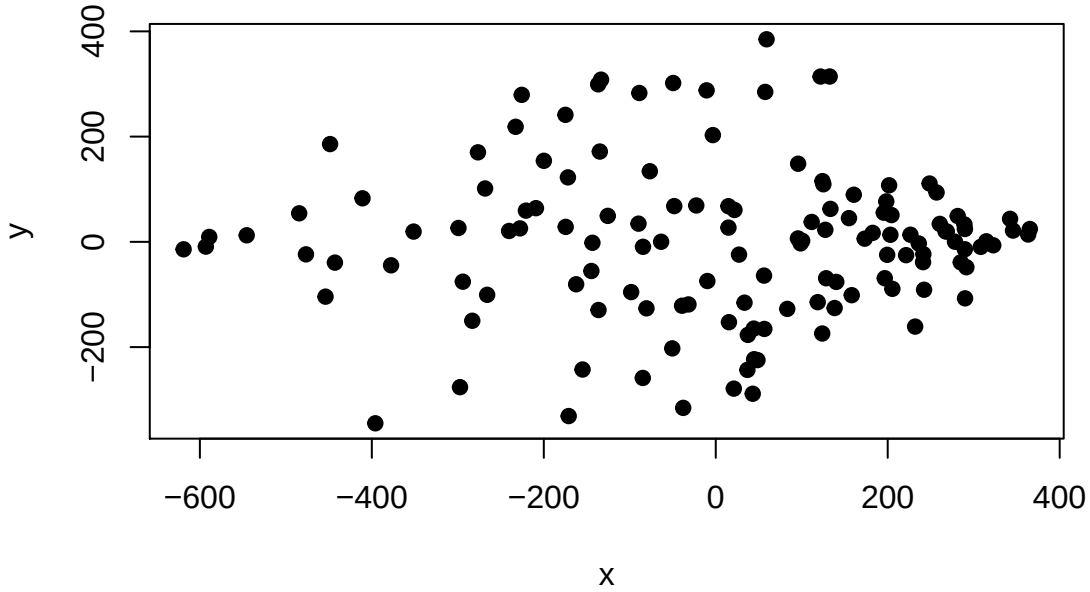
Graficamos dos componentes de la aproximación de X.

```
tab.gg <- aprox %>%
  as_tibble() %>%
  mutate(senador.id = row.names(.)) %>%
  left_join(tab.senadores, by = 'senador.id')

ggplot(tab.gg, aes(x = V1, y = V2, color = partido)) +
  geom_point(size = 3) +
  scale_color_manual(values = c("gray", "blue", "yellow", "red", "black", "green"))
```



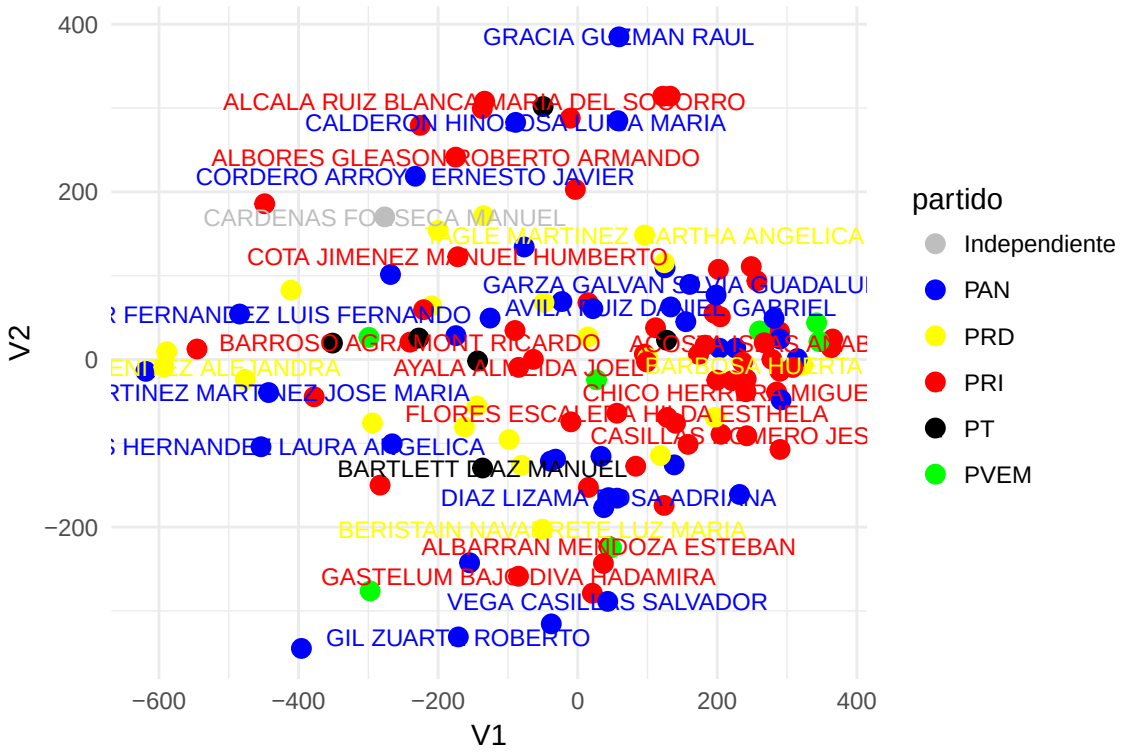
Usando el MDS con la función `cmdscale` de R y se obtienen resultados iguales.

```r
fit <- cmdscale(d, eig = TRUE, k = 2)
x <- fit$points[, 1]
y <- fit$points[, 2]
plot(x, y, pch = 19)
```



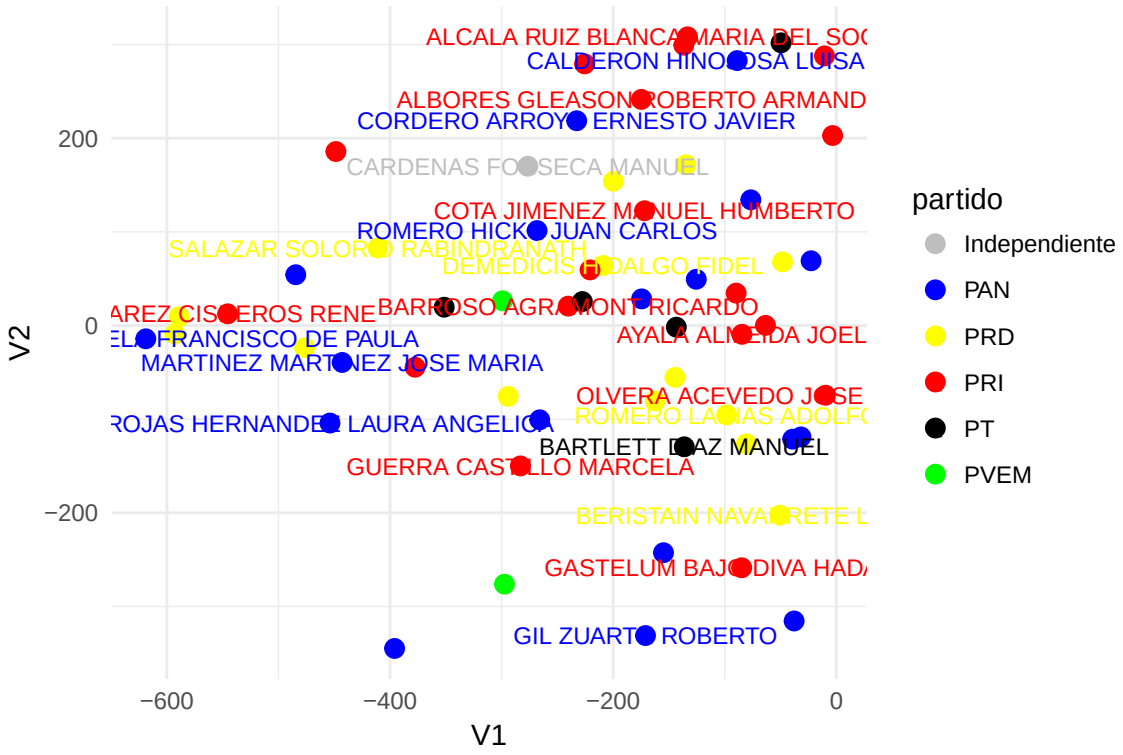
No se observa una agrupación clara entre los senadores de acuerdo al partido al que pertenecen. A continuación se presenta la gráfica incluyendo nombres de senadores.

```r
ggplot(tab.gg, aes(x = V1, y = V2,
                   color = partido,label = tab.gg$senador)) +
  geom_point(size = 3) +
  scale_color_manual(values = c("gray", "blue", "yellow", "red", "black", "green"))+
  geom_text(check_overlap = T, size = 3)
```



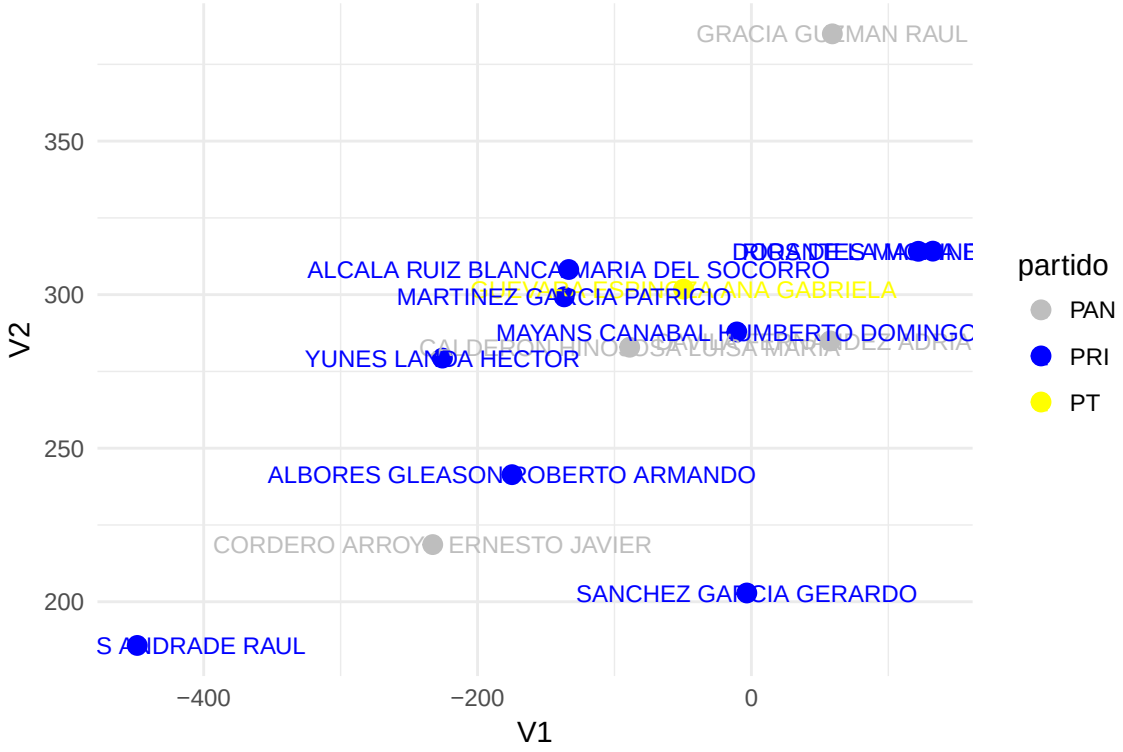
Sin embargo, si se puede ver un grupo compuesto principalmente por senadores del PAN y PRI. En la siguiente gráfica, se presentan algunos nombres de senadores agrupación mencionada.

```
ggplot(tab.gg[tab.gg$V1<0,], aes(x = V1, y = V2,
                                 color = partido,label = tab.gg[tab.gg$V1<0,]$senador)) +
  geom_point(size = 3) +
  scale_color_manual(values = c("gray", "blue", "yellow", "red", "black", "green"))+
  geom_text(check_overlap = T, size = 3)
```



Se observan algunos senadores aislados, en su mayoría senadores independientes, del PAN y del PRD.

```
ggplot(tab.gg[tab.gg$V2>175,],
       aes(x = V1, y = V2,
           color = partido,label = tab.gg[tab.gg$V2>175,]$senador)) +
  geom_point(size = 3) +
  scale_color_manual(values = c("gray", "blue", "yellow", "red", "black", "green"))+
  geom_text(check_overlap = F, size = 3)
```



Sería interesante buscar por nombre a los senadores que pertenecen al grupo más delimitado y ver que relación hay entre ellos, si son figuras líderes en sus partidos o si pertenecen a estados cercanos geográficamente.

6. Ejemplo Adicional

Como ejemplo adicional, se presenta la asociación de iniciativas propuestas en el senado. A continuación, se realiza el proceso del análisis multidimensional scaling.

```
d <- senado_votaciones %>%
  dplyr::select(-1:-3) %>%
  as_tibble() %>%
  mutate_all(funs(NAReplace(.))) %>%
  dist(method = "euclidean") %>%
  as.matrix()
n <- nrow(d)
kn <- diag(1, n) - (1/n)*rep(1, n)*rep(1, n)

dim(d)
```

```
## [1] 115 115
```

```
dim(kn)
```

```
## [1] 115 115
```

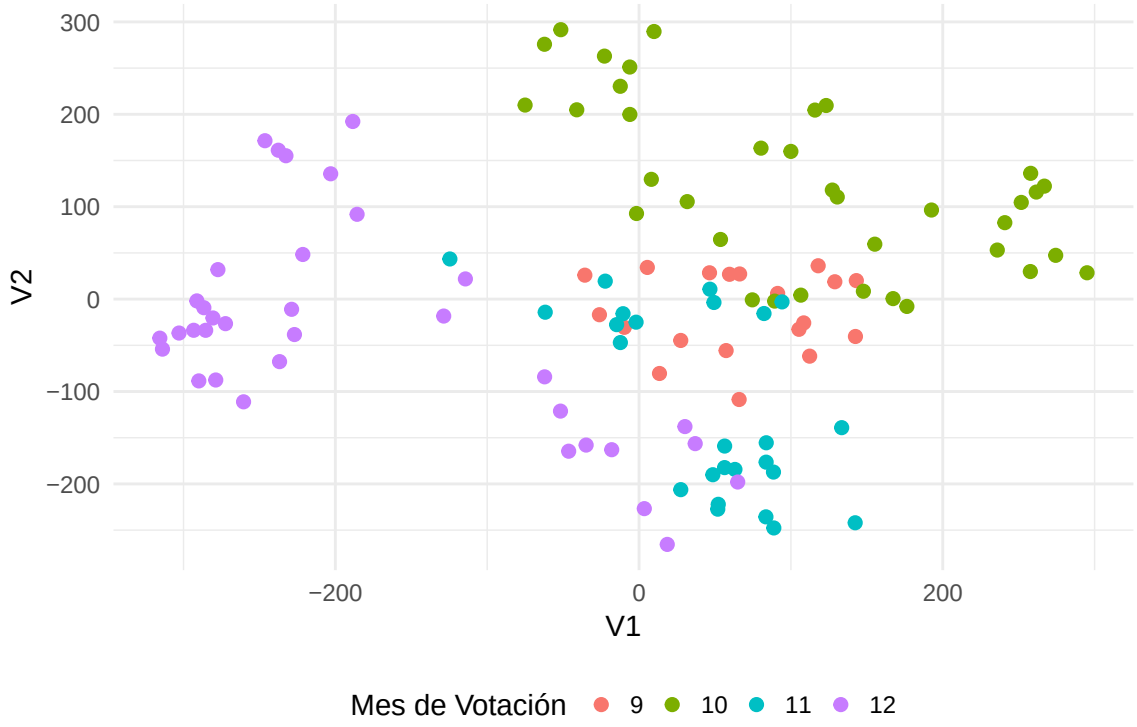
```
b <- (-1/2) *((kn %*% d^2) %*% kn)

eigenval <- eigen(b)
c <-  diag(eigenval$values)
p <-  eigenval$vectors
aprox <- p %*% sqrt( abs(c) ) %>%
  as_tibble()
```

```
aprox[1:5, 1:10]
```

```
## # A tibble: 5 × 10
##          V1        V2        V3       V4         V5           V6        V7
##       <dbl>     <dbl>     <dbl>    <dbl>      <dbl>        <dbl>     <dbl>
## 1 128.92899  18.78570 -216.1552 106.8306   60.75997   31.4269250 -32.88844
## 2 143.10604  20.03910 -207.6796 118.5734   37.33346    3.1686896  17.26219
## 3 117.96141  36.10625 -197.3675 107.4685   22.55415    0.8787043 -34.58044
## 4 105.30720 -32.68209 -202.7846 126.3941   16.49872   38.4524553 -20.54627
## 5  59.56375  26.86572 -126.8960 172.6954 -126.34844 -170.6654449 -83.97526
## # ... with 3 more variables: V8 <dbl>, V9 <dbl>, V10 <dbl>
```

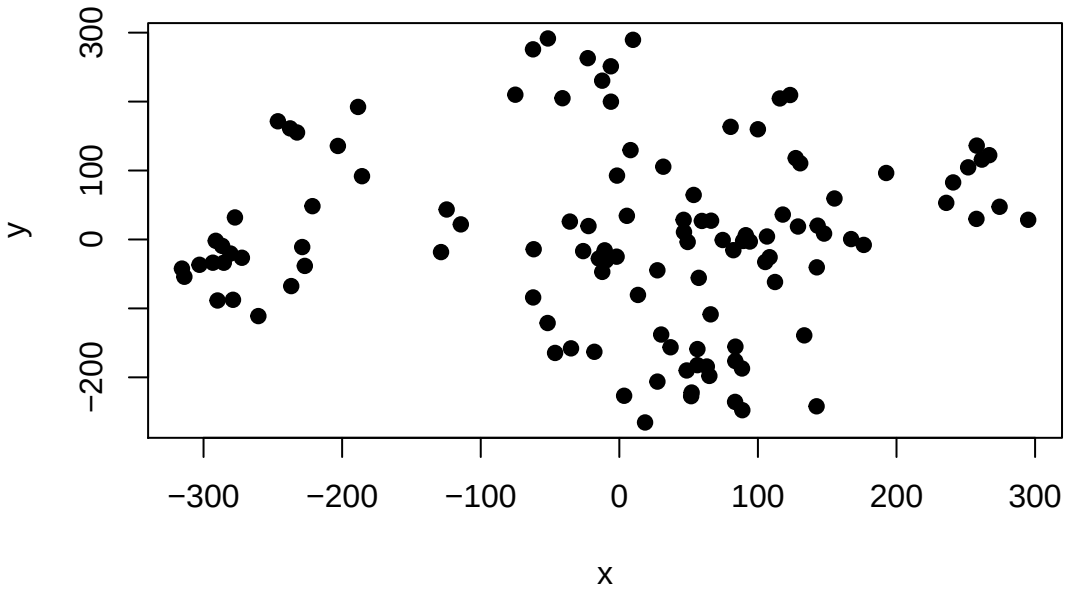
Con el resultado obtenido graficamos las distancias entre iniciativas y lo visualizamos de acuerdo a las fechas en las que fueron votadas.

```
fechas.vec <- factor(month(senado_votaciones$FECHA))
ggplot(aprox, aes(x = V1, y = V2)) +
  geom_point(aes(color = fechas.vec), size = 2)  +
  theme(legend.position = 'bottom') +
  guides(color = guide_legend(title = "Mes de Votación"))
```



Usando el MDS con la función `cmdscale` de R y se obtienen resultados iguales.

```
fit <- cmdscale(d, eig = TRUE, k = 2)
x <- fit$points[, 1]
y <- fit$points[, 2]
plot(x, y, pch = 19)
```



En este caso, se puede observar una agrupación de propuestas por mes.