# R Notebook

# Aplicación de k-Medias sobre datos de censo histórico

En este cuaderno de R se siguen los pasos necesarios para aplicar el arlgoritmo de IA no supervisado k-Medias sobre las entradas de un censo. Nuestros datos recogen en una sóla columna el nombre y apellido de cada entrada. También contiene el año de registro y la geolocalización de la localidad de cada persona. El proceso a seguir será: 1. La exportación de los datos. 2. Filtraremos por apellido y por año. 3. Ajustar el hiperparámetro k y aplicar k-Medias. 4. Representación en un mapa.

# 1. Exportación de datos.

La variable `dirección` contiene la ruta al archivo de datos del censo estructurado. Para exportar los datos a RStudio usaremos el comando `read.table` . Úsese en cada caso el separador de datos correcto; en nuestro caso es `sep = '\t'` . De todos los datos disponibles en la tabla, nos quedaremos solamente con el *nombre* de cada entrada, el *año*, el nombre de la *localidad*, su *latitud*, *longitud* y *altitud*.

```
# Cámbiese la `dirección` por la que corresponda al archivo 'GeolocLocalidades.tsv' en su propio ordenador. En Github, este archivo se encuentra en la carpeta 'Documentos'.
dirección <- './Privado/GeolocLocalidadesTOTAL.tsv'
data <- read.table(file = dirección, sep = '\t', header = TRUE, stringsAsFactors = FALSE)
data <- data.frame(Nombre=data$Nombre, Año=data$Año, Localidad=data$Localidad.1, Latitud=
data$Latitud.1, Longitud=data$Longitud.1, stringsAsFactors = FALSE)
```

## Limpieza de datos

Nuestra tabla de datos contiene algunos que no deben ser tenidos en cuenta por no tratarse de entradas del censo, sino los nombres de los empadronadores. Estos datos tienen asociados el texto `'None'` en *Localidad*; los quitaremos.

```
data <- data[!data$Localidad=="None",]
data <- transform(data, Latitud = as.numeric(Latitud), Longitud = as.numeric(Longitud))
```

# 2. Filtrar por apellido y año.

Para ello creamos y aplicamos la función `filtro` .

## Función `filtro`

Esta función nos servirá para extraer todos los datos de un determinado *año* y un determinado *apellido* o *nombre*. Para filtrar el apellido introduciremos todas las variantes posibles que se encuentre en los datos; por ejemplo, para el apellido “Fernández” podemos escrir como variantes “Fdez”, “Fernandez” (sin tilde) y “Fernández”.

### Sintaxis

```
subdata <- filtro(data, nombres, valores)
```

### Valor de los parámetros

| Parámetro | Tipo | Descripción |
|---|---|---|
| data | data.frame | Son los datos sin filtrar. |

| Parámetro | Tipo | Descripción |
|---|---|---|
| nombres | vector | Nombres de las columnas de datos a filtrar. |
| valores | list | Cada entrada contiene un vector de posibilidades para la etiqueta para la etiqueta correspondiente, p. ej. “Fdez” y “Fernández”. |

### Ejemplo de uso

```
subdata <- filtro(data, c("Nombre", "Año"), list(c("Fdez", "Fernandez", Fernández"), c("1773")))
```

### Código

```
filtro <- function(data, nombres, valores)
{
  library(stringr)
  for (elemento in 1:length(nombres))
  {
    nombre <- nombres[elemento]
    valor <- valores[[elemento]]
    match <- str_detect(data[,nombre], paste(valor, collapse="|"))
    data$matches <- match
    data <- data[!data$matches==FALSE,]
    data <- data[,!(names(data) %in% c("matches"))]
  }
  result <- data
}
```

## Aplicamos el filtro

```
patrón <- c("Rúa", "Rua", "rua", "rúa")
subdata1698 <- filtro(data, c("Nombre", "Año"), list(patrón, c("1698")))
subdata1773 <- filtro(data, c("Nombre", "Año"), list(patrón, c("1773")))
```

# 3. Ajustar el hiperparámetro k y aplicar k-Medias.

## Determinar los datos de estudio

Usaremos como parámetros de estudio la *Latitud*, la *Longitud* y la *Altitud*, serán la variable `X` . Estos tres datos comparten la misma *localidad*, que será en valor con el que compararemos la pertencia a uno u otro cluster en la matriz de confusión. La *Localidad* será la variable `Y` .

```
X1698 <- subdata1698[, c("Latitud", "Longitud")]
Y1698 <- subdata1698$Localidad

X1773 <- subdata1773[, c("Latitud", "Longitud")]
Y1773 <- subdata1773$Localidad
```

## Ajuste de k

El experimento consiste en calcular k-Medias para una cantidad de centros de 1 a 10. El experimento se repetirá 100 veces. Primero lo haremos para los datos del 1698.

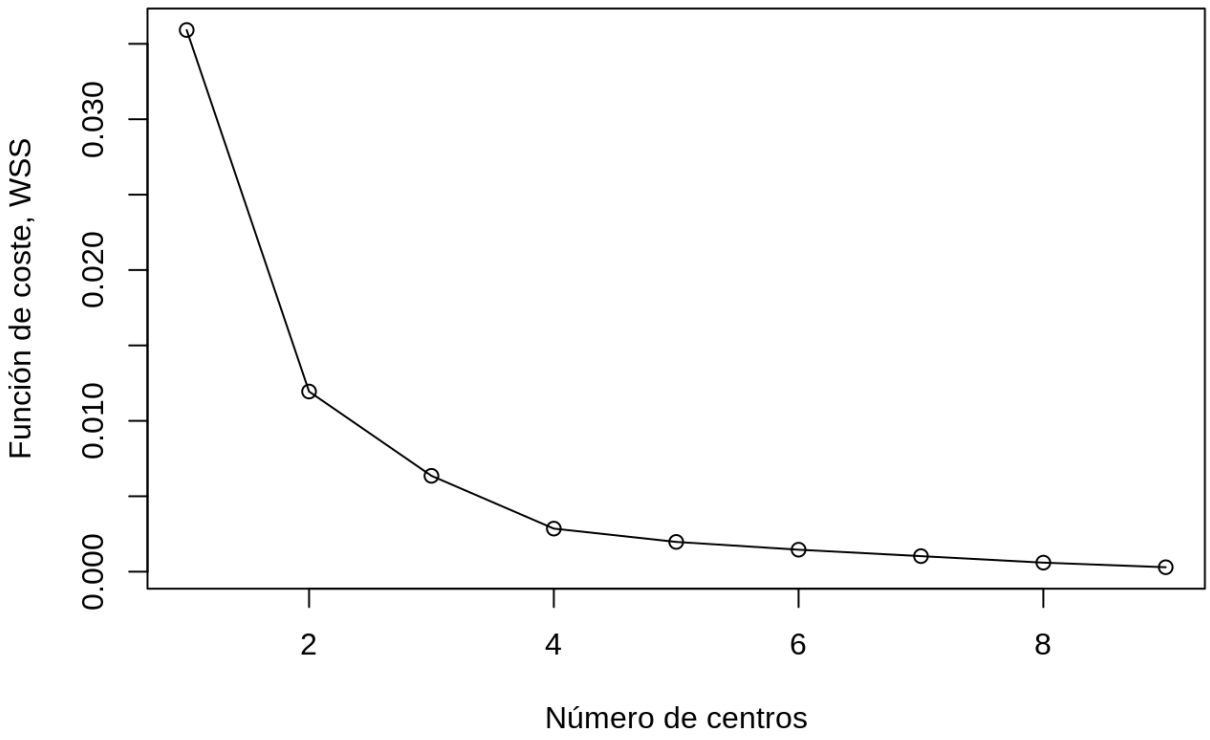
```
M <- 100
N <- 9
w1698 <- matrix(rep(0, N), nrow=N, ncol=M)
for (j in 1:M)
{
  for (i in 1:N)
  {
    kc1698 <- kmeans(X1698, centers=i, iter.max=999)
    w1698[i, j] <- kc1698$tot.withinss
  }
}
```

Para cada número de centros, se representa la mediana, con la que se aplicará el método de heurístico, explicado a continuación; pero también representamos el histograma de dicho número de grupos o centros.

## La mediana

```
library(matrixStats)
```

```
medianas1698 <- rowMedians(w1698)
plot(medianas1698, type='o', xlab='Número de centros',
     ylab='Función de coste, WSS')
```




####

Ecogemos un valor de k Usaremos el método del “codo”: para ello hemos representado la mediana de la WSS (within-cluster sum of squares: suma de cuadrados dentro del cluster) de los experimentos. El número de centros, k, sugerido por el método es el entero más pequeño para el que la figura anterior se aplana. Haciendo una similitud entre la figura de las medias y un brazo, el valor de k “correponde con la posición del codo”. Para más información vea esta página web (https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/).

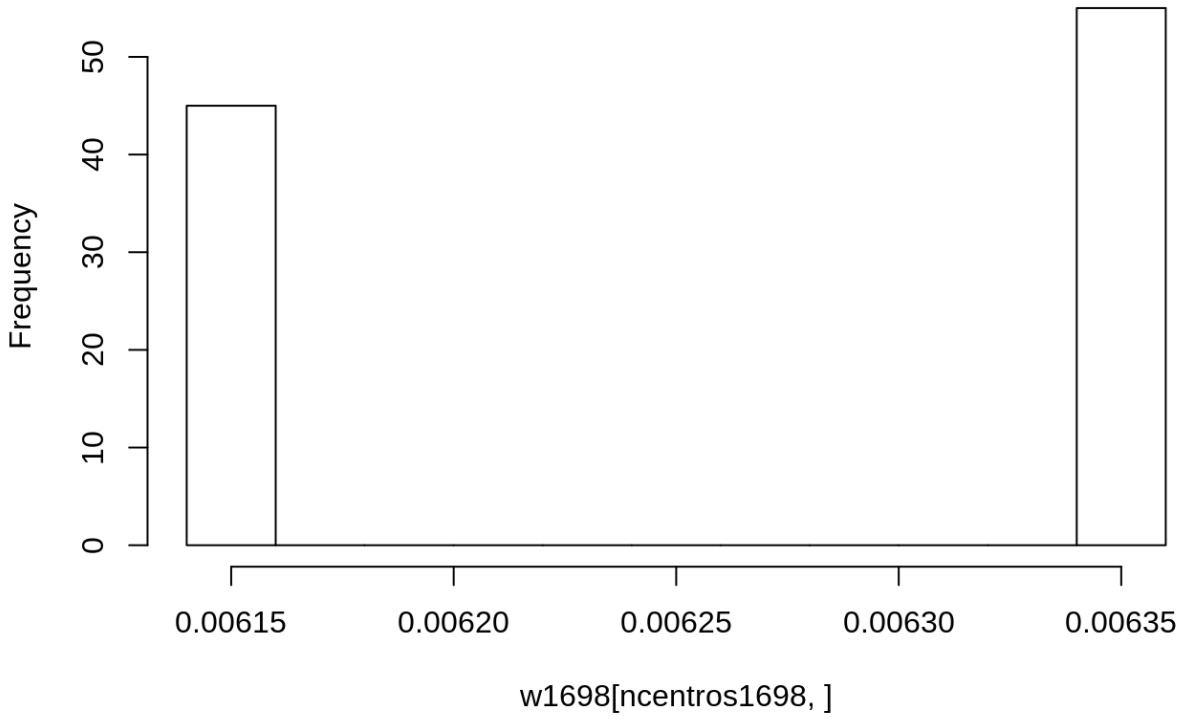
Además, véase que hemos hecho el experimento varias veces para asegurar la coherencia de los resultados. En efecto, al representar el histograma de k=“valor del codo” observamos que la mayoría de veces se obtiene un WSS bajo, lo que sugiere una solución estable. Si se representa el mismo histograma para un k menor, se observa una

variabilidad mucho más grande.

## El histograma de un centro

Escoja un `centro` determinado y calcule su histograma.

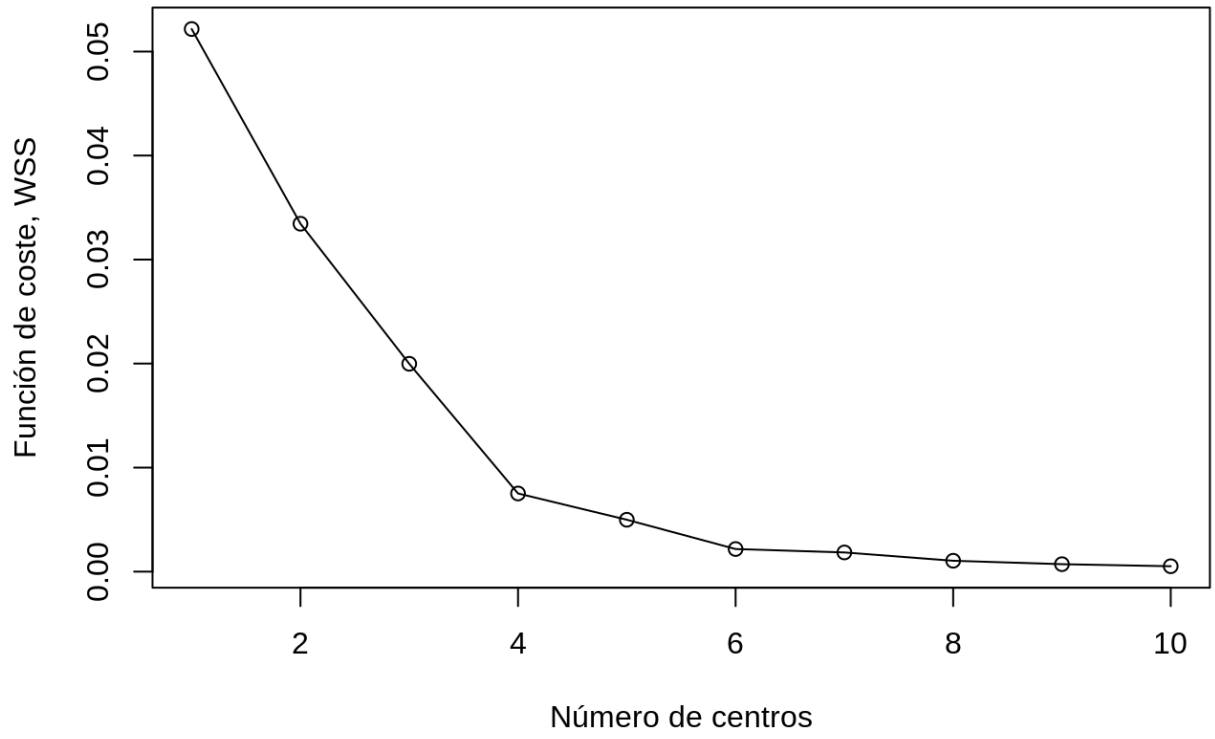
```
ncentros1698 <- 3
hist(w1698[ncentros1698,])
```




## Repetición del proceso para el censo de 1773

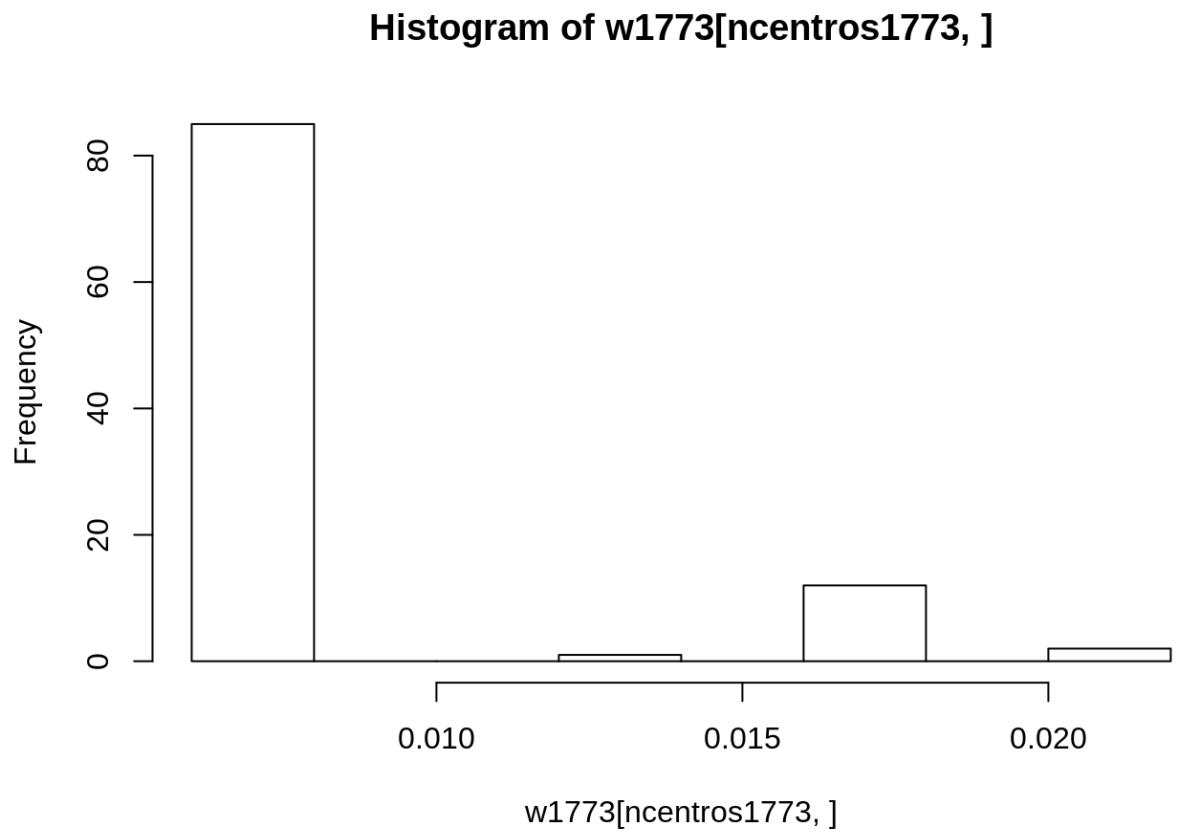
```
M <- 100
N <- 10
w1773 <- matrix(rep(0, N), nrow=N, ncol=M)
for (j in 1:M)

{
  for (i in 1:N)
  {
    kc1773 <- kmeans(X1773, centers=i, iter.max=999)
    w1773[i, j] <- kc1773$tot.withinss
  }
}
medianas1773 <- rowMedians(w1773)
plot(medianas1773, type='o', xlab='Número de centros',
     ylab='Función de coste, WSS')
```




Claramente el valor k del “codo” es 4, aunque en muchas ocasiones se puede ser flexible con este método y escoger un k similar, que se adapte mejor a la agrupación visual de los datos.

```
ncentros1773 <- 4
hist(w1773[ncentros1773, ])
```



# Aplicación de las k-Medias

En este caso vamos a usar como parámetros la *Latitud*, *Longitud* y *Altitud*.

```
kc1698 <- kmeans(X1698, ncentros1698, nstart=5*ncentros1698)
kc1773 <- kmeans(X1773, ncentros1773, nstart=5*ncentros1773)
```

La matriz de confusión nos muestra la distribución de las entradas en cada cluster en función de la *localidad* a la que pertenecen. Veámoslo para 1698

```
table(Y1698, kc1698$cluster, dnn=c("Localidades", "Cluster nº"))
```

```
##                 Cluster nº
## Localidades       1 2 3
##   Arbeyales       1 0 0
##   Baldedo         2 0 0
##   Berducedo       1 0 0
##   Carcedo de Lago 1 0 0
##   Cornollo        0 1 0
##   Is              1 0 0
##   San Emiliano    0 0 1
##   San Pedro       1 0 0
##   Teijedo         1 0 0
##   Tremado         0 1 0
##   Villanueva      0 1 0
```

Veamos cúantas entradas de cada localidad tenemos en 1698.

```
Localidades1698 <- as.factor(subdata1698$Localidad)
table(Localidades1698)
```

```
## Localidades1698
##       Arbeyales         Baldedo       Berducedo Carcedo de Lago
##               1               2               1               1
##        Cornollo              Is    San Emiliano       San Pedro
##               1               1               1               1
##         Teijedo         Tremado      Villanueva
##               1               1               1
```

Lo repetimos para 1773:

```
table(Y1773, kc1773$cluster, dnn=c("Localidades", "Cluster nº"))
```

```
##              Cluster nº
## Localidades    1 2 3 4
##   Arbeyales    0 0 0 1
##   Baldedo      0 0 0 1
##   Berducedo    0 0 0 1
##   Beveraso     1 0 0 0
##   Bustantigo   0 0 1 0
##   Castro, El   0 0 0 2
##   Coba         0 1 0 0
##   Herías       1 0 0 0
##   San Emiliano 1 0 0 0
##   Sarzol       1 0 0 0
##   Teijedo      0 0 0 2
##   Tremado      0 1 0 0
```

```
Localidades1773 <- as.factor(subdata1773$Localidad)
table(Localidades1773)
```

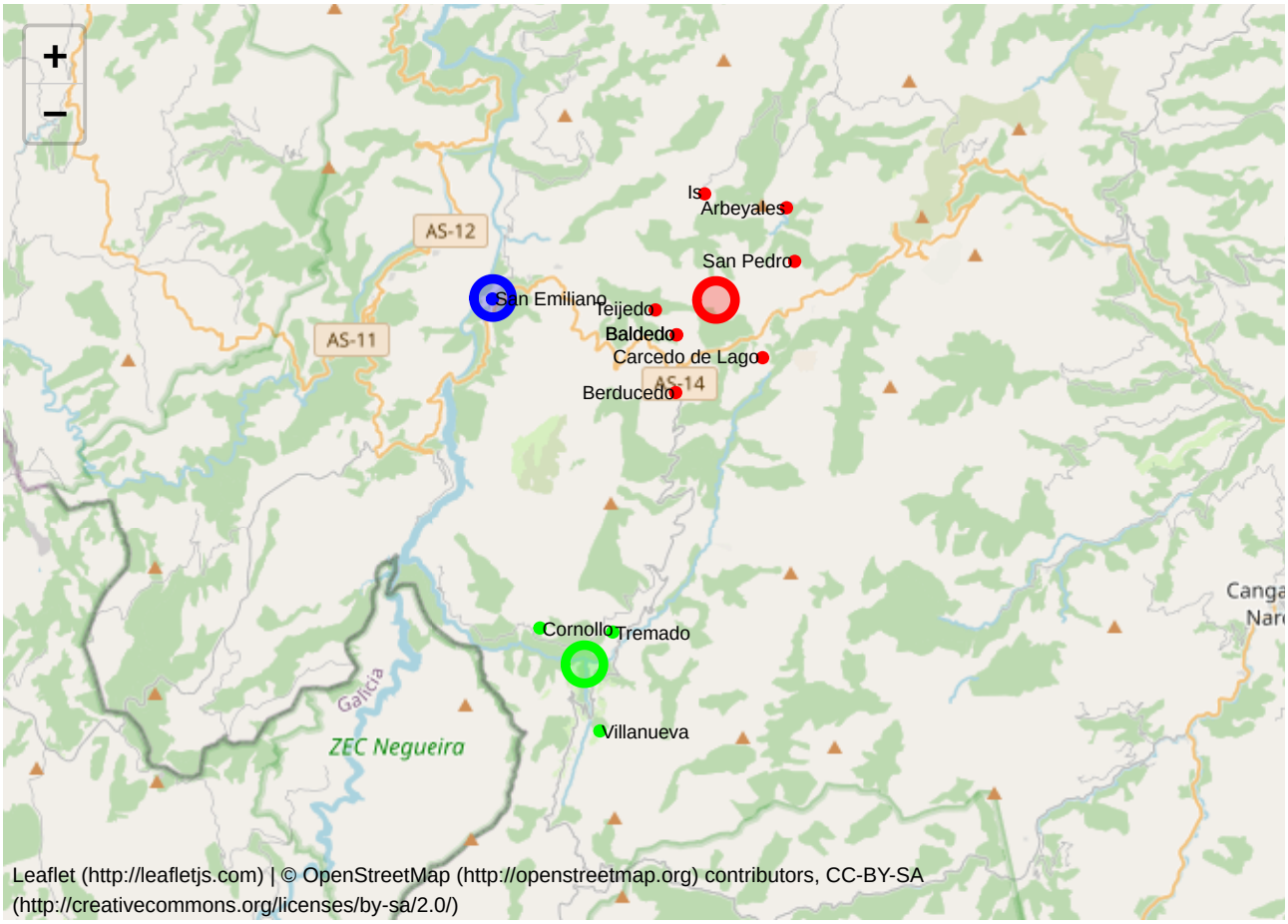
```
## Localidades1773
##    Arbeyales      Baldedo    Berducedo     Beveraso   Bustantigo
##            1            1            1            1            1
##   Castro, El         Coba       Herías San Emiliano       Sarzol
##            2            1            1            1            1
##      Teijedo      Tremado
##            2            1
```

# 4. Representación en un mapa.

Representaremos en azul los centroides obtenidos por el algoritmo de las k-Medias. En rojo cada una de las entradas del censo usadas ubicadas en su correspondiente localidad. Usaremos la librería `leaflet` . Primero representamos el mapa para 1698.

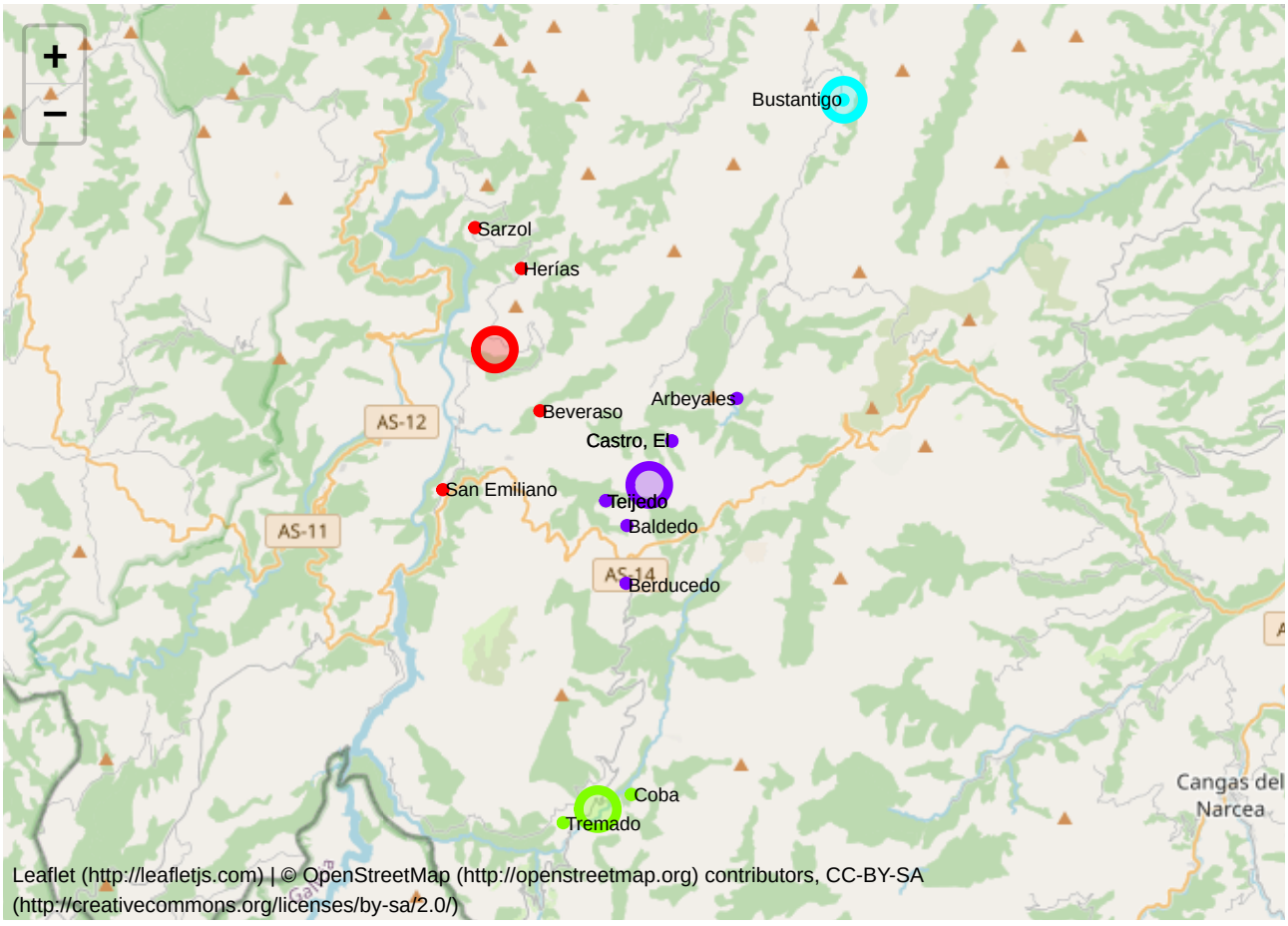
```
library(leaflet)
```

```
pal <- colorFactor(palette= rainbow(ncentros1698), domain = c(1:ncentros1698))
coords1698 <- data.frame(kc1698$centers[,c("Latitud", "Longitud")])
leaflet(subdata1698) %>% addTiles() %>%
    addCircleMarkers(lng = ~coords1698$Longitud,
              lat = ~coords1698$Latitud,
              weight = 5,
              col= ~pal(c(1:ncentros1698)),
              label = c(1:ncentros1698),
              labelOptions = labelOptions(noHide = F, textOnly = FALSE),
              fillOpacity = 0.25,
              opacity = 1,
              stroke = TRUE)  %>%
  addCircles(lng = ~subdata1698$Longitud,
              lat = ~subdata1698$Latitud,
              weight = 5,
              color = ~pal(kc1698$cluster),
              opacity = 1,
              fillOpacity = 1) %>%
  addLabelOnlyMarkers(lng = ~subdata1698$Longitud,
                      lat = ~subdata1698$Latitud,
                      label = ~subdata1698$Localidad,
                      labelOptions = labelOptions(noHide = T, textOnly = TRUE))
```



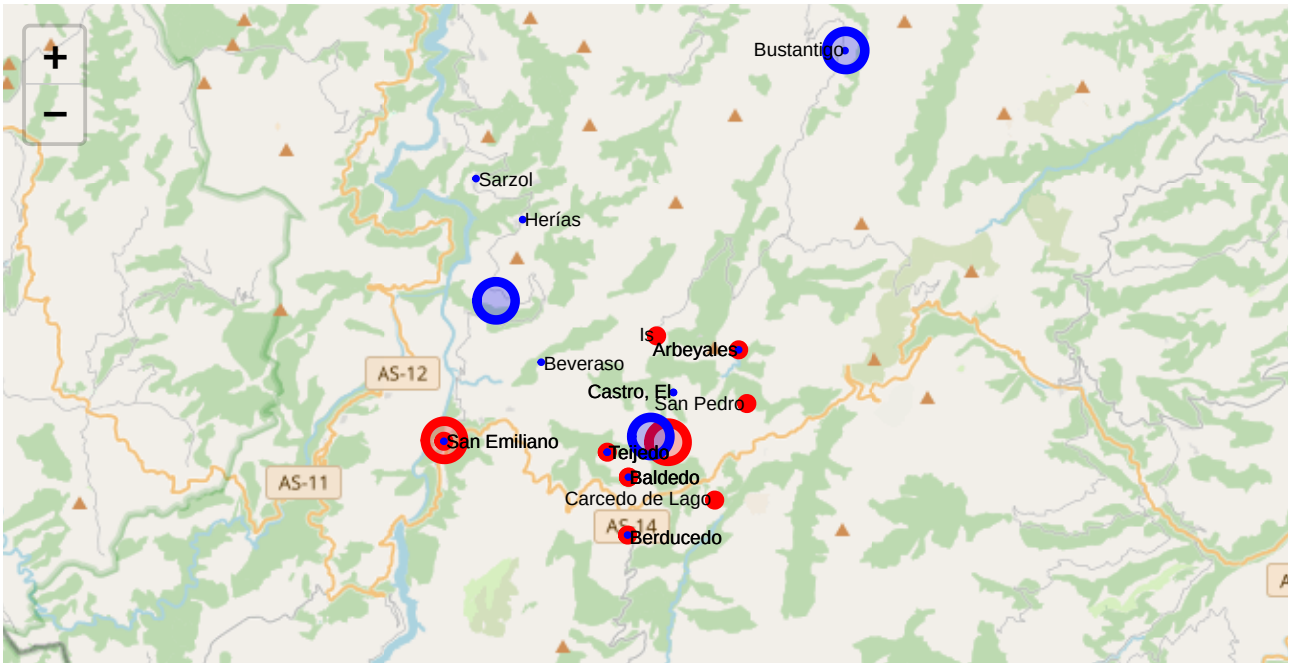

Ahora para 1773

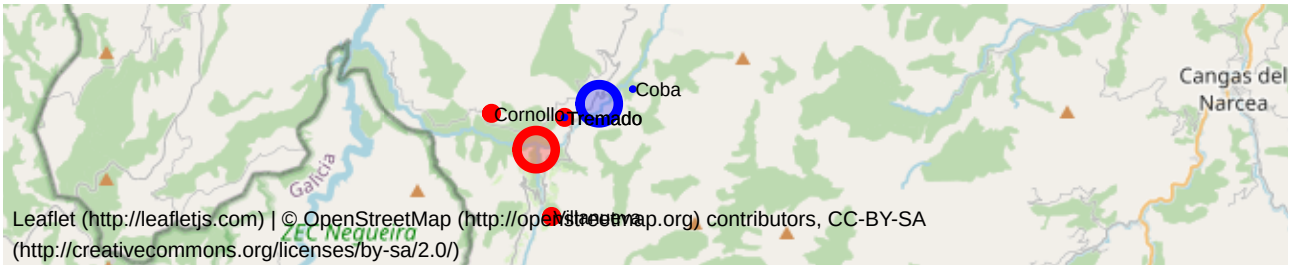
```
pal <- colorFactor(palette= rainbow(ncentros1773), domain = c(1:ncentros1773))
coords1773 <- data.frame(kc1773$centers[,c("Latitud", "Longitud")])
leaflet(subdata1773) %>% addTiles() %>%
  addCircleMarkers(lng = ~coords1773$Longitud,
             lat = ~coords1773$Latitud,
             weight = 5,
             col= ~pal(c(1:ncentros1773)),
             label = c(1:ncentros1773),
             labelOptions = labelOptions(noHide = F, textOnly = FALSE),
             fillOpacity = 0.25,
             opacity = 1,
             stroke = TRUE)   %>%
  addCircles(lng = ~subdata1773$Longitud,
             lat = ~subdata1773$Latitud,
             weight = 5,
             color = ~pal(kc1773$cluster),
             opacity = 1,
             fillOpacity = 1) %>%
  addLabelOnlyMarkers(lng = ~subdata1773$Longitud,
                      lat = ~subdata1773$Latitud,
                      label = ~subdata1773$Localidad,
                      labelOptions = labelOptions(noHide = T, textOnly = TRUE))
```




# Ahora se representan los clusters de 1698 en rojo y los de 1773 en azul

```
pal <- colorFactor(palette= rainbow(3), domain = c(1:3))
leaflet(subdata1698) %>% addTiles() %>%
    addCircleMarkers(lng = ~coords1698$Longitud,
              lat = ~coords1698$Latitud,
              weight = 5,
              col= ~pal(1),
              label = c(1:ncentros1698),
              labelOptions = labelOptions(noHide = F, textOnly = FALSE),
              fillOpacity = 0.25,
              opacity = 1,
              stroke = TRUE)  %>%
  addCircleMarkers(lng = ~coords1773$Longitud,
              lat = ~coords1773$Latitud,
              weight = 5,
              col= ~pal(3),
              label = c(1:ncentros1773),
              labelOptions = labelOptions(noHide = F, textOnly = FALSE),
              fillOpacity = 0.25,
              opacity = 1,
              stroke = TRUE)     %>%
  addCircles(lng = ~subdata1698$Longitud,
              lat = ~subdata1698$Latitud,
              weight = 8,
              color = ~pal(1),
              opacity = 1,
              fillOpacity = 1) %>%
  addLabelOnlyMarkers(lng = ~subdata1698$Longitud,
                      lat = ~subdata1698$Latitud,
                      label = ~subdata1698$Localidad,
                      labelOptions = labelOptions(noHide = T, textOnly = TRUE)) %>%
  addCircles(lng = ~subdata1773$Longitud,
              lat = ~subdata1773$Latitud,
              weight = 2,
              color = ~pal(3),
              opacity = 1,
              fillOpacity = 1) %>%
  addLabelOnlyMarkers(lng = ~subdata1773$Longitud,
                      lat = ~subdata1773$Latitud,
                      label = ~subdata1773$Localidad,
                      labelOptions = labelOptions(noHide = T, textOnly = TRUE))
```




Coba
Cornollo
Tremado
Cangas del Narcea
Galicia
ZEC Nequeiro
Leaflet (http://leafletjs.com) | © OpenStreetMap (http://openstreetmap.org) contributors, CC-BY-SA (http://creativecommons.org/licenses/by-sa/2.0/)