

# Clase 4

Sergio Andrés Hernández

17/3/2022

## Estadísticas descriptivas

```
#Hoja de calculo disponible en: "https://docs.google.com/spreadsheets/d/11FRDO_mRtu_r3mbduePRlGmRCZMK9-
library(readxl)
XPABLO <- read_excel("C:/Users/57321/Downloads/XPABLO.XLSX")
XPABLO <- XPABLO[-c(16,17,18,19)]
View(XPABLO)

#XPABLO <- read_excel("Cargar ruta del archivo")
#XPABLO <- XPABLO[-c(16,17,18,19)]
#View(XPABLO)
#df = data.frame(calcio = rnorm(n = , mean = , sd = )) # Dataframe -> Marco de datos, de forma tabular

colnames(XPABLO)
```

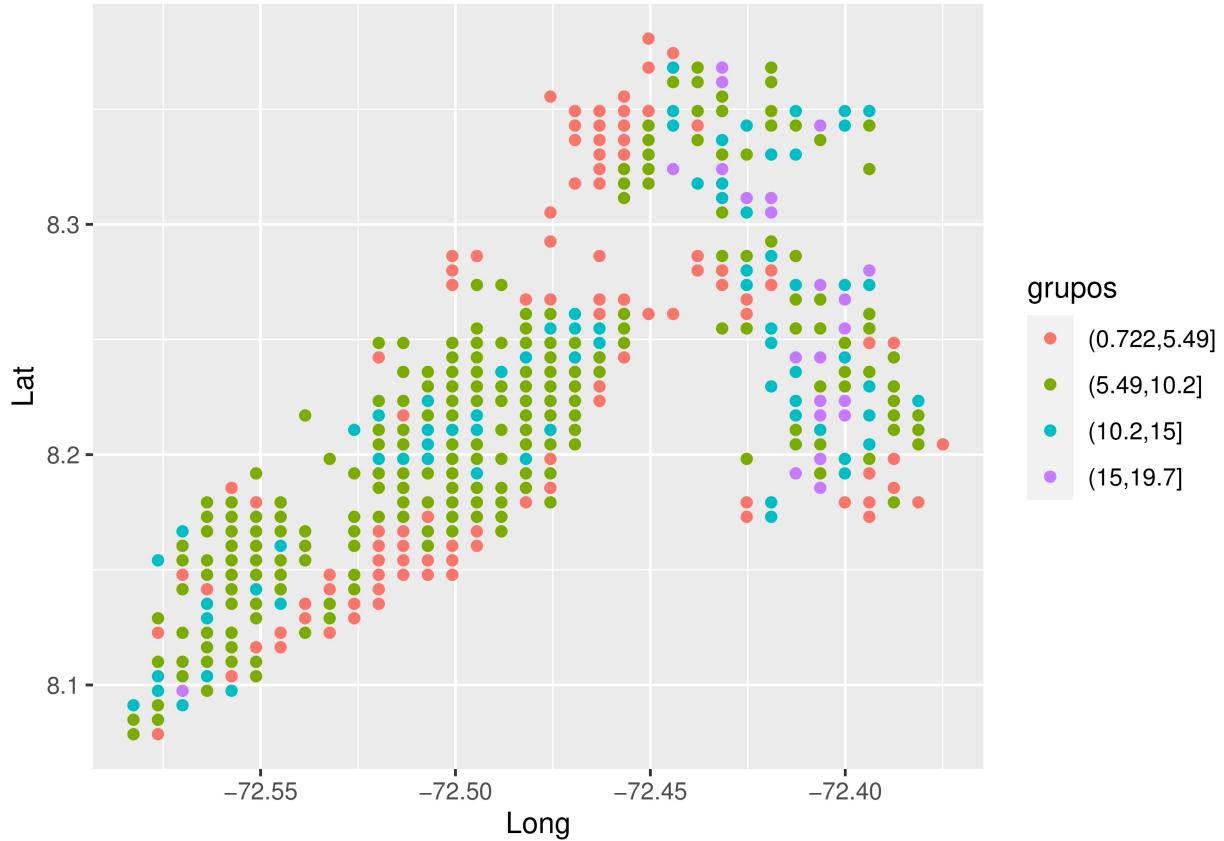
## Nombre de las columnas de la base de datos

```
## [1] "id"      "Long"    "Lat"     "z"       "M0"      "Ca"      "Mg"      "K"       "Na"      "CICE"
## [11] "CE"      "Fe"      "Cu"      "Zn"      "cos"
```

**Función para crear categorías** La función `unic()` en R se utiliza para eliminar o eliminar los valores duplicados o las filas presentes en el vector, marco de datos o matriz también.

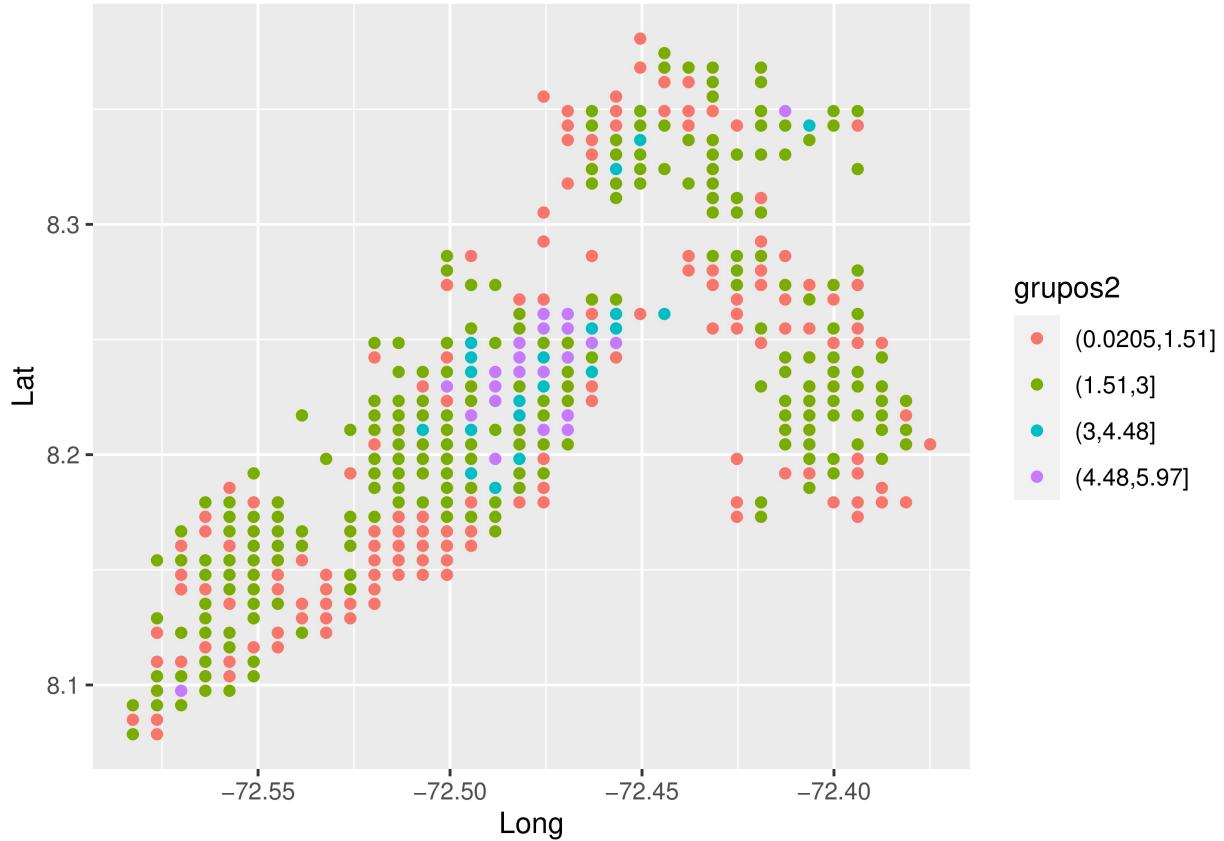
```
grupos <- cut(x = XPABLO$Ca, breaks = 4) # cut -> Función para crear categorías
unique(grupos)
```

```
## [1] (5.49,10.2]  (0.722,5.49] (10.2,15]    (15,19.7]
## Levels: (0.722,5.49] (5.49,10.2] (10.2,15] (15,19.7]
ggplot(XPABLO, aes(x = Long, y = Lat, color = grupos)) + #Aesthetic
  geom_point()
```



```
grupos2 <- cut(x = XPABLO$Mg, breaks = 4) # cut -> Función para crear categorías
unique(grupos2)
```

```
## [1] (1.51,3]      (0.0205,1.51] (4.48,5.97]   (3,4.48]
## Levels: (0.0205,1.51] (1.51,3] (3,4.48] (4.48,5.97]
ggplot(XPABLO, aes(x = Long, y = Lat, color = grupos2)) + #Aesthetic
  geom_point()
```



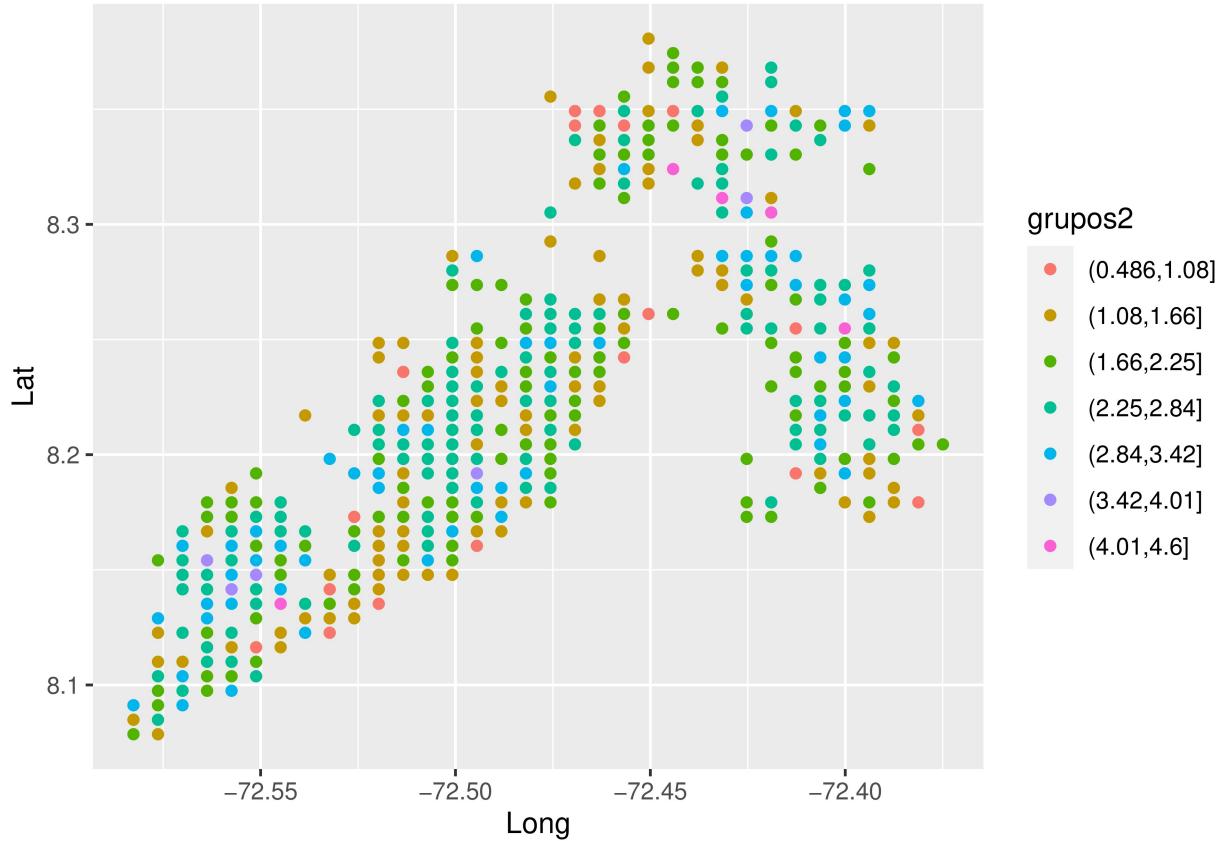
## Asignación (Tarea)

Mapear (con coordenadas) el resto de variables, no necesariamente con 4 grupos

*MO*

```
grupos2 <- cut(x = XPABLO$MO, breaks = 7) # cut -> Función para crear categorías
unique(grupos2)
```

```
## [1] (1.66,2.25]  (1.08,1.66]  (2.25,2.84]  (2.84,3.42]  (0.486,1.08]
## [6] (4.01,4.6]   (3.42,4.01]
## 7 Levels: (0.486,1.08] (1.08,1.66] (1.66,2.25] (2.25,2.84] ... (4.01,4.6]
ggplot(XPABLO, aes(x = Long, y = Lat, color = grupos2)) + #Aesthetic
  geom_point()
```

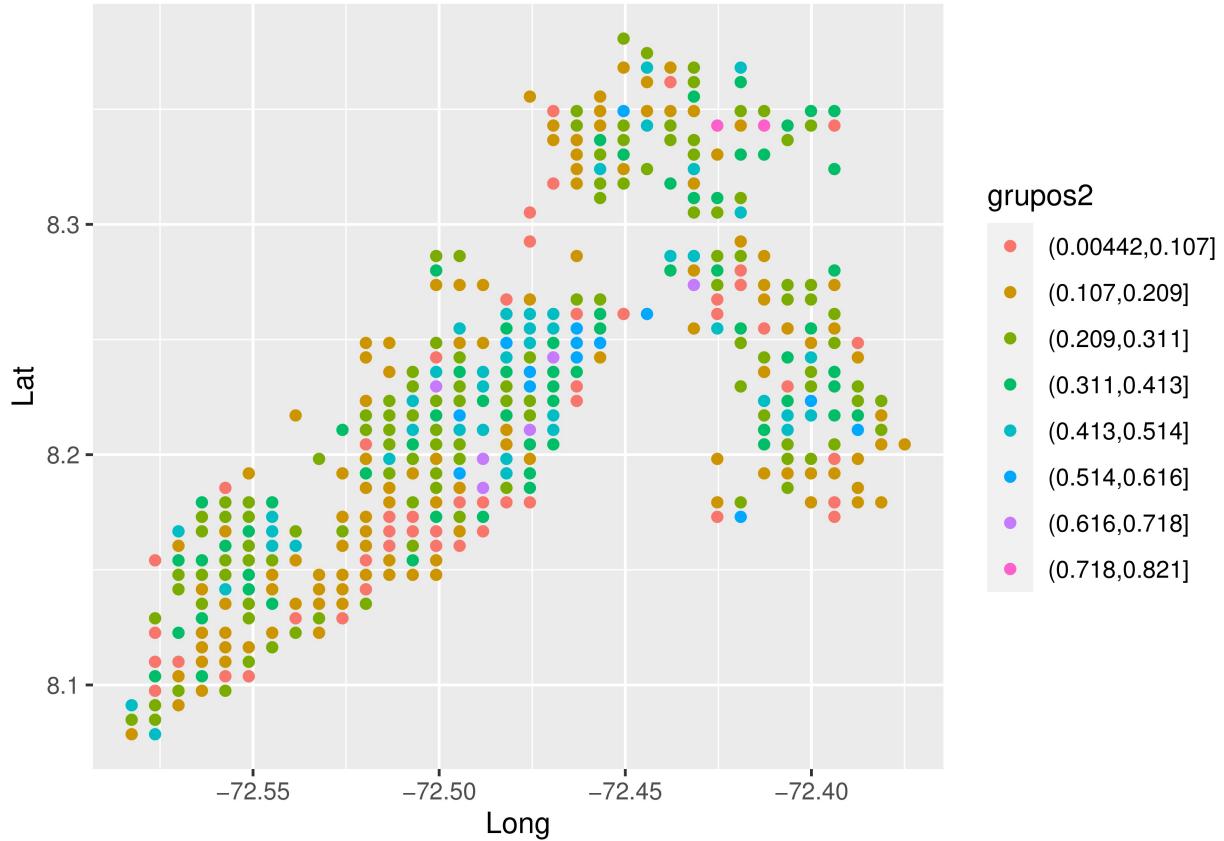


*CICE*

```
grupos2 <- cut(x = XPABLO$K, breaks = 8) # cut -> Función para crear categorías
unique(grupos2)
```

```
## [1] (0.107,0.209]   (0.413,0.514]   (0.209,0.311]   (0.00442,0.107]
## [5] (0.311,0.413]   (0.514,0.616]   (0.616,0.718]   (0.718,0.821]
## 8 Levels: (0.00442,0.107] (0.107,0.209] (0.209,0.311] ... (0.718,0.821]
```

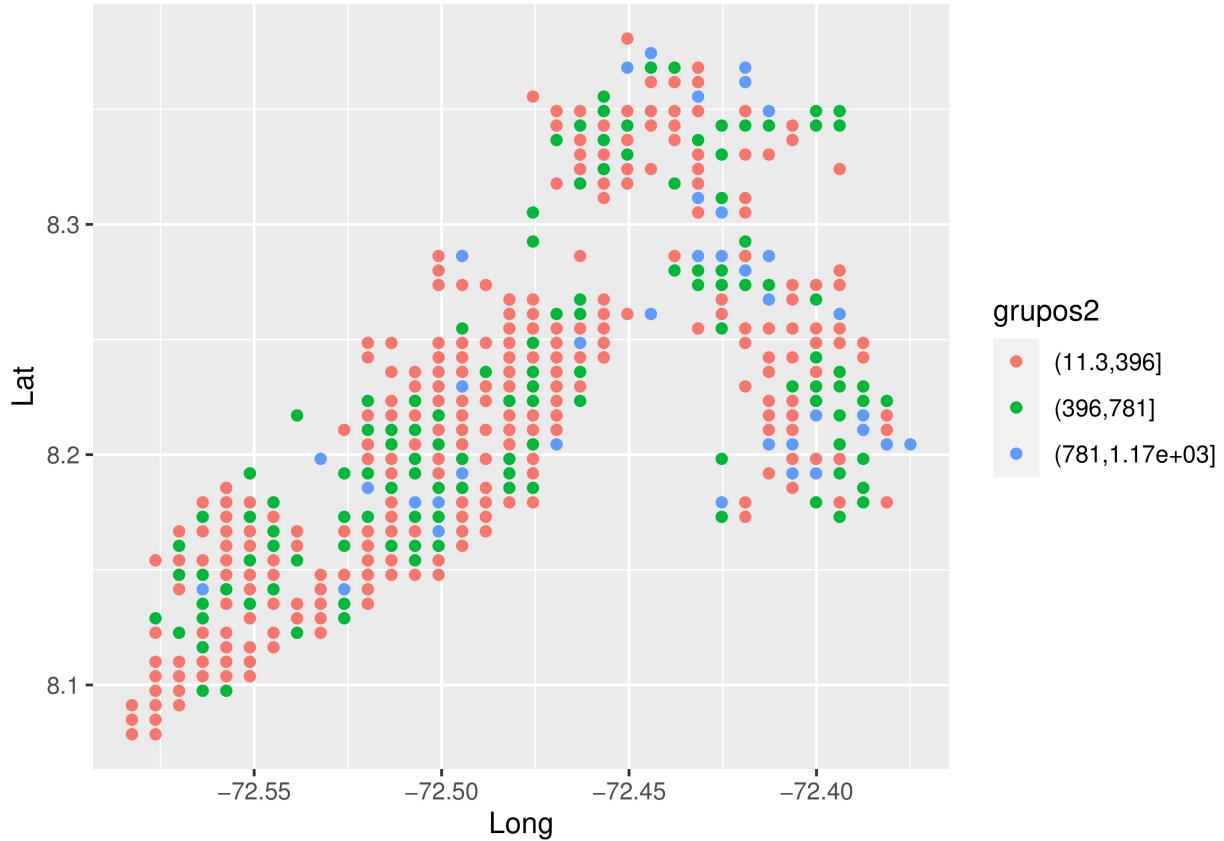
```
ggplot(XPABLO, aes(x = Long, y = Lat, color = grupos2)) + #Aesthetic
  geom_point()
```



*Fe*

```
grupos2 <- cut(x = XPABLO$Fe, breaks = 3) # cut -> Función para crear categorías
unique(grupos2)
```

```
## [1] (11.3,396]      (396,781]      (781,1.17e+03]
## Levels: (11.3,396] (396,781] (781,1.17e+03)
ggplot(XPABLO, aes(x = Long, y = Lat, color = grupos2)) + #Aesthetic
  geom_point()
```

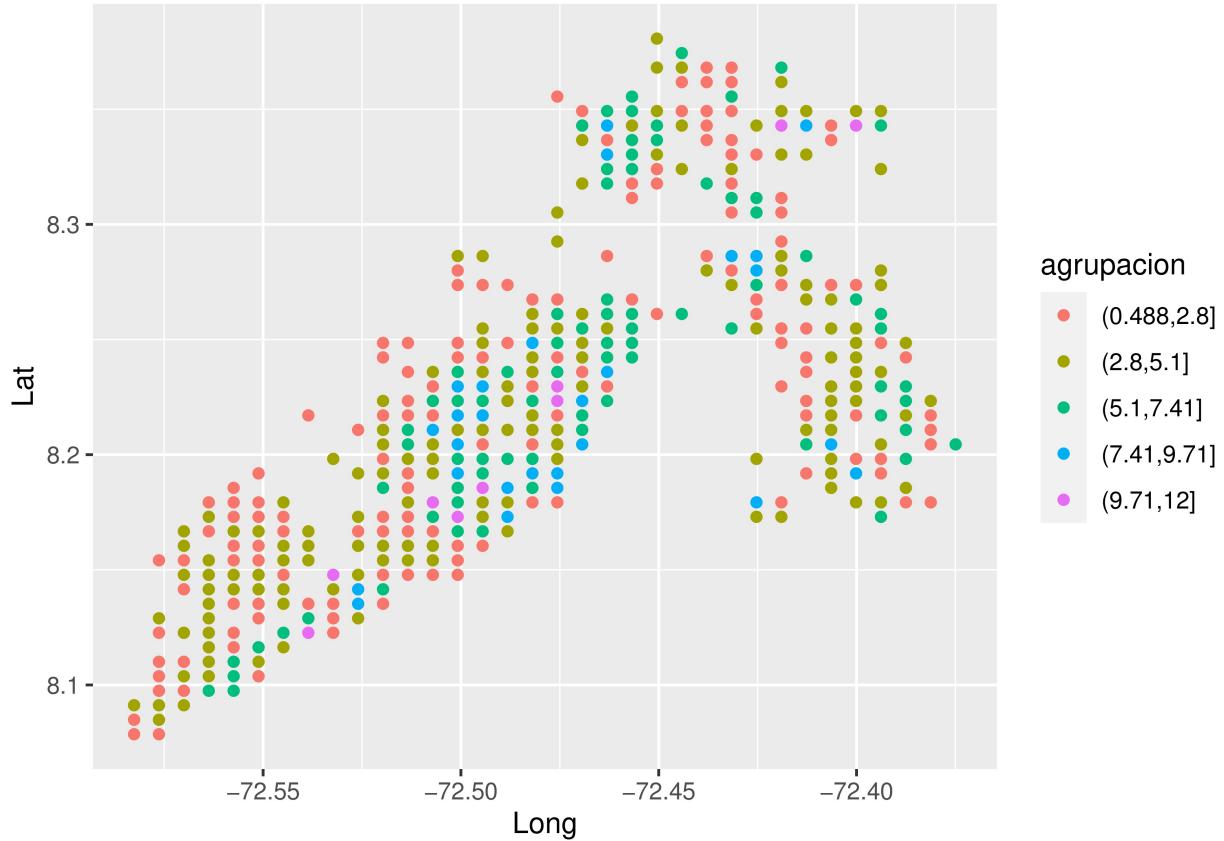


*Zn*

```
agrupacion<-cut(x=XPABLO$Zn, breaks=5)
unique(agrupacion)
```

```
## [1] (0.488,2.8] (2.8,5.1] (5.1,7.41] (9.71,12] (7.41,9.71]
## Levels: (0.488,2.8] (2.8,5.1] (5.1,7.41] (7.41,9.71] (9.71,12]
```

```
ggplot(XPABLO, aes(x=Long, y=Lat, color=agrupacion))+
  geom_point()
```



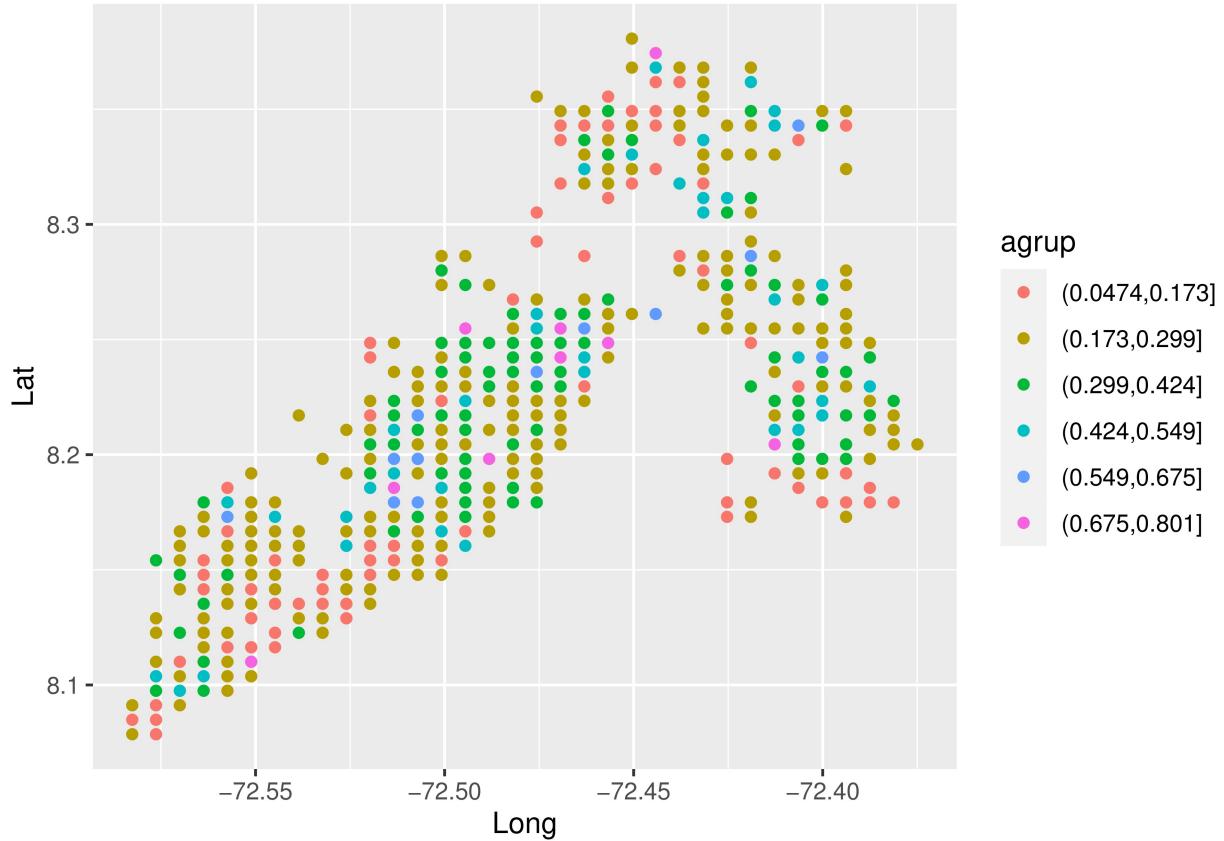
*Na*

```

agrup<-cut(x=XPABLO$Na, breaks=6)
unique(agrup)

## [1] (0.173,0.299]  (0.0474,0.173] (0.299,0.424]  (0.424,0.549]  (0.675,0.801]
## [6] (0.549,0.675]
## 6 Levels: (0.0474,0.173] (0.173,0.299] (0.299,0.424] ... (0.675,0.801]
ggplot(XPABLO, aes(x=Long, y=Lat, color=agrup))+
  geom_point()

```



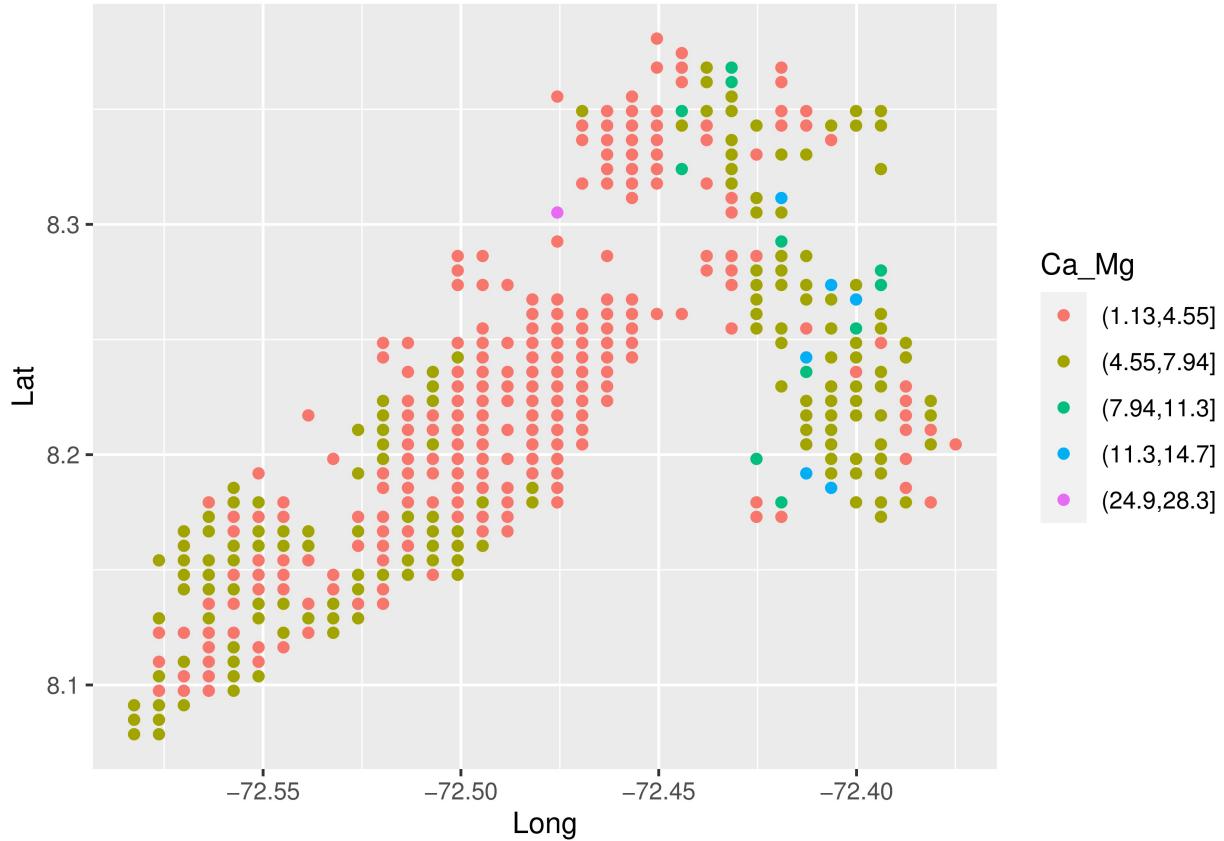
## 2. Simplificación de las matrices de datos (DRISS)

Relación

$$\frac{\text{Calcio}}{\text{Magnesio}}$$

```
Ca_Mg = XPABLO$Ca/XPABLO$Mg
Ca_Mg = cut(Ca_Mg, breaks = 8)
unique(Ca_Mg )
```

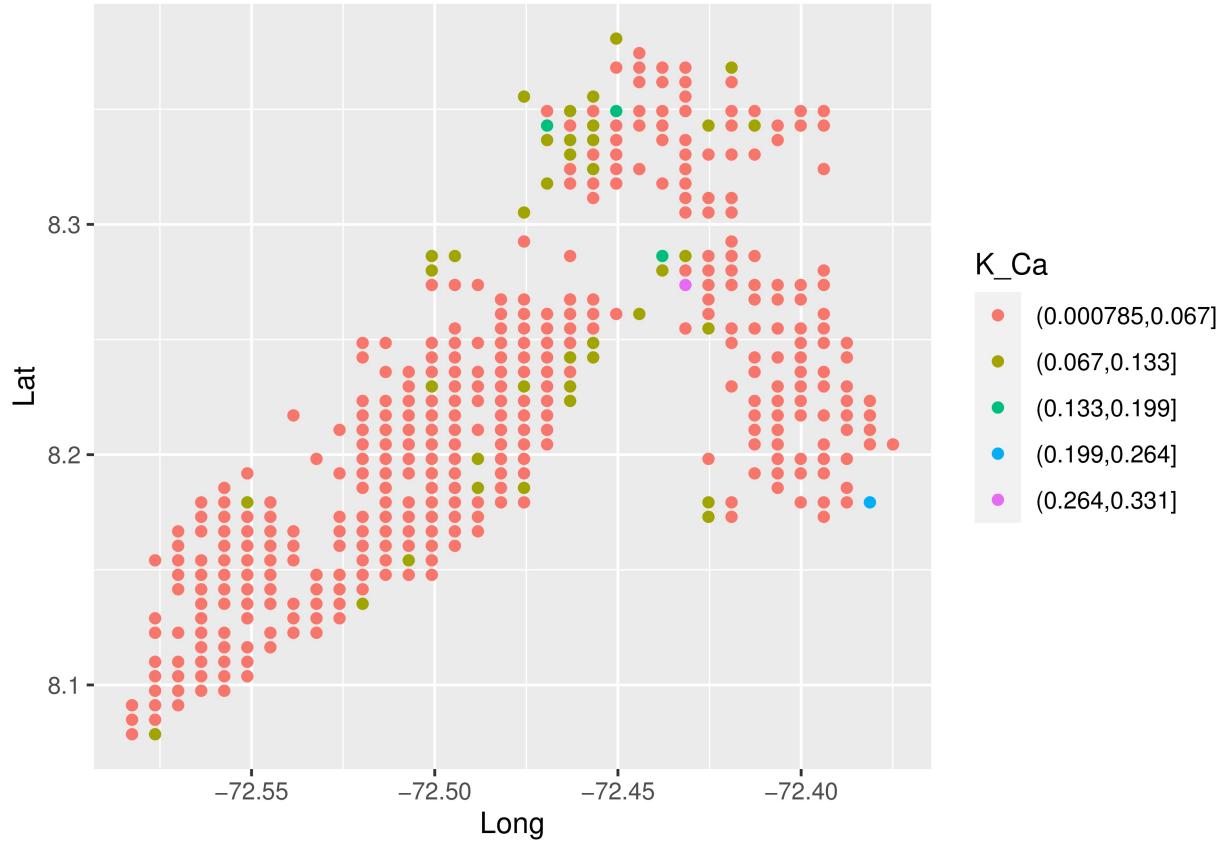
```
## [1] (4.55,7.94] (1.13,4.55] (7.94,11.3] (11.3,14.7] (24.9,28.3]
## 8 Levels: (1.13,4.55] (4.55,7.94] (7.94,11.3] (11.3,14.7] ... (24.9,28.3]
ggplot(XPABLO, aes(Long, Lat, color = Ca_Mg))+
  geom_point()
```



Relación:

$$\frac{\text{Potasio}}{\text{Calcio}}$$

```
K_Ca = XPABLO$K/XPABLO$Ca
K_Ca = cut(x = K_Ca, breaks = 5)
ggplot(XPABLO, aes(Long, Lat, color = K_Ca))+
  geom_point()
```

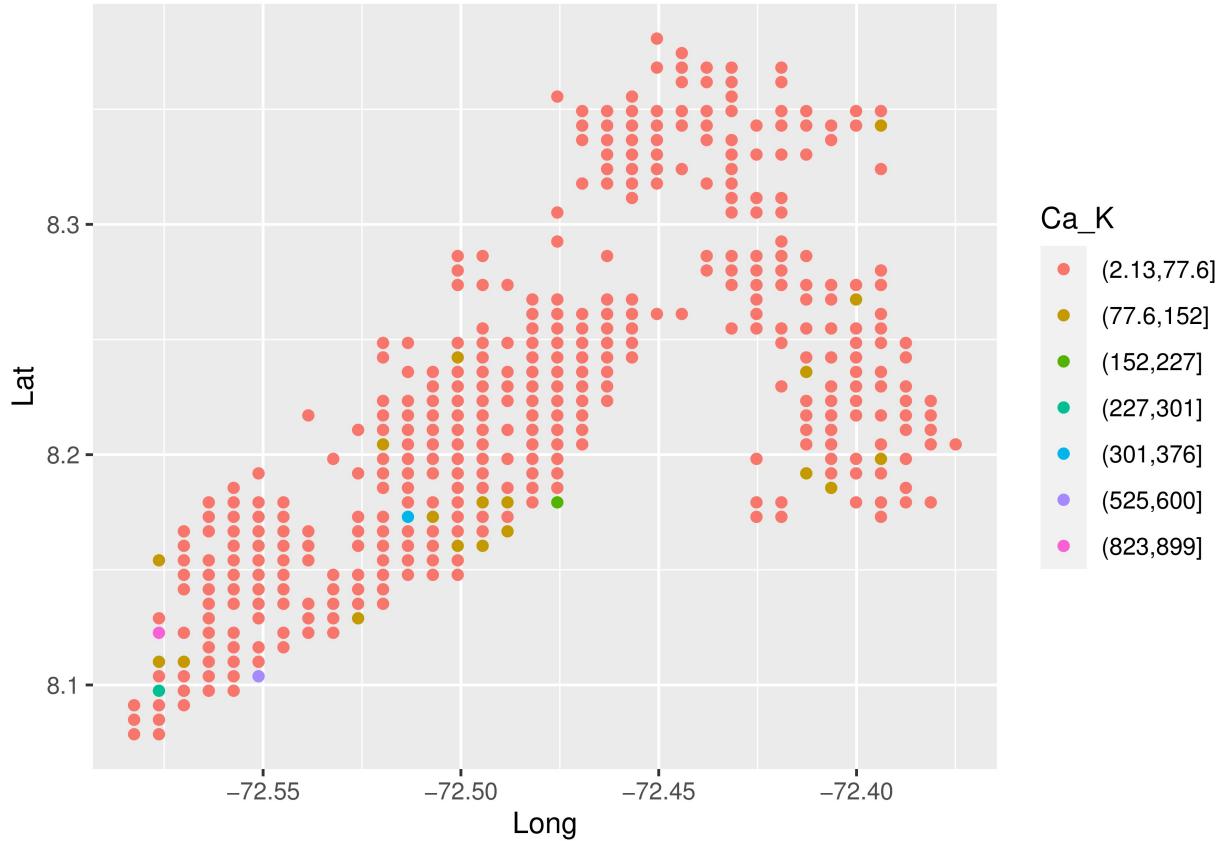


Relación

$$\frac{\text{Calcio}}{\text{Potasio}}$$

```

Ca_K = XPABLO$Ca/XPABLO$K
Ca_K = cut(x = Ca_K, breaks = 12)
ggplot(XPABLO, aes(Long, Lat, color = Ca_K))+
  geom_point()
  
```



## Creando distribución cuartil

```

Ca_K = XPABLO$Ca/XPABLO$K
nuevos_grupos <- quantile(x = Ca_K, probs = c(0.25, 0.5, 0.75))
nuevos_grupos

##      25%      50%      75%
## 21.99456 31.76159 45.37446
q_1 = nuevos_grupos[1]; q_1

##      25%
## 21.99456
q_2 = nuevos_grupos[2]; q_2

##      50%
## 31.76159
q_3 = nuevos_grupos[3]; q_3

##      75%
## 45.37446

data1 <- subset(Ca_K, Ca_K < q_1); data1
data2 <- subset(Ca_K, Ca_K < q_2 & Ca_K > q_1); data2
data3 <- subset(Ca_K, Ca_K < q_3 & Ca_K > q_2); data3
data4 <- subset(Ca_K, Ca_K > q_3); data4

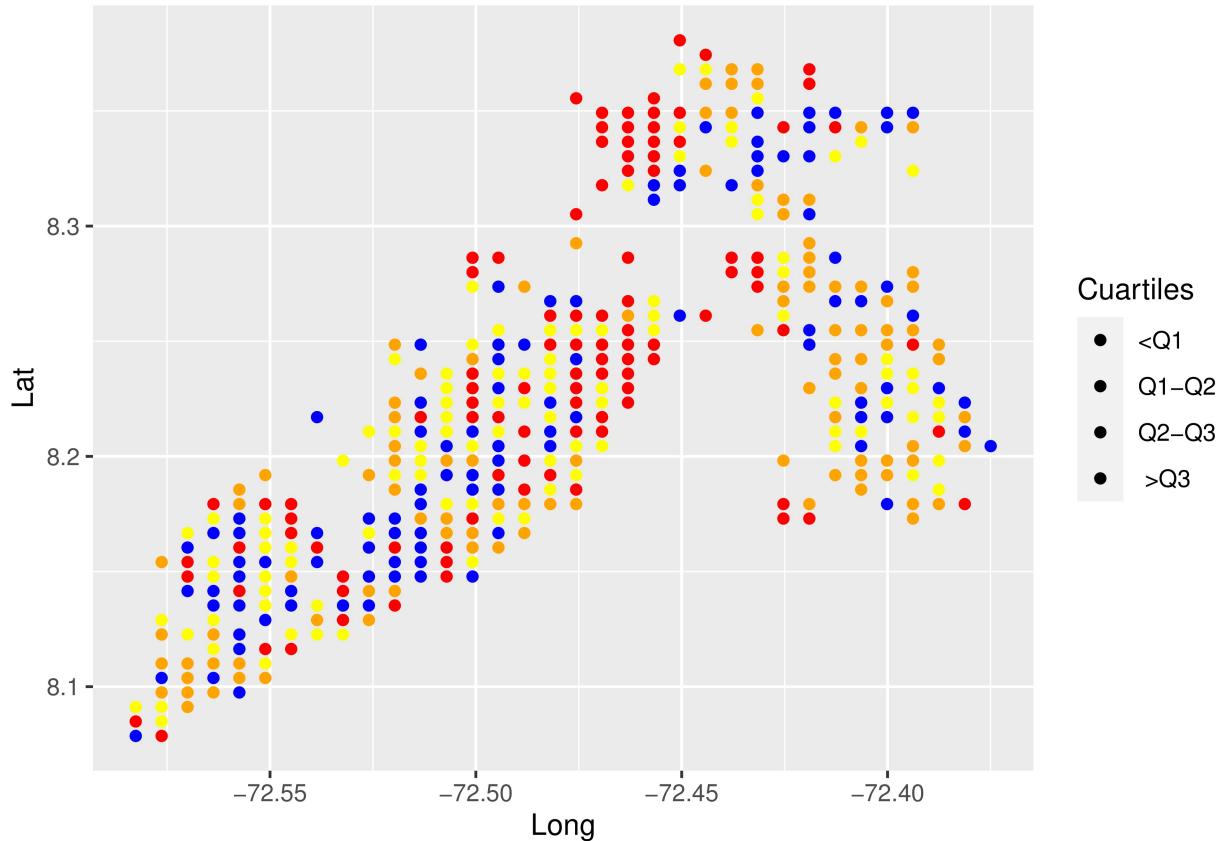
```

```

colors <- ifelse(Ca_K %in% data1, 'red',
                  ifelse(Ca_K %in% data2, 'yellow',
                        ifelse(Ca_K %in% data3, 'blue','orange')))

ggplot(XPABLO, aes(Long, Lat, fill = colors ))+
  geom_point(color = colors)+ 
  scale_fill_discrete(name = 'Cuartiles', labels = c('<Q1', 'Q1-Q2', 'Q2-Q3', '>Q3'))

```



```

# Relación Ca_Mg

Ca_Mg = XPABLO$Ca/XPABLO$Mg
nuevos_grupos <- quantile(x = Ca_Mg, probs = c(0.25, 0.5, 0.75))
nuevos_grupos

##      25%      50%      75%
## 3.206522 4.313106 5.130552

q_1 = nuevos_grupos[1]; q_1

##      25%
## 3.206522

q_2 = nuevos_grupos[2]; q_2

##      50%
## 4.313106

q_3 = nuevos_grupos[3]; q_3

##      75%

```

```

## 5.130552

data1 <- subset(Ca_K, Ca_K < q_1) #;data1
data2 <- subset(Ca_K, Ca_K < q_2 & Ca_K > q_1) #;data2
data3 <- subset(Ca_K, Ca_K < q_3 & Ca_K > q_2) #;data3
data4 <- subset(Ca_K, Ca_K > q_3) #;data4

colors <- ifelse(Ca_K %in% data1, 'red',
                  ifelse(Ca_K %in% data2, 'yellow',
                         ifelse(Ca_K %in% data3, 'blue','orange')))

ggplot(XPABLO, aes(Long, Lat, fill = colors ))+
  geom_point(color = colors)+ 
  scale_fill_discrete(name = 'Cuartiles', labels = c('<Q1', 'Q1-Q2', 'Q2-Q3', ' >Q3'))

```

