

Clusters

Construcción de clusters con las observaciones de 41 estaciones en CA

Existen muchas opciones para agrupar los caudales de las estaciones, así que se elige trabajar con 3:

- Distancia univariada de las variancias de cada estación.
- k-means usando los primeros componentes de PCA.
- Cluster para las series de tiempo utilizando <https://www.jstatsoft.org/article/view/v062i01/v62i01.pdf>

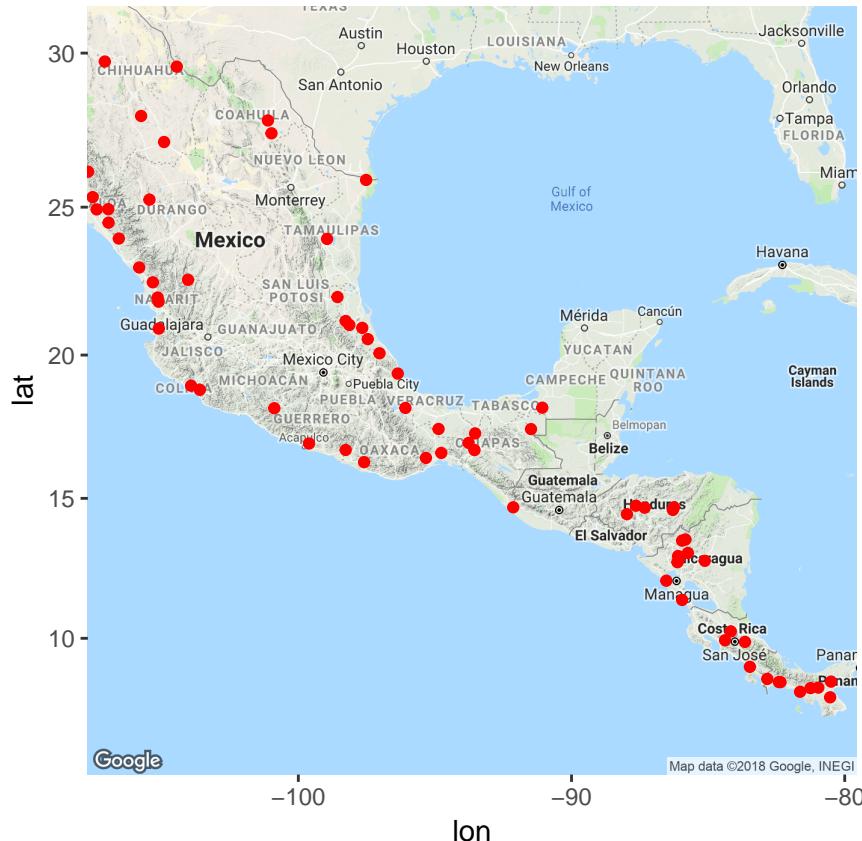
Los datos

Los datos consisten en observaciones mensuales de cuadal en 74 estaciones: las fechas de observación van desde enero de 1969 hasta diciembre de 1979. Cada estación representa una cuenca hidrológica en Centro América, y en total se tienen observaciones de 12 meses en cada uno de los 11 años para 74 locaciones. También, se tienen datos de climatología mensual para cada una de las estaciones.

```
## [1] 132 74  
## [1] 12 74
```

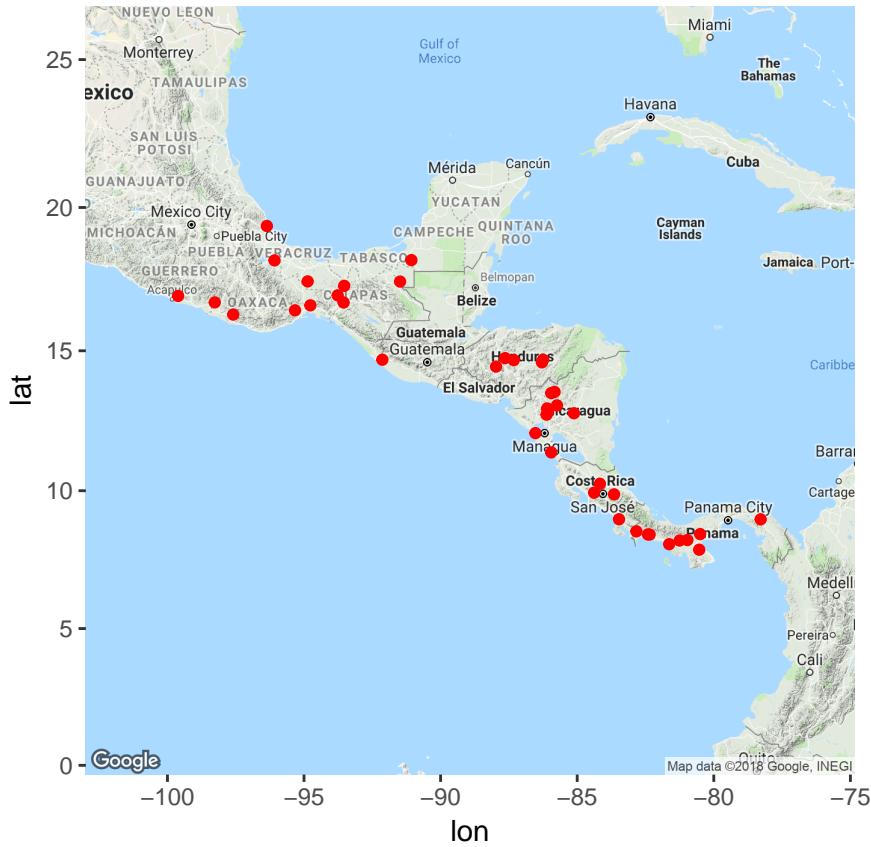
Las locaciones de las estaciones se pueden apreciar en el siguiente mapa:

```
## [1] 74 4  
##      left    bottom     right      top  
## -112.1223   5.6804 -75.2067  31.9196  
## converting bounding box to center/zoom specification. (experimental)  
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=18.8,-93.6645&zoom=5&size=640x640
```



Como las locaciones incluyen varias estaciones en el Norte de México, se realiza un corte en la latitud 20 y en la longitud -100, para obtener 41 estaciones localizadas en Centroamérica:

```
## [1] 41  4
##      left    bottom     right     top
## -101.7405   6.7175 -76.1505 20.5115
## converting bounding box to center/zoom specification. (experimental)
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=13.6145,-88.9455&zoom=5&size=640
```



Clustering

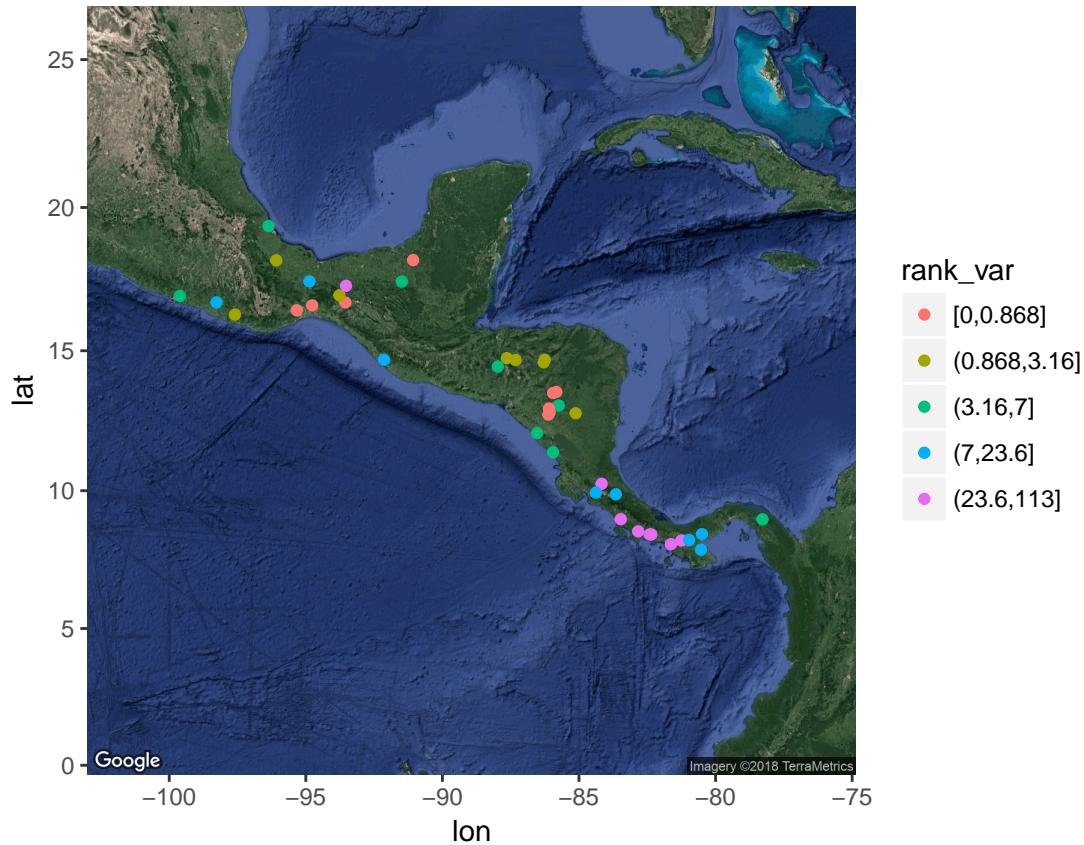
- OPCIÓN 1: Utilizar las variancias y luego agrupar por magnitud.

```

caudal2 <- caudal[,-c(1:3)][,estaciones]
aa<-round(c(apply(caudal2,2,var)),4)
loca$rank_var <- cut_number(aa,5)
sq_map <- get_map(location = sbbox, maptype = "satellite", source = "google")

## converting bounding box to center/zoom specification. (experimental)
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=13.6145,-88.9455&zoom=5&size=640
ggmap(sq_map) + geom_point(data = loca, mapping = aes(x = Longitud, y = Latitud, colour=rank_var))

```



Descripción de los clusters:

```
medianas <- apply(caudal2,2,median)
data <- as_tibble(cbind(loca,medianas))
names(data) <- c("cod","lat","lon","area","rank_var","Mediana_de_cluster")
## Area promedio de cada cluster:
as.integer(tapply(data$area, data$rank_var,mean))

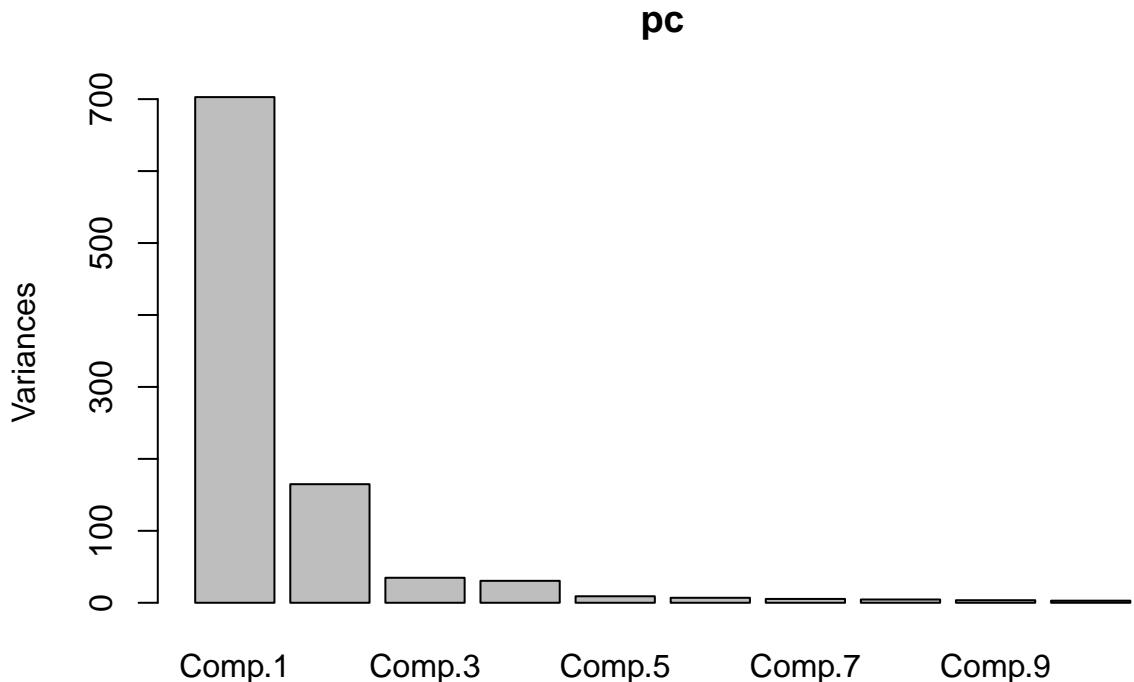
## [1]      5452      6800      7414      2336 150501132
## Mediana del Caudal de cada cluster:
round(tapply(data$Mediana_de_cluster, data$rank_var,median),4)

##      [0,0.868] (0.868,3.16]      (3.16,7]      (7,23.6]  (23.6,113]
##      0.2100      0.6966      1.2184      2.9613      5.8753
```

- OPCIÓN 2: k-means usando los componentes de PCA.

Primero, se deben calcular los PCA utilizando las anomalías en lugar de las observaciones. Como se tiene la climatología mensual para cada locación, el cálculo consiste en restar la climatología mensual a cada observación, según el mes correspondiente. Luego, se procede a hacer el PCA y por último agrupar las estaciones utilizando k-means de los primeros 10 componentes.

```
clima2 <- (as.matrix(clima[,-1]) %x% rep(1, 11))[,estaciones]
anomalies <- caudal2-clima2
pc <- princomp(anomalies)
plot(pc)
```



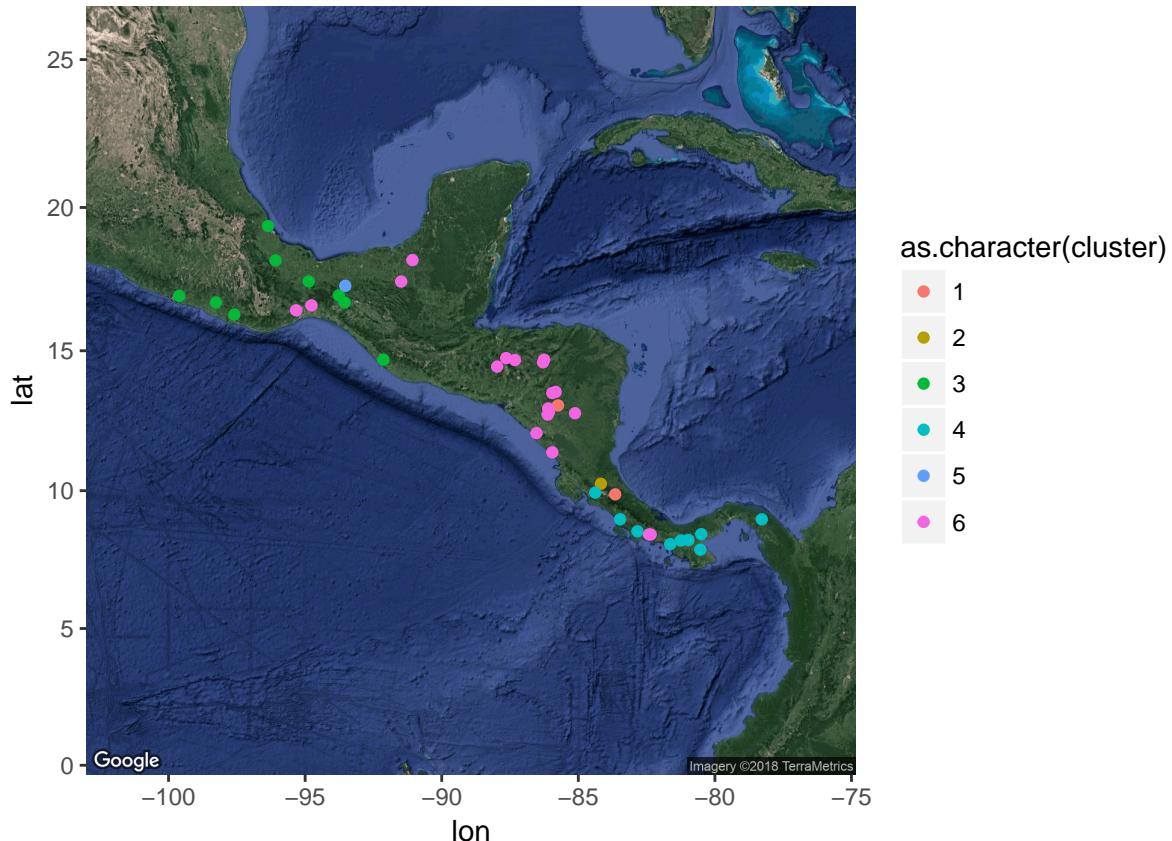
```

mydata <- (pc$loadings[,1:10])
fit <- kmeans(mydata, 6)
aggregate(mydata, by=list(fit$cluster), FUN=mean)

##   Group.1      Comp.1      Comp.2      Comp.3      Comp.4      Comp.5
## 1     1 -0.11233800  0.04250103 -0.187330958 -0.11083914  0.11319933
## 2     2 -0.24592266  0.91456306 -0.166646054  0.07449124  0.03767719
## 3     3 -0.09424171 -0.07769546 -0.216389282  0.09284698  0.01519469
## 4     4 -0.20821281 -0.02161293  0.104754798 -0.04423074 -0.10179408
## 5     5 -0.21288492 -0.02967169  0.373404996  0.76764079  0.37206089
## 6     6 -0.07587092 -0.03482275 -0.004815917 -0.03130364  0.08769432
##           Comp.6      Comp.7      Comp.8      Comp.9      Comp.10
## 1  0.09858380 -0.288782108  0.39100539  0.257065721 -0.004188229
## 2 -0.11972971  0.009819237 -0.09304008 -0.003632703 -0.002704492
## 3 -0.01929671  0.099209355 -0.02670685 -0.047062298 -0.009543630
## 4 -0.07479228  0.040555410  0.12280041 -0.108345507  0.016535947
## 5  0.22721034 -0.086379727  0.08896825  0.013361527 -0.013674519
## 6 -0.07470252 -0.065546645 -0.07059747  0.014633171  0.056269114

loca <- data.frame(loca, cluster=fit$cluster)
ggmap(sq_map) + geom_point(data = loca, mapping = aes(x = Longitud, y = Latitud, colour=as.character(clu

```



Descripción de los clusters:

```
data <- as_tibble(cbind(loca,medianas))
names(data) <- c("cod","lat","lon","area","rank_var","cluster2","Mediana_de_cluster")
## Area promedio de cada cluster:
round(as.integer(tapply(data$area,data$cluster2,mean)),0)

## [1] 818 73 7302 1406 200 63374224

## Medianan del Caudal de cada cluster:
round(tapply(data$Mediana_de_cluster, data$cluster2,median),4)

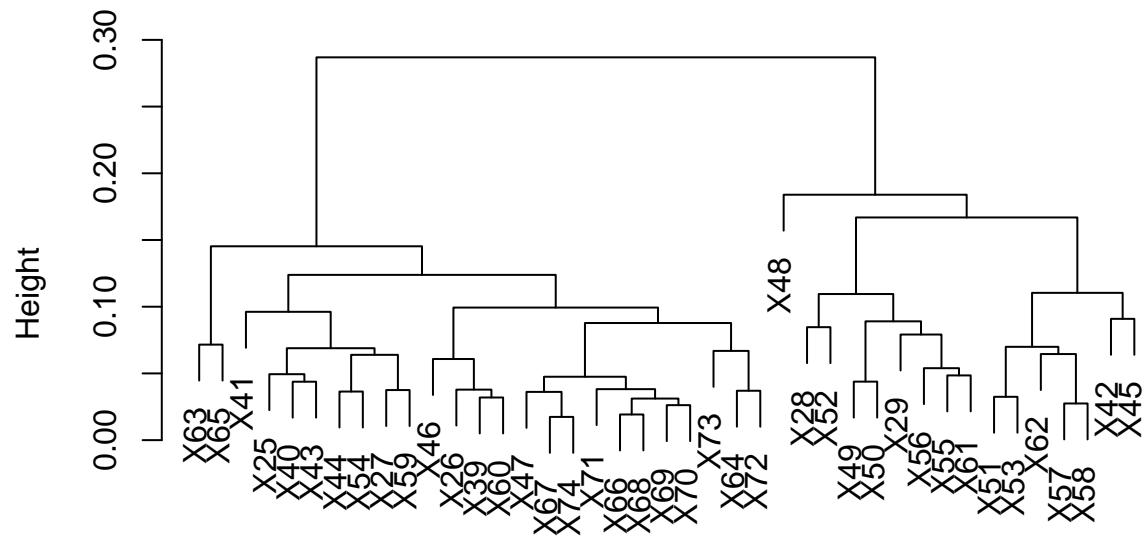
## 1 2 3 4 5 6
## 3.9604 9.4685 1.2092 4.1007 4.7279 0.5400
```

- OPCIÓN 3: Cluster para las series de tiempo.

En este caso también se deben calcular las anomalías. Luego, se procede a aplicar el algoritmo de TSclust, que se describe aquí: <https://www.jstatsoft.org/article/view/v062i01/v62i01.pdf>

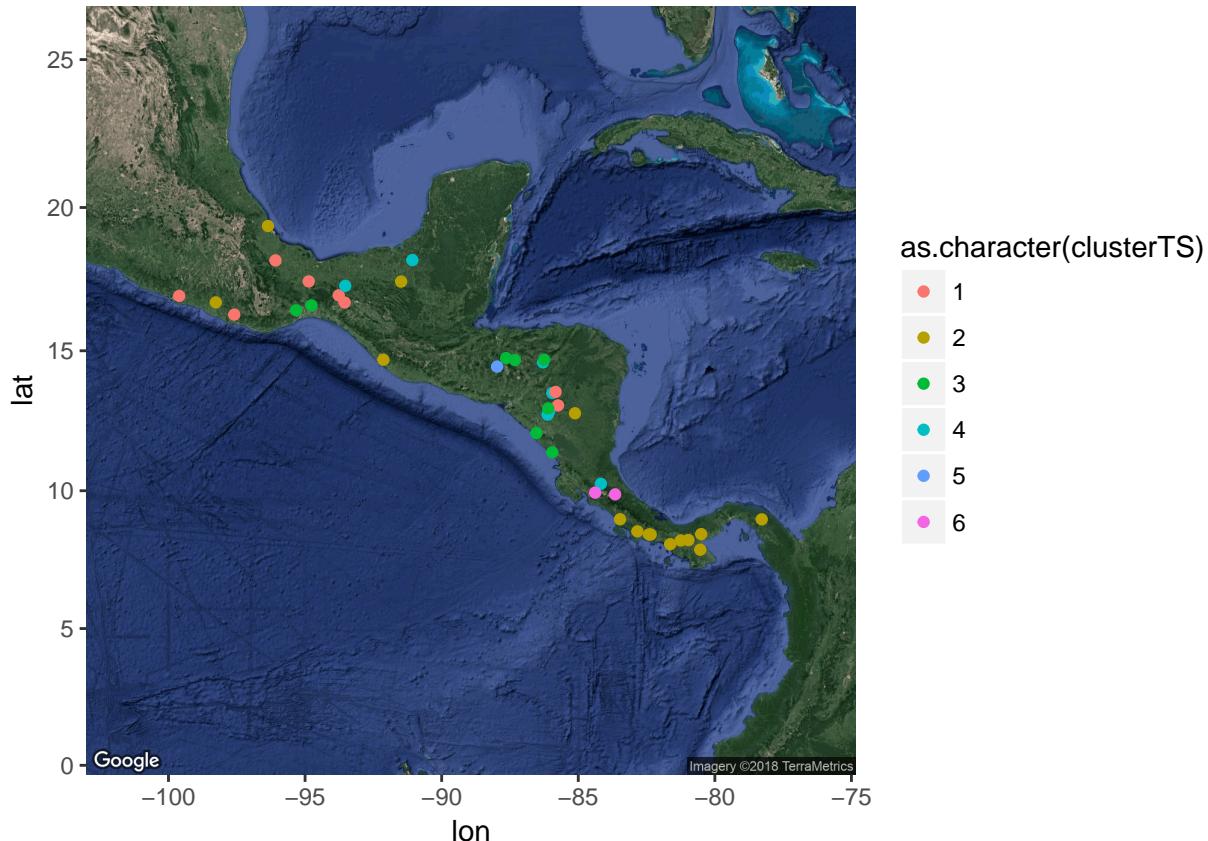
```
dpred <- diss(anomalies, "ACF", p=0.05)
hc.pred <- hclust(dpred)
plot(hc.pred)
```

Cluster Dendrogram



```
dpred  
hclust (*, "complete")
```

```
aa<-cutree(hc.pred, k = 6)  
loca <- data.frame(loca, clusterTS=aa )  
ggmap(sq_map) + geom_point(data = loca, mapping = aes(x = Longitud, y = Latitud, colour=as.character(clu
```



Descripción de los clusters:

```
data <- as_tibble(cbind(loca,medianas))
names(data) <- c("cod","lat","lon","area","rank_var","cluster2","clusterTS", "Mediana_de_cluster")
## Area promedio de cada cluster:
as.integer(tapply(data$area, data$clusterTS,mean))

## [1] 8372 80271495 3563 2686 836 1487
## Mediana del Caudal de cada cluster:
round(tapply(data$Mediana_de_cluster, data$cluster2,median),4)

## 1 2 3 4 5 6
## 3.9604 9.4685 1.2092 4.1007 4.7279 0.5400
```