Aprendizaje Automático I

Ejercicio. Aprendizaje No Supervisado

Isaac Martín de Diego

Nov. 2023

En este ejercicio vamos a trabajar con los datos del *"Dow Jones Index Data Set"* que podéis descargar en el siguiente enlace: DOW JONES INDEX. Se trata de datos semanales del Dow Jone Industrial Index.

Datos

En primer lugar descargamos los datos y leemos los datos

```
djidata = read.table("./dow_jones_index/dow_jones_index.data",
    header = TRUE, sep = ",")
djidata = as.data.frame(djidata)
head(djidata)
```

```
##   quarter stock      date   open   high    low  close    volume
## 1       1    AA  1/7/2011 $15.82 $16.72 $15.78 $16.42 239655616
## 2       1    AA 1/14/2011 $16.71 $16.71 $15.64 $15.97 242963398
## 3       1    AA 1/21/2011 $16.19 $16.38 $15.60 $15.79 138428495
## 4       1    AA 1/28/2011 $15.87 $16.63 $15.82 $16.13 151379173
## 5       1    AA  2/4/2011 $16.18 $17.39 $16.18 $17.14 154387761
## 6       1    AA 2/11/2011 $17.33 $17.48 $16.97 $17.37 114691279
##   percent_change_price percent_change_volume_over_last_wk previous_weeks_volume
## 1             3.792670                                NA                    NA
## 2            -4.428490                          1.380223             239655616
## 3            -2.470660                        -43.024959             242963398
## 4             1.638310                          9.355500             138428495
```

```
## 5             5.933250                          1.987452            151379173
## 6             0.230814                        -25.712195            154387761
##   next_weeks_open next_weeks_close percent_change_next_weeks_price
## 1          $16.71           $15.97                        -4.428490
## 2          $16.19           $15.79                        -2.470660
## 3          $15.87           $16.13                         1.638310
## 4          $16.18           $17.14                         5.933250
## 5          $17.33           $17.37                         0.230814
## 6          $17.39           $17.28                        -0.632547
##   days_to_next_dividend percent_return_next_dividend
## 1                    26                     0.182704
## 2                    19                     0.187852
## 3                    12                     0.189994
## 4                     5                     0.185989
## 5                    97                     0.175029
## 6                    90                     0.172712
```

```
table(djidata$stock)
```

```
##
##   AA  AXP   BA  BAC  CAT CSCO  CVX   DD  DIS   GE   HD  HPQ  IBM INTC  JNJ  JPM
##   25   25   25   25   25   25   25   25   25   25   25   25   25   25   25   25
##   KO KRFT  MCD  MMM  MRK MSFT  PFE   PG    T  TRV  UTX   VZ  WMT  XOM
##   25   25   25   25   25   25   25   25   25   25   25   25   25   25
```

Cada fila corresponde a datos semanales de un valor bursatil. En este ejercicio vamos a trabajar con los datos correspondientes a la variable *close*, esto es, el valor al cierre de la semana del stock.

Necesitamos transformar la variable de interés como sigue:

```
djidata$close = as.numeric(sub("\\$", "", djidata$close))
```

1. Construir la matriz de series temporales

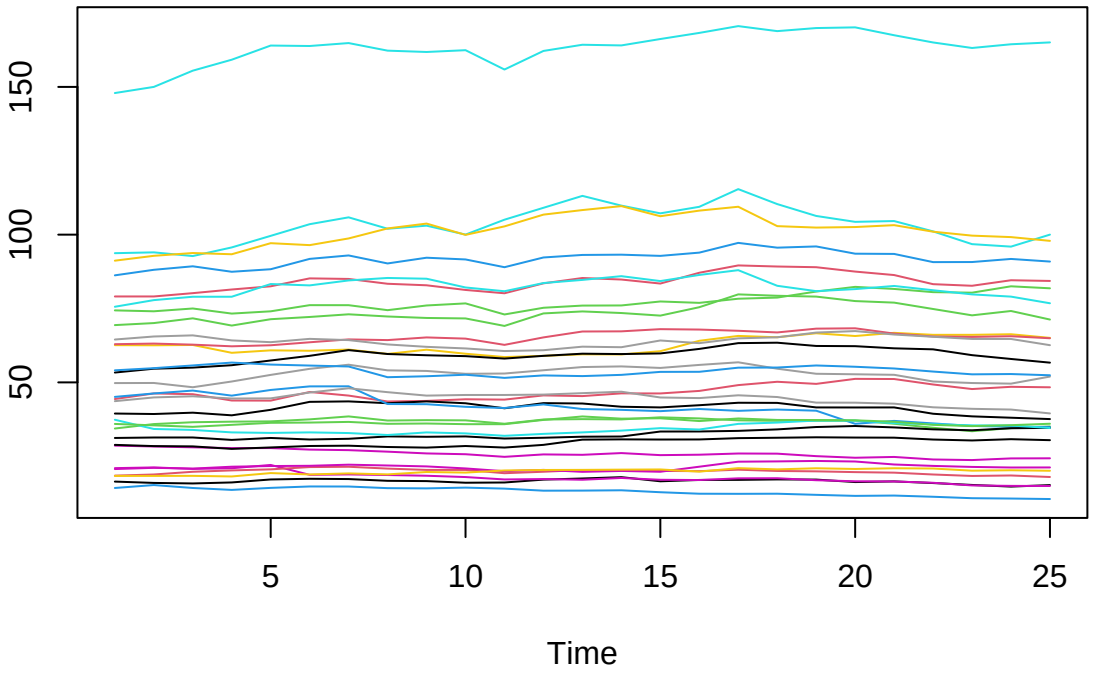
En primer lugar hemos de construir la matriz con las series que necesitamos. Necesitamos una matriz de series con las series por columnas para cada uno de los valores bursátiles.

```
stocks = unique(djidata[, "stock"])
n = dim(djidata[stocks == "AA", ])[1]
stocksdata = matrix(0, n, length(stocks))
for (i in 1:length(stocks)) stocksdata[, i] = djidata[djidata$stock ==
    stocks[i], "close"]
colnames(stocksdata) = stocks
```

```
stocksts1 = as.ts(stocksdata[1:12, ])
stocksts2 = as.ts(stocksdata[13:25, ])
stocksts = as.ts(stocksdata)
```

2. Representar las series con las que vamos a trabajar

```
ts.plot(stocksts, col = seq(1:25))
```



A la vista de este gráfico, podríamos realizar nuestro análisis basándono, únicamente, en dos caractarísticas de las series: su media y desviación típica.

3. Realizar un análisis cluster usando como variables de interés la media y desviación estándar de cada serie

```
require(dplyr)
```

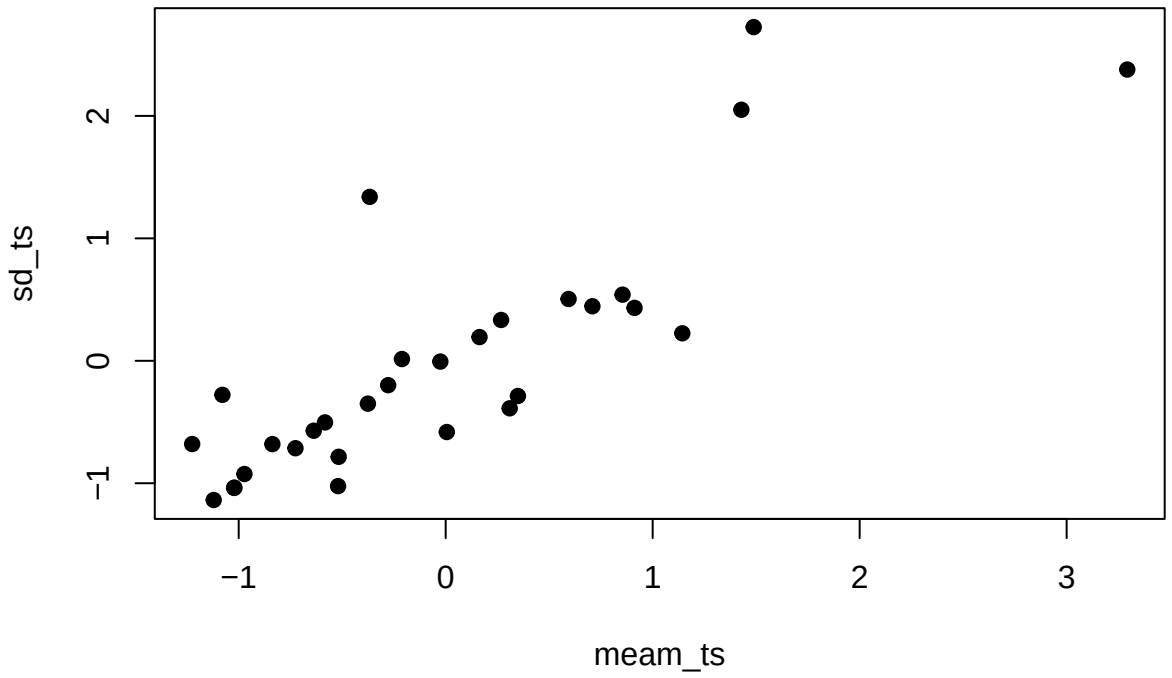
```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

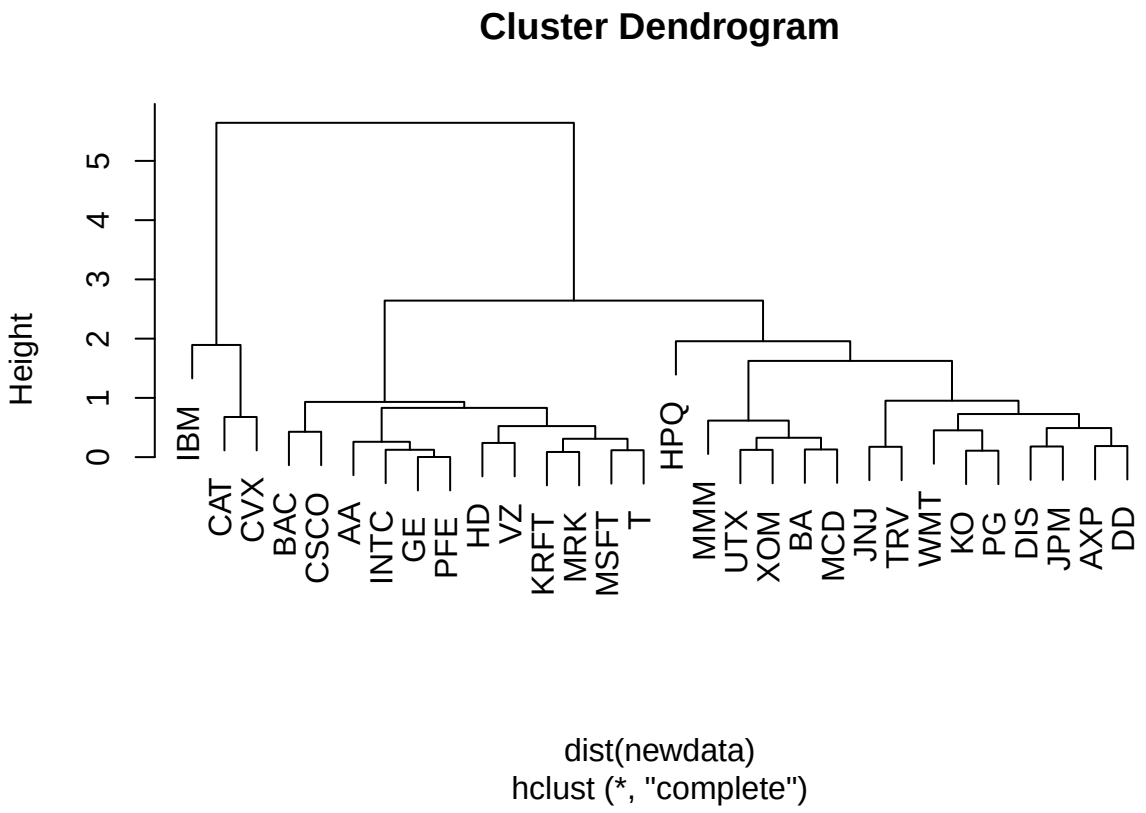
```
newdata = djidata %>%
    group_by(stock) %>%
    summarise(meam_ts = mean(close), sd_ts = sd(close))
newdata = as.data.frame(newdata)
row.names(newdata) = newdata[, 1]
newdata = scale(newdata[, -1])
plot(newdata, pch = 19)
```



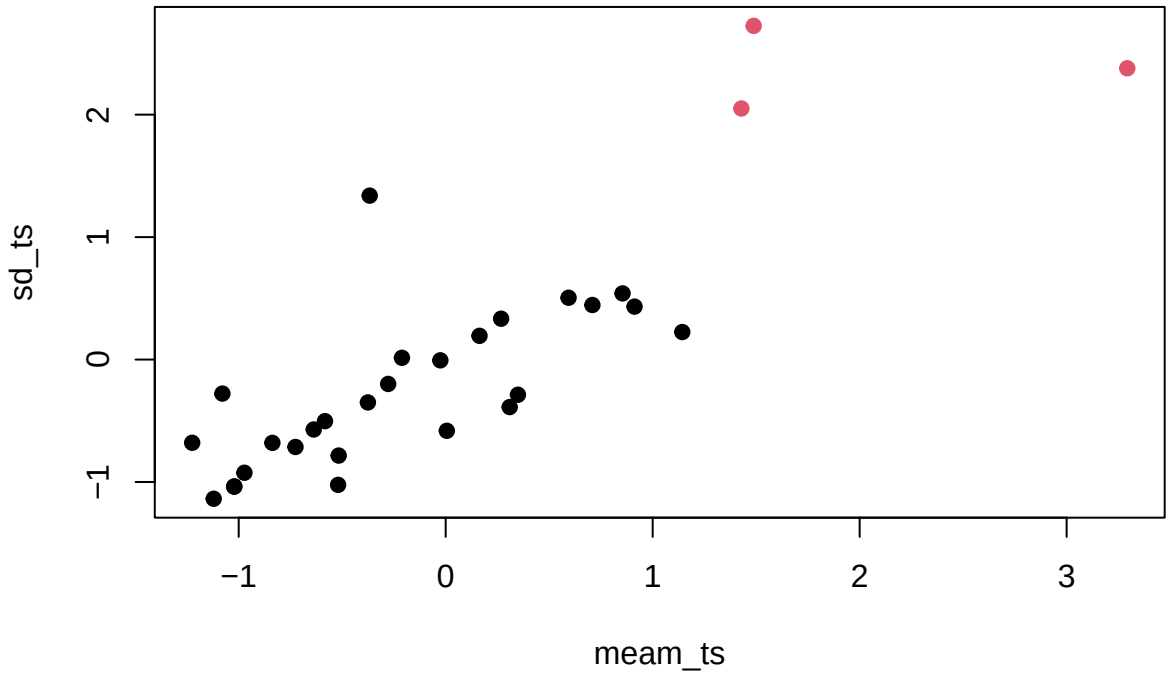

```
require(cluster)
```

```
## Loading required package: cluster
```

```
djicluster = hclust(dist(newdata))
plot(djicluster)
```




```r
djicluster2 = cutree(djicluster, k = 2)
plot(newdata, pch = 19, col = djicluster2)
```



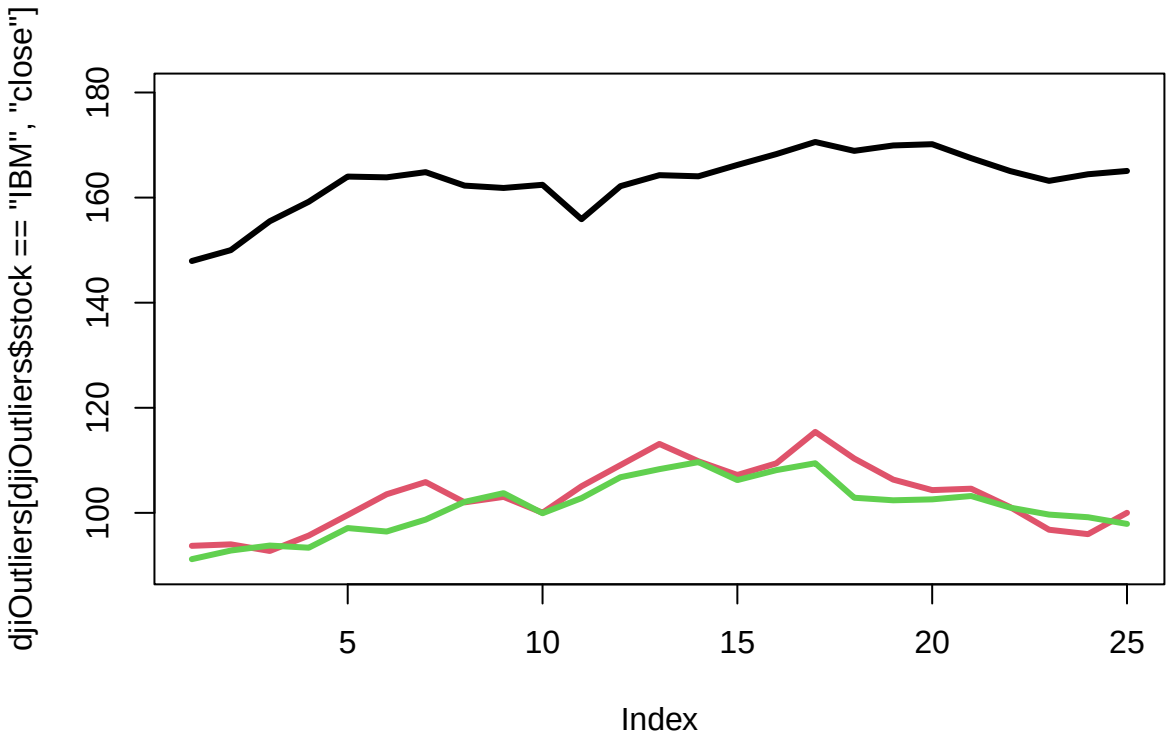
```
djicluster2
```

```
##    AA  AXP   BA  BAC  CAT CSCO  CVX   DD  DIS   GE   HD  HPQ  IBM INTC  JNJ  JPM
##     1    1    1    1    2    1    2    1    1    1    1    1    2    1    1    1
##    KO KRFT  MCD  MMM  MRK MSFT  PFE   PG    T  TRV  UTX   VZ  WMT  XOM
##     1    1    1    1    1    1    1    1    1    1    1    1    1    1
```

¿Pueden identificarse valores atípicos?

Todos los atipicos aparecen en el grupo 2.

```
cselect = c("CAT", "CVX", "IBM")
djiOutliers = djidata %>%
    filter(stock == "CAT" | stock == "CVX" | stock == "IBM")
plot(djiOutliers[djiOutliers$stock == "IBM", "close"], type = "l",
    lwd = 3, ylim = c(90, 180))
points(djiOutliers[djiOutliers$stock == "CAT", "close"], type = "l",
    lwd = 3, col = 2)
points(djiOutliers[djiOutliers$stock == "CVX", "close"], type = "l",
    lwd = 3, col = 3)
```



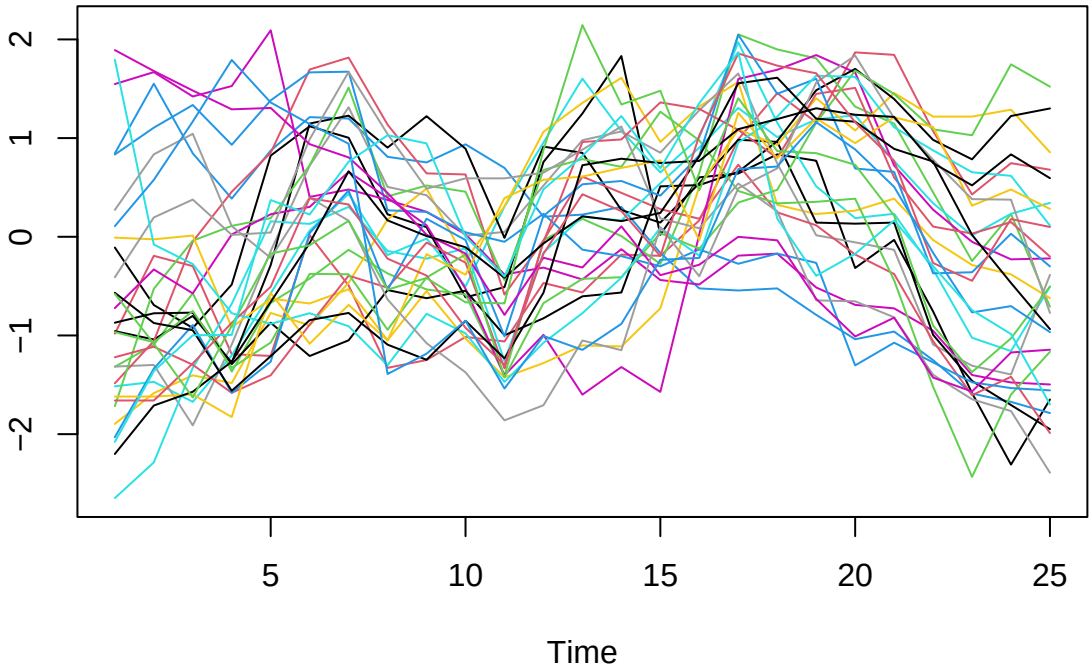
¿Existe relacion entre las dos variables consideradas en el analisis? ¿Como interpretas este resultado?

Efectivamente, existe una relacion lineal positiva entre media y desviacion tipica. Por lo

tanto, lo mas logico seria escalar los datos para que todos tengan la misma media y desviacion tipica.

4. Representar las series escaladas

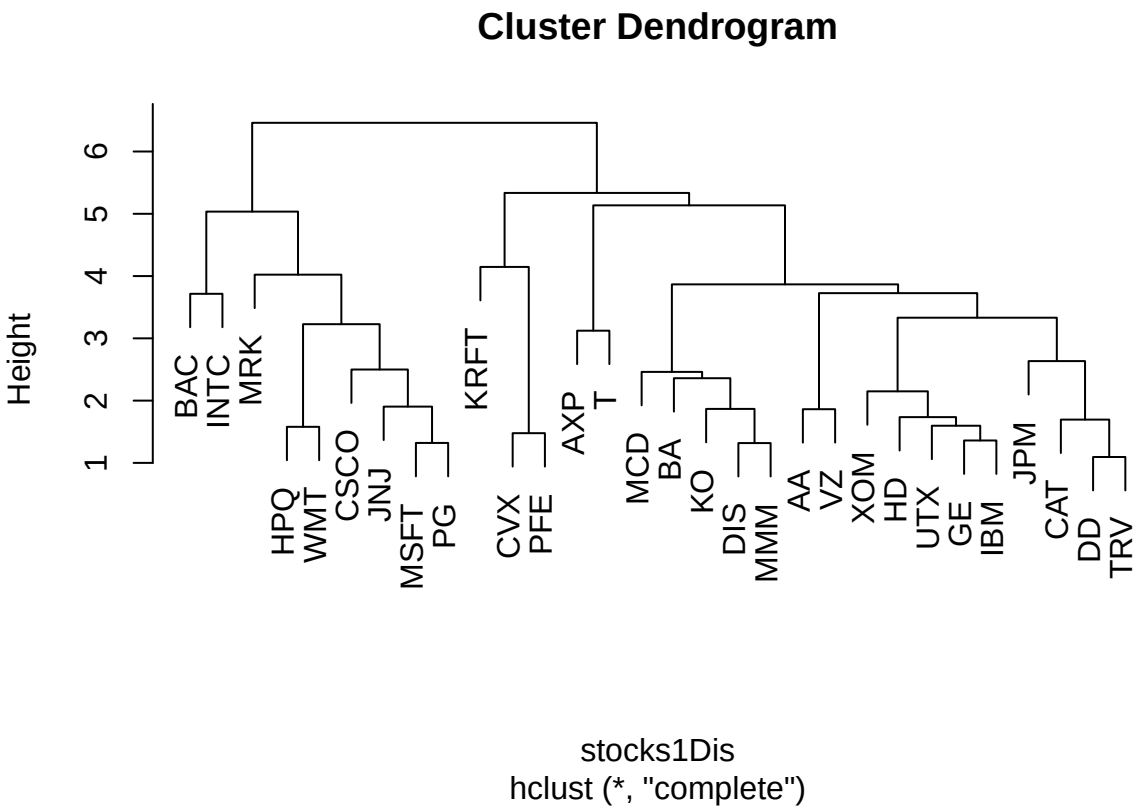
```
ts.plot(scale(stocksts), col = seq(1:25))
```



Una vez contruida la matriz con las series temporales, podemos pasar a analizar los datos. Separaremos los datos por cuatrimestres (*quarter*). Realizaremos un análisis cluster para cada uno de los cuatrimestres y otro empleando todo el periodo.

5. Análisis Cluster para cada uno de los cuatrimestres

```
# Usamos la distancia euclídea y un método jerárquico.
stocksts1Scaled = scale(stocksts1)
stocks1Dis = dist(t(stocksts1Scaled))
cluster1 = hclust(stocks1Dis)
plot(cluster1)
```



¿En cuantos grupos podemos dividir la muestra?

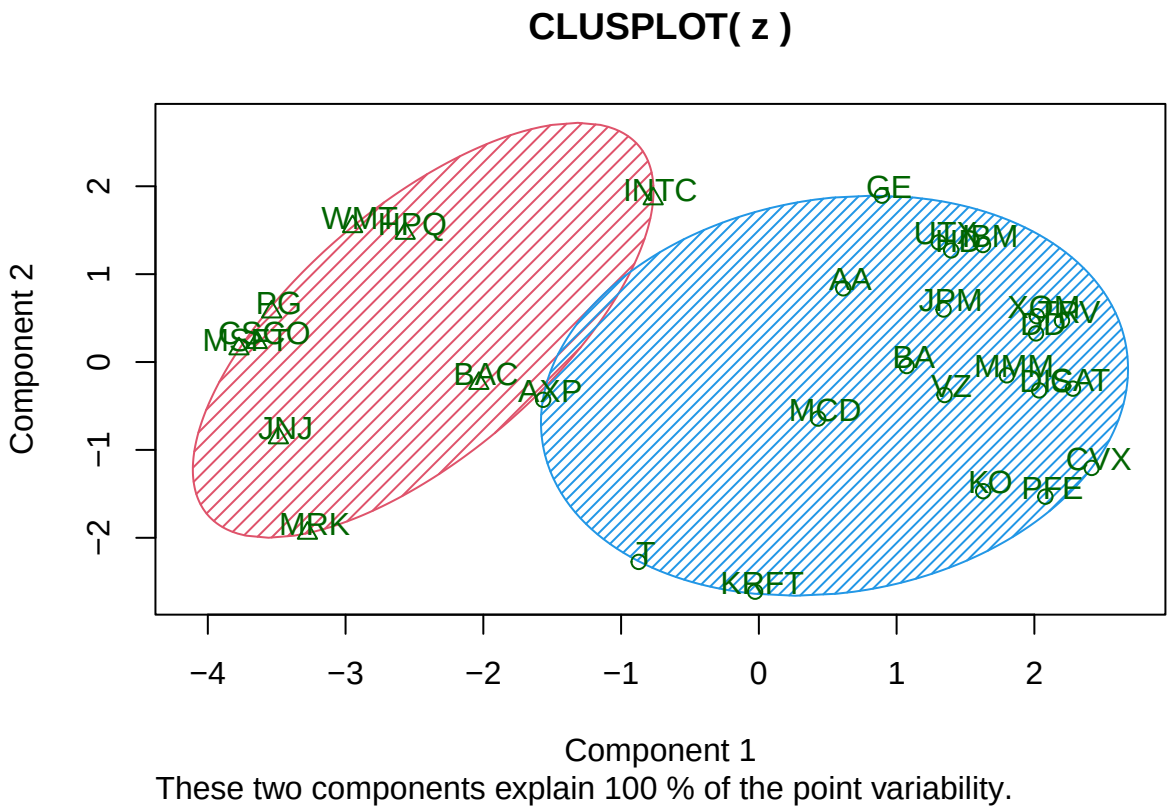
A la vista del dendograma intuimos 2 grupos

```
# z3=pam(t(stocksts1Scaled),2)
# z1=kmeans(t(stocksts1Scaled),2,nstart=25)
z1 = cutree(cluster1, 2)
require(useful)
```

```
## Loading required package: useful
```

```
## Loading required package: ggplot2
```

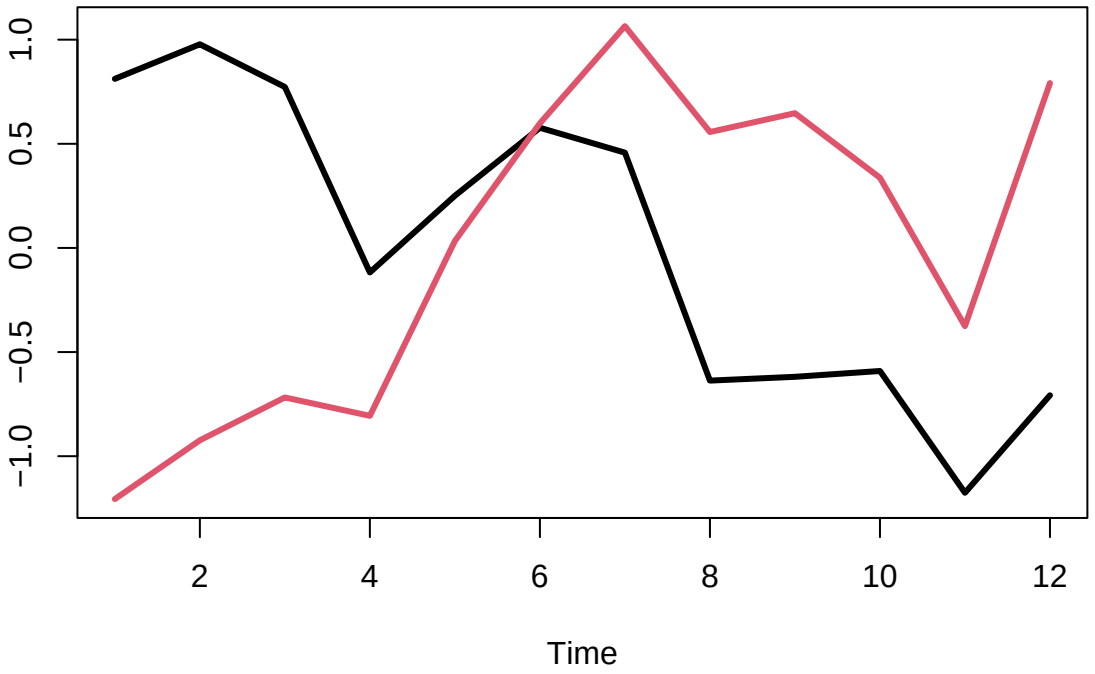
```
z = cmdscale(stocks1Dis)
clusplot(z, labels = 3, clus = z1, shade = TRUE, color = TRUE)
```



Representar graficamente la media de cada cluster para tratar de identificar el comportamiento medio de los valores en cada cluster.

Buscamos una representación media del comportamiento en cada cluster.

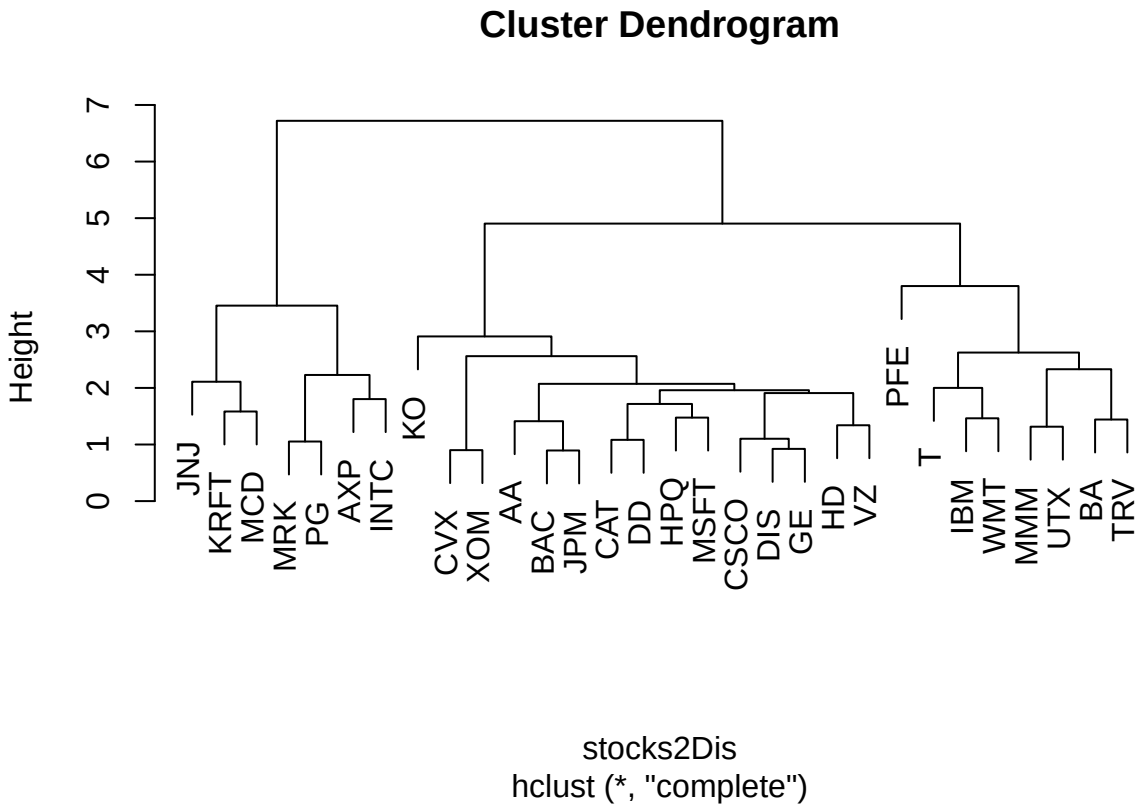
```
z1 = kmeans(t(stocksts1Scaled), 2, nstart = 25)
ts.plot(t(z1$centers), col = 1:2, lwd = 3)
```



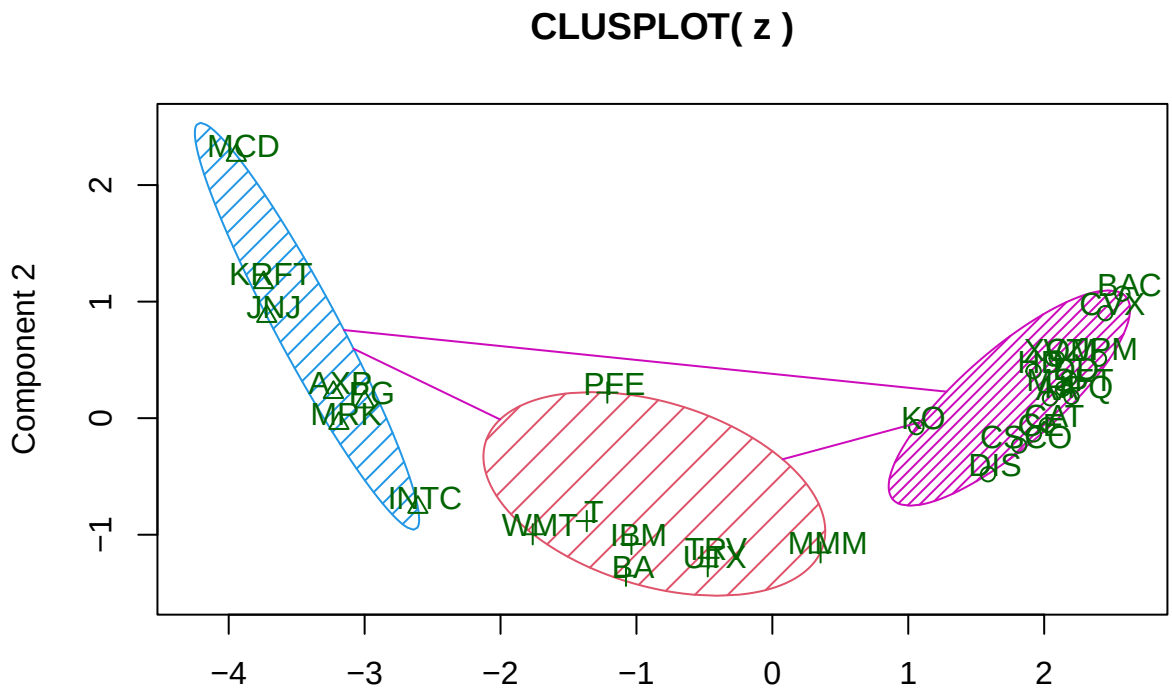
Podemos observar que el primer cluster corresponde a valores que crecen con el tiempo y el segundo cluster a valores que decrecen con el tiempo.

Pasamos a trabajar con el segundo cuatrimestre.

```
stocksts2Scaled = scale(stocksts2)
stocks2Dis = dist(t(stocksts2Scaled))
cluster2 = hclust(stocks2Dis)
plot(cluster2)
```

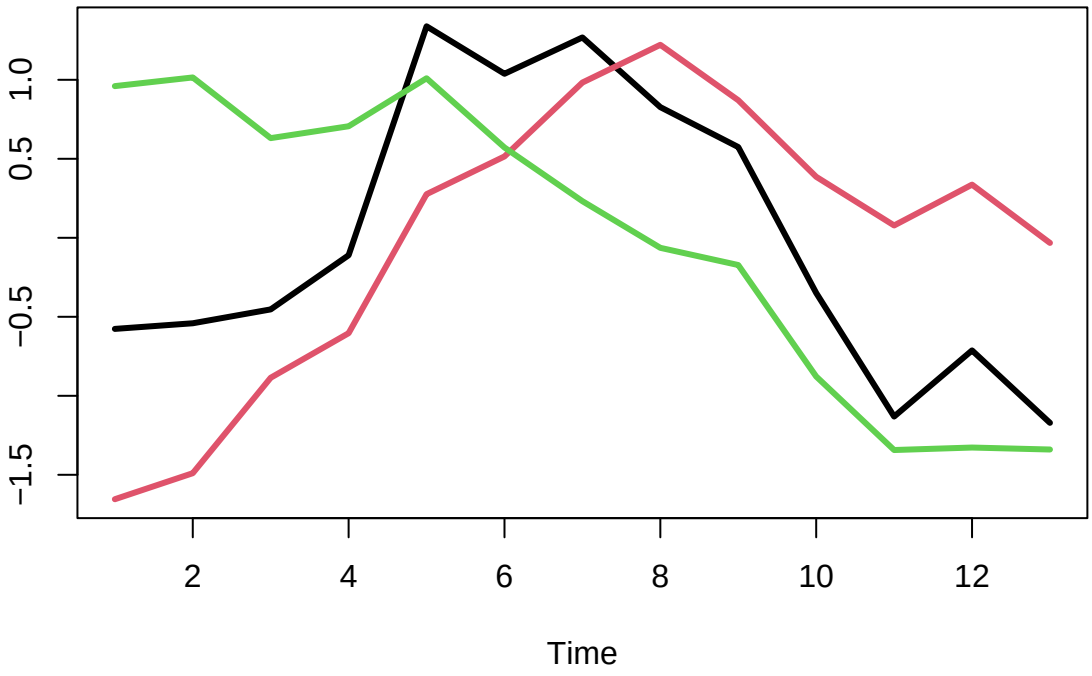



```r
# A la vista del dendograma intuimos 3 grupos
z2 = cutree(cluster2, 3)
# z2=kmeans(t(stockstsScaled),3,nstart=25)
require(useful)
z = cmdscale(dist(t(stockstsScaled)))
clusplot(z, labels = 3, clus = z2, shade = TRUE, color = TRUE)
```



Buscamos una representación media del comportamiento en cada cluster.

```
z2 = kmeans(t(stocksts2Scaled), 3, nstart = 25)
ts.plot(t(z2$centers), col = 1:3, lwd = 3)
```



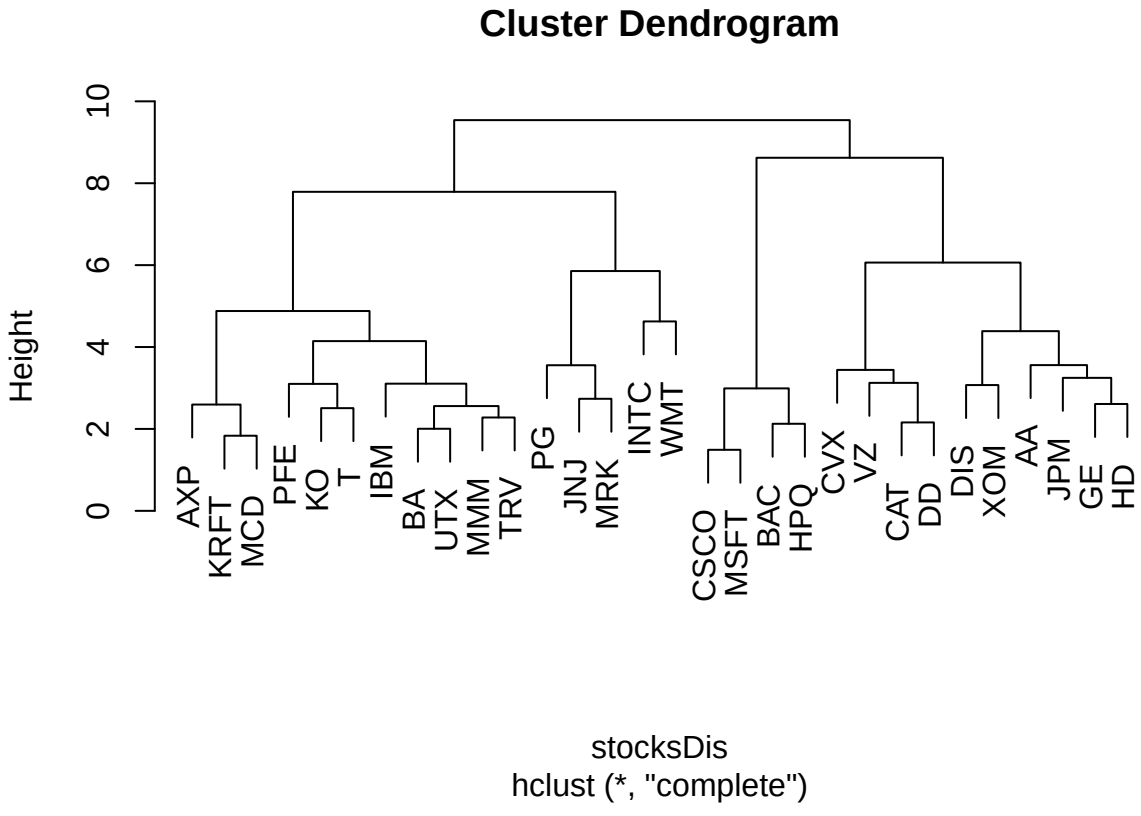
Podemos observar que el primer cluster corresponde a valores que decrecen con el tiempo. El

segundo cluster corresponde a valores que crecen, se mantienen más o menos constantes y luego decrecen. El tercer cluster corresponde a valores que crecen y luego decrecen.

6. Analisis Cluster para todo el periodo

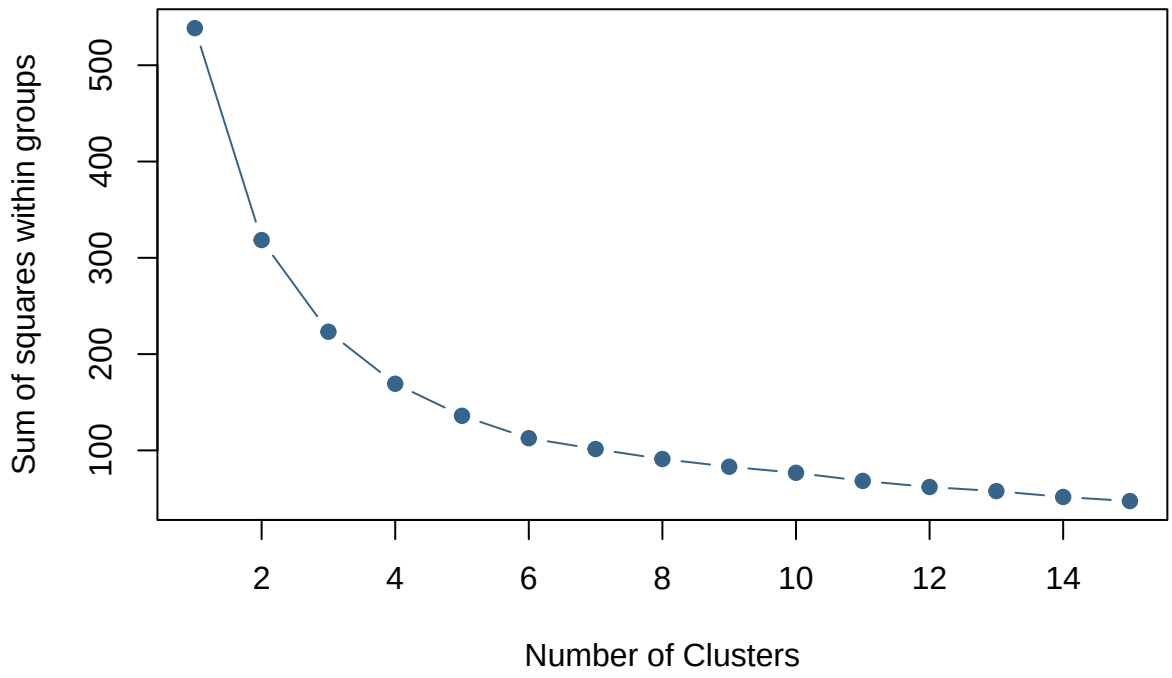
Trabajamos con los datos de todo el periodo analizado.

```
# todo el periodo
stockstsScaled = scale(stocksts)
stocksDis = dist(t(stockstsScaled))
clusterTotal = hclust(stocksDis)
plot(clusterTotal)
```

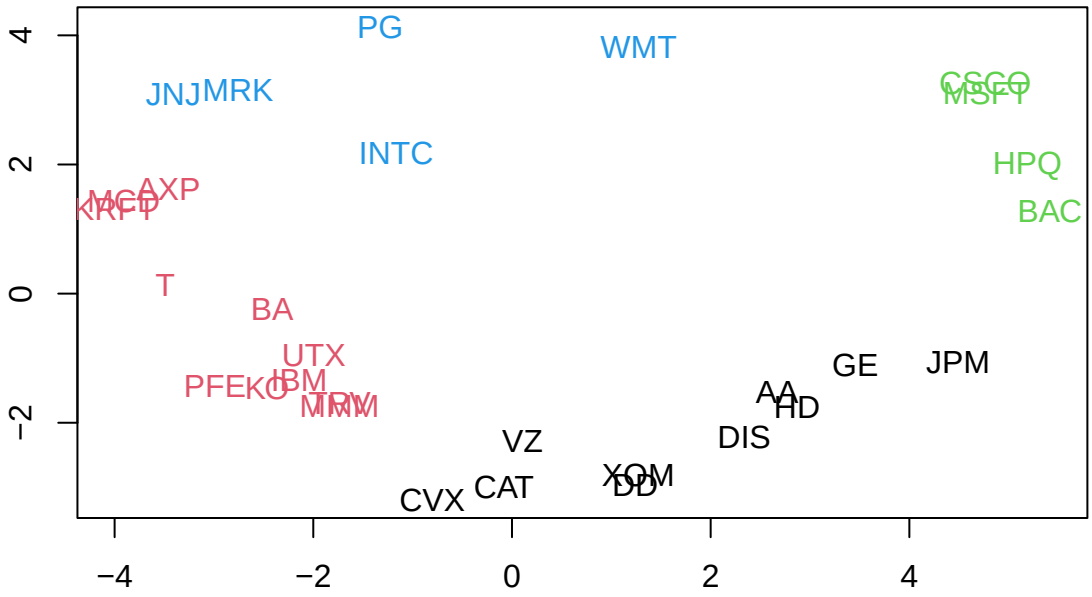


Elegir una tecnica para determinar el mejor numero de clusters.

```
SSW <- vector(mode = "numeric", length = 15)
SSW[1] <- (30 - 1) * sum(apply(t(stockstsScaled), 2, var))
for (i in 2:15) SSW[i] <- sum(kmeans(t(stockstsScaled), centers = i,
    nstart = 25)$withinss)
plot(1:15, SSW, type = "b", xlab = "Number of Clusters", ylab = "Sum of squares within g
    pch = 19, col = "steelblue4")
```

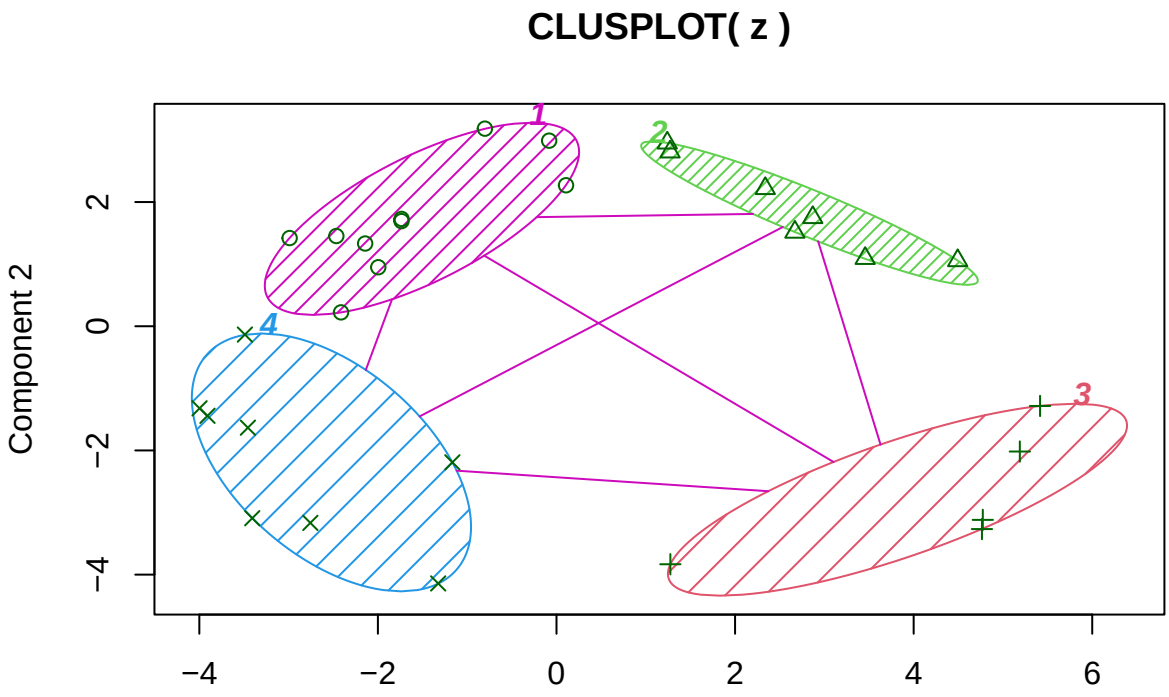



```r
# A la vista del dendograma intuimos 4 grupos
z3 = cutree(clusterTotal, 4)
require(useful)
z = cmdscale(dist(t(stockstsScaled)))
plot(z[, 1], z[, 2], type = "n", xlab = "", ylab = "")
text(z[, 1], z[, 2], rownames(z), cex = 1, col = z3)
```

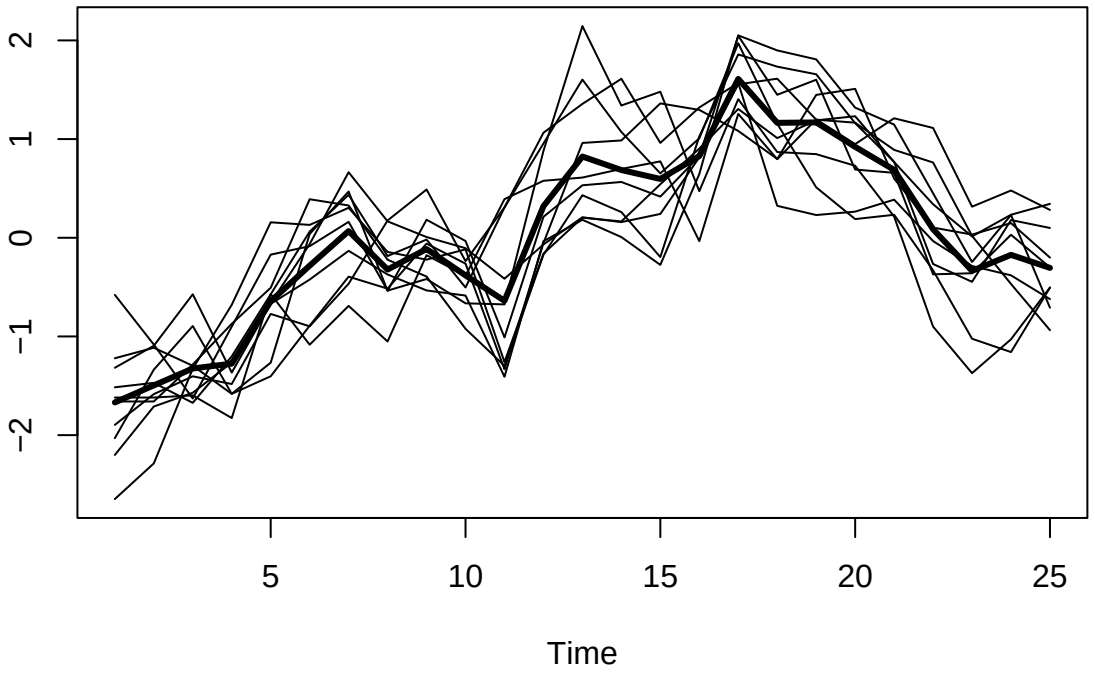


7. Representar graficamente la media de cada cluster Vamos a ver,las medias de cada uno de los clusters.

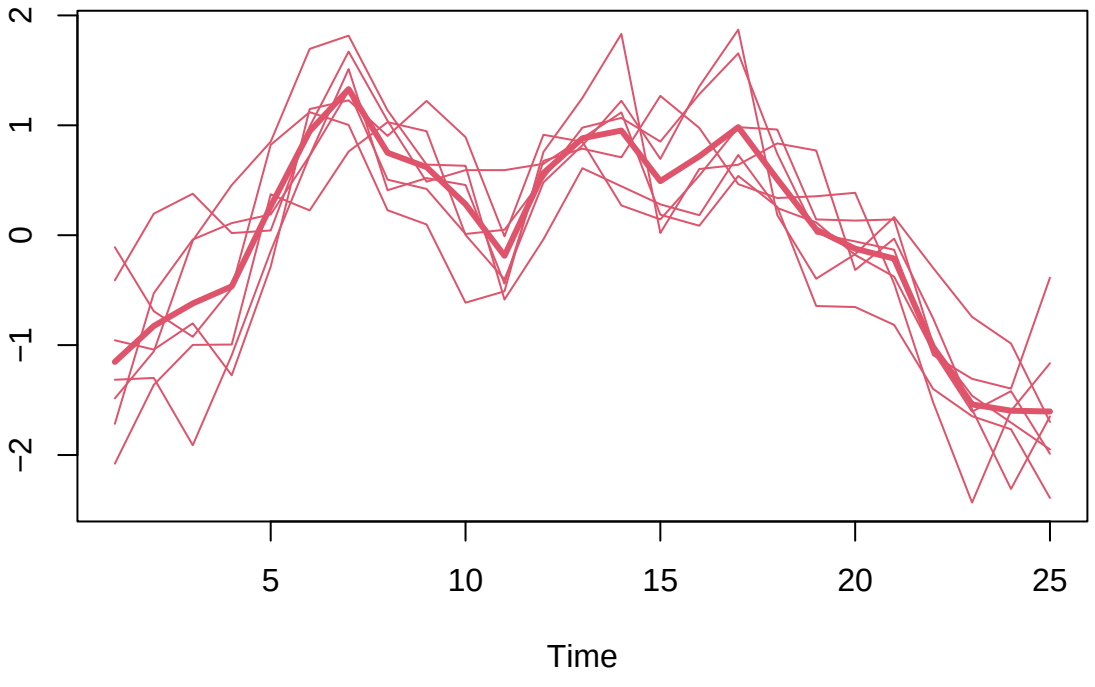
```r
# A la vista del dendograma intuimos 3 grupos
z3 = kmeans(t(stockstsScaled), 4, nstart = 25)
require(useful)
clusplot(z, labels = 4, clus = z3$cluster, shade = TRUE, color = TRUE)
```



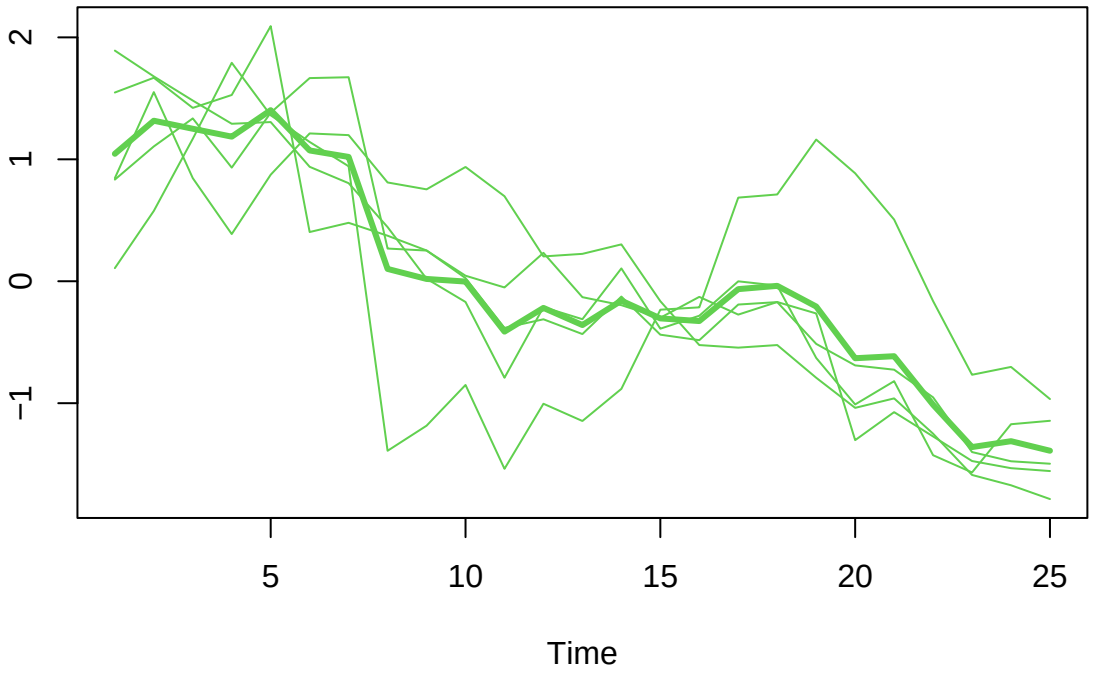

```r
ts.plot(cbind((z3$centers[1, ]), stockstsScaled[, z3$cluster ==
    1]), lwd = c(3, rep(1, sum(z3$cluster == 1))), col = 1)
```



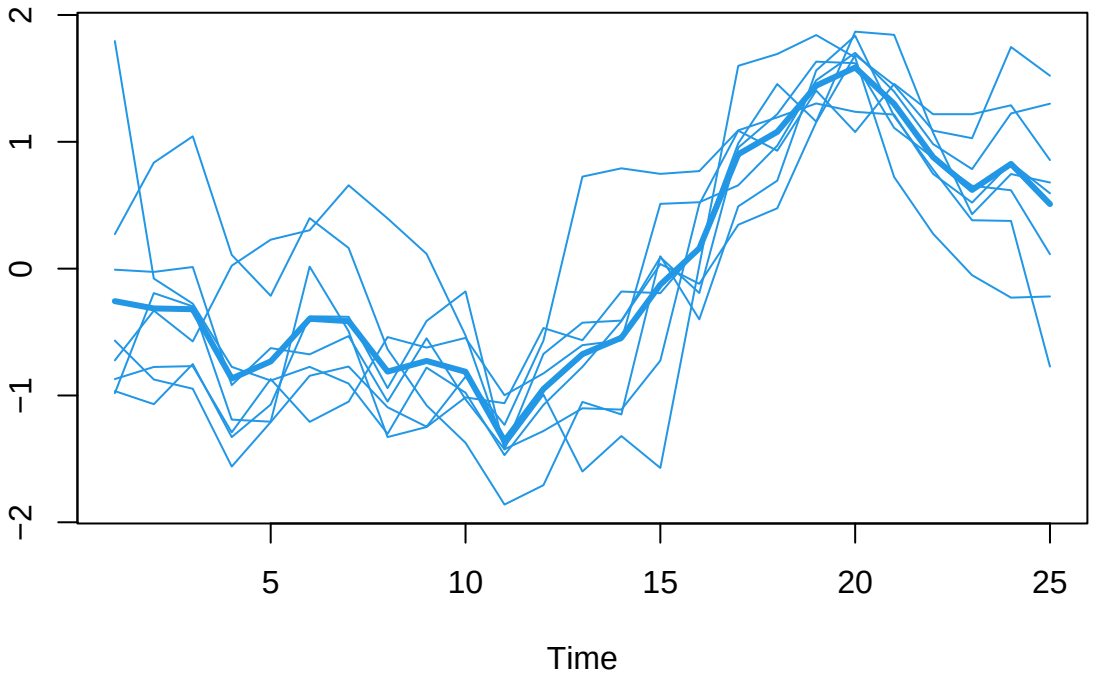

```
ts.plot(cbind((z3$centers[2, ]), stockstsScaled[, z3$cluster ==
    2]), lwd = c(3, rep(1, sum(z3$cluster == 2))), col = 2)
```




```
ts.plot(cbind((z3$centers[3, ]), stockstsScaled[, z3$cluster ==
    3]), lwd = c(3, rep(1, sum(z3$cluster == 3))), col = 3)
```



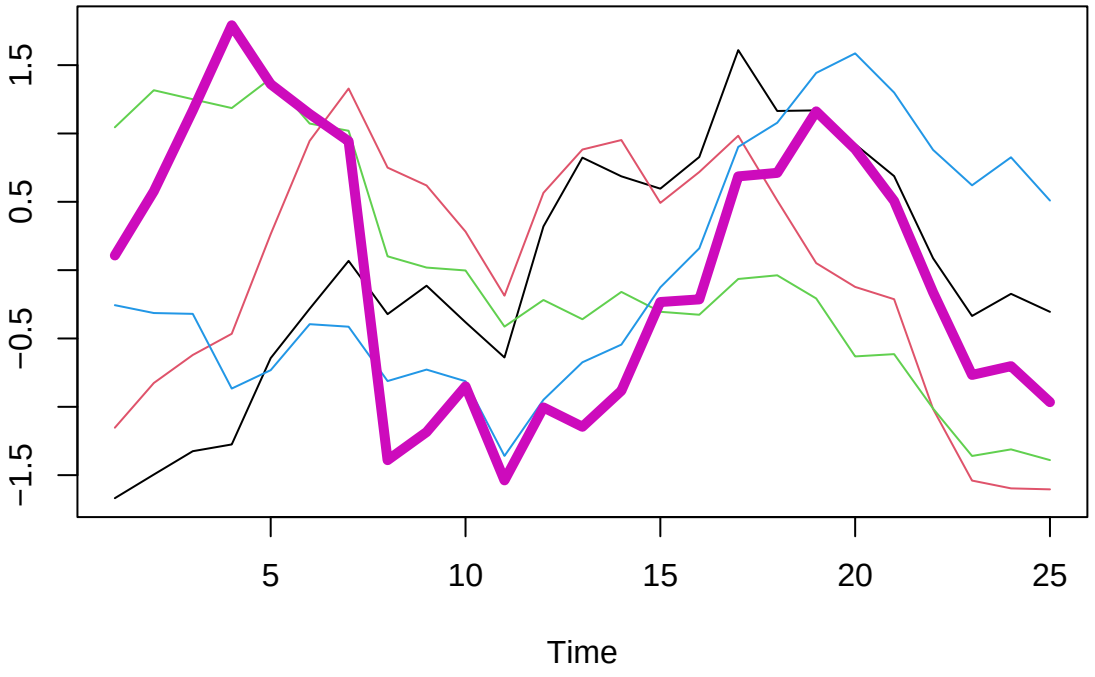

```
ts.plot(cbind((z3$centers[4, ]), stockstsScaled[, z3$cluster ==
    4]), lwd = c(3, rep(1, sum(z3$cluster == 4))), col = 4)
```



El primer cluster corresponde a valores decrecientes en el tiempo. El segundo cluster corresponde a valores que permanecen aproximadamente constantes hasta la mitad del periodo para crecer a partir de ese momento. El tercer cluster corresponde a valores crecientes en el tiempo. El cuarto cluster corresponde a valores que crecen, permanecen contantes, y decrecen.

8. Localizar atipicos en los clusters Observando detalladamente las series y los resultados obtenidos en este análisis podemos ver que la serie correspondiente a *WMT* presenta un comportamiento que no responde a ninguno de estos clusters.

```
ts.plot(cbind(t(z3$centers), stockstsScaled[, "WMT"]), col = c(1:4,
    6), lwd = c(rep(1, 4), 5))
```



---

9. Repetir el análisis, para todo el periodo, empleando la distancia DTW.

```
require(dtw)
```

```
## Loading required package: dtw
```

```
## Loading required package: proxy
```

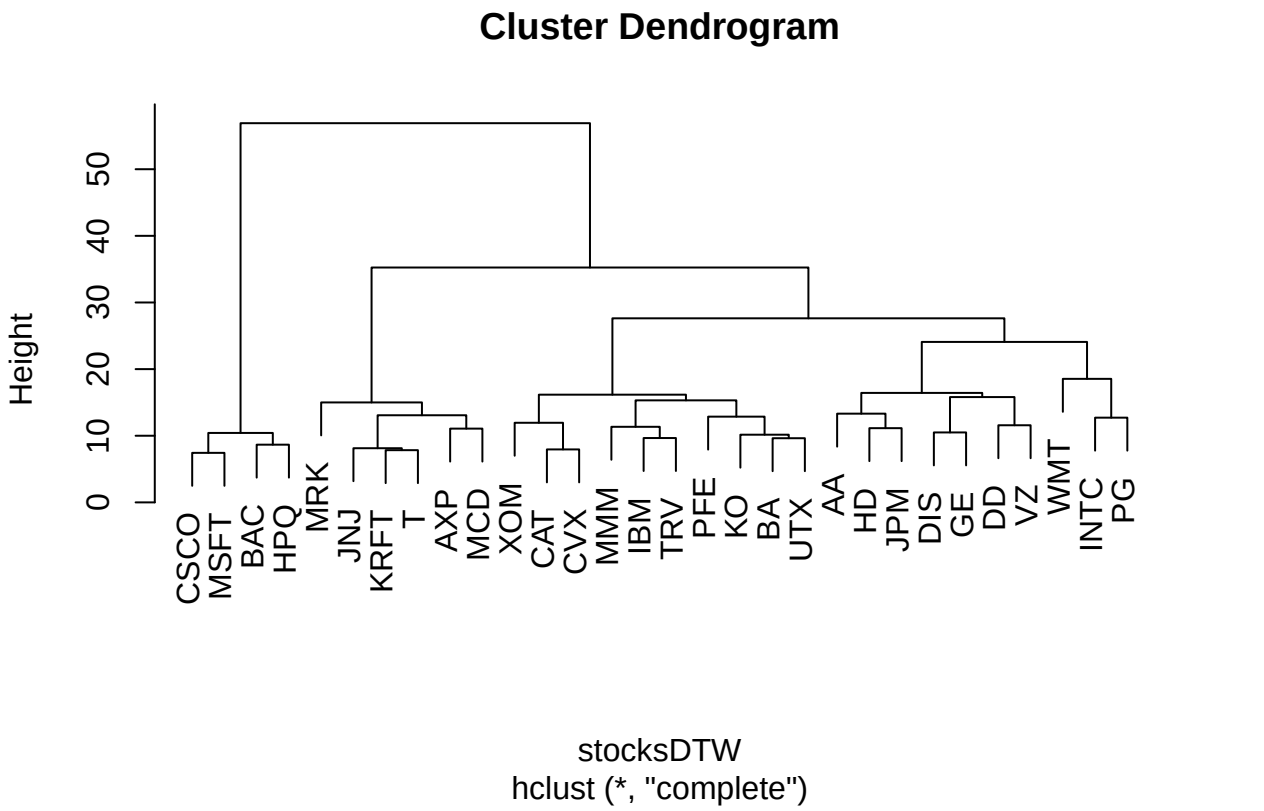
```
##
## Attaching package: 'proxy'
```

```
## The following objects are masked from 'package:stats':
##
##     as.dist, dist
```

```
## The following object is masked from 'package:base':
##
##     as.matrix
```

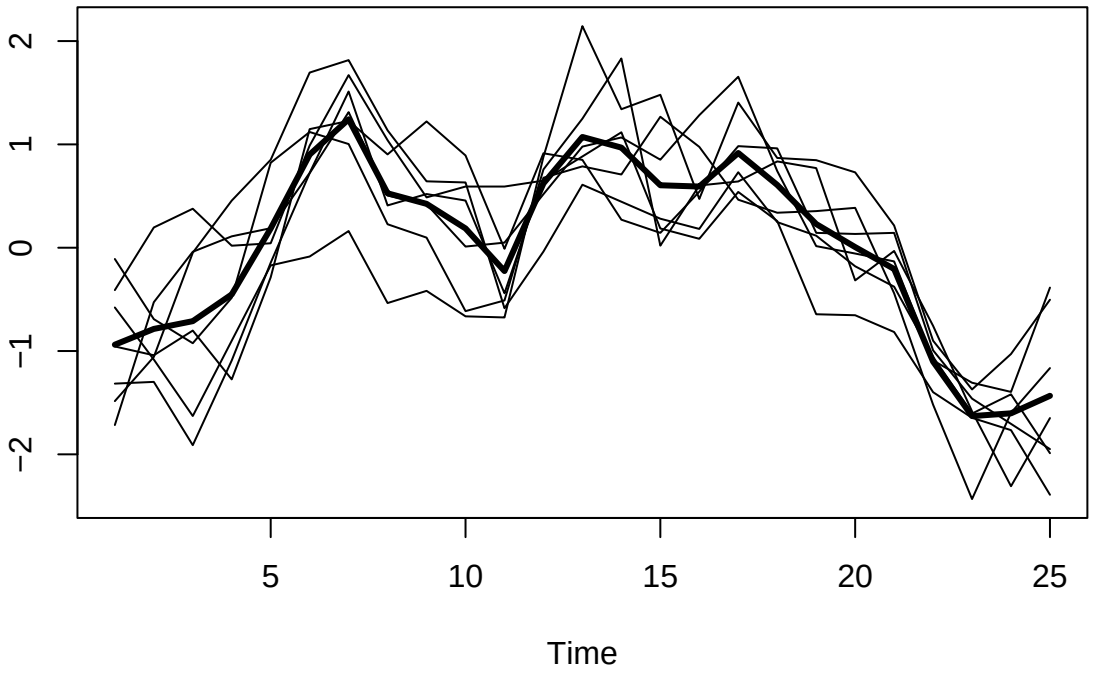
```
## Loaded dtw v1.23-1. See ?dtw for help, citation("dtw") for use in publication.
```

```
stocksDTW = dist(t(stockstsScaled), method = "DTW")
clusterTotalDTW = hclust(stocksDTW)
plot(clusterTotalDTW)
```

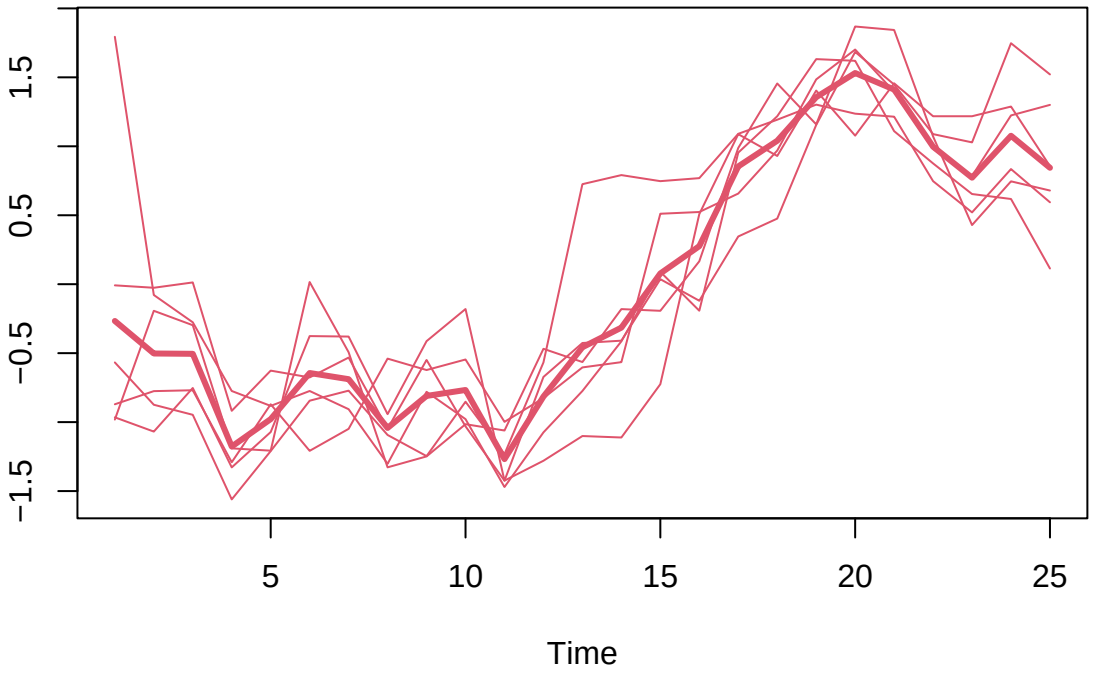


En este caso podríamos quedarnos con 3 o 5 grupos.

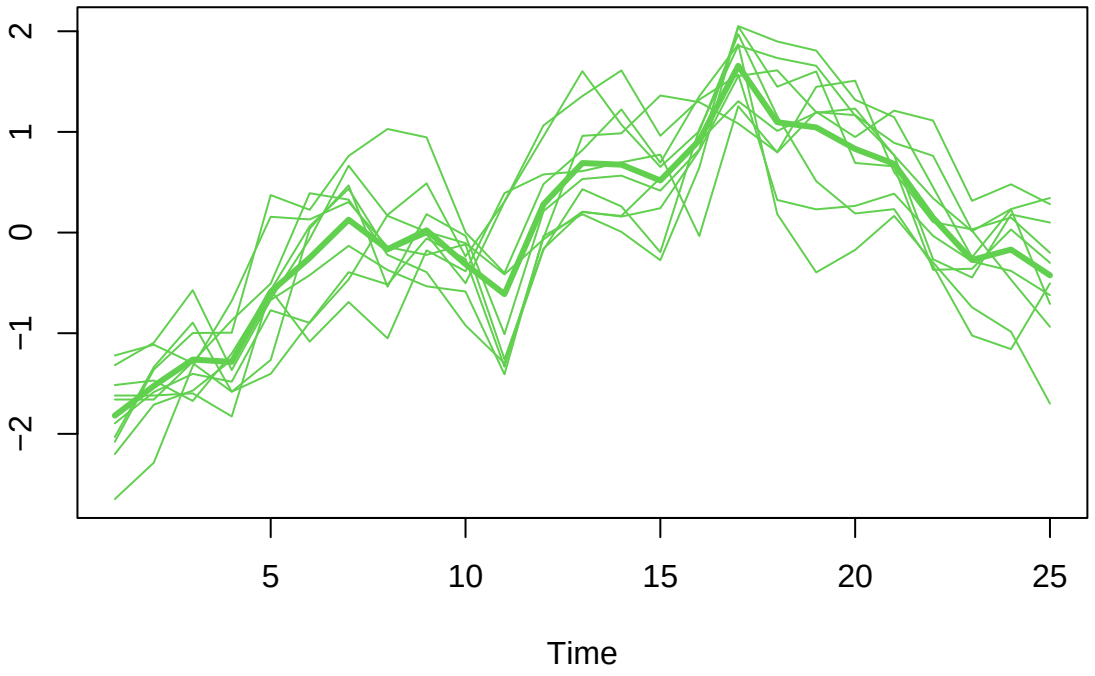
```
zDTW = cutree(clusterTotalDTW, 5)
m1 = apply(stockstsScaled[, zDTW == 1], 1, mean)
m2 = apply(stockstsScaled[, zDTW == 2], 1, mean)
m3 = apply(stockstsScaled[, zDTW == 3], 1, mean)
m4 = apply(stockstsScaled[, zDTW == 4], 1, mean)
m5 = apply(stockstsScaled[, zDTW == 5], 1, mean)
# A la vista del dendograma intuimos 3 grupos
require(useful)
ts.plot(cbind(m1, stockstsScaled[, zDTW == 1]), lwd = c(3, rep(1,
    sum(zDTW == 1))), col = 1)
```



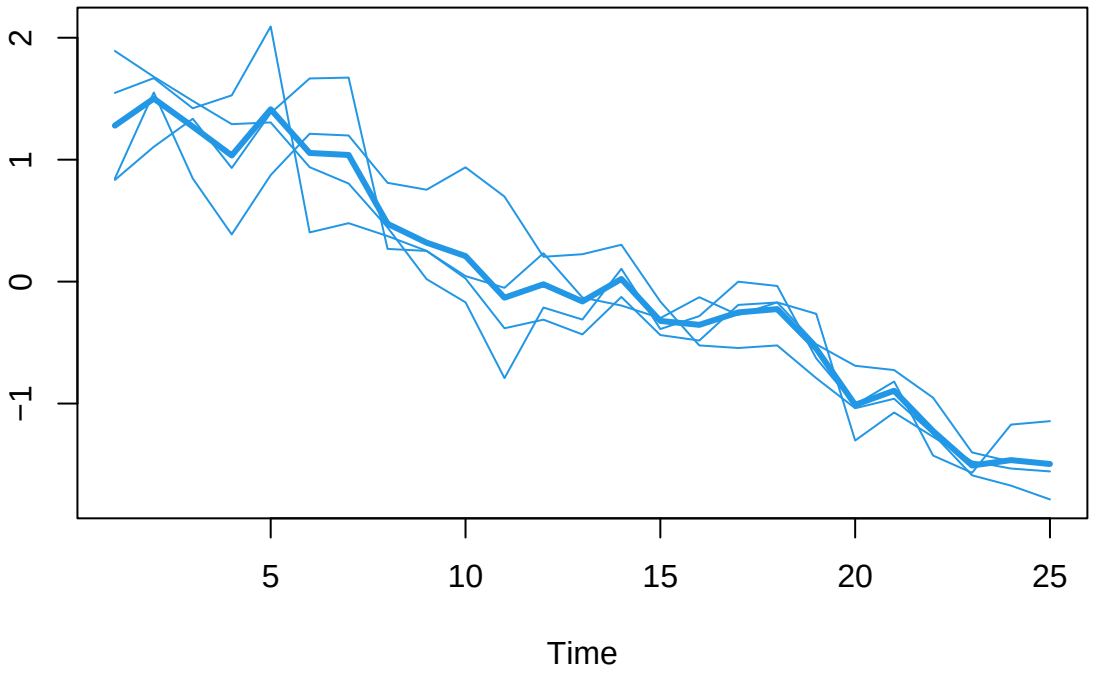

```
ts.plot(cbind(m2, stockstsScaled[, zDTW == 2]), lwd = c(3, rep(1,
    sum(zDTW == 2))), col = 2)
```



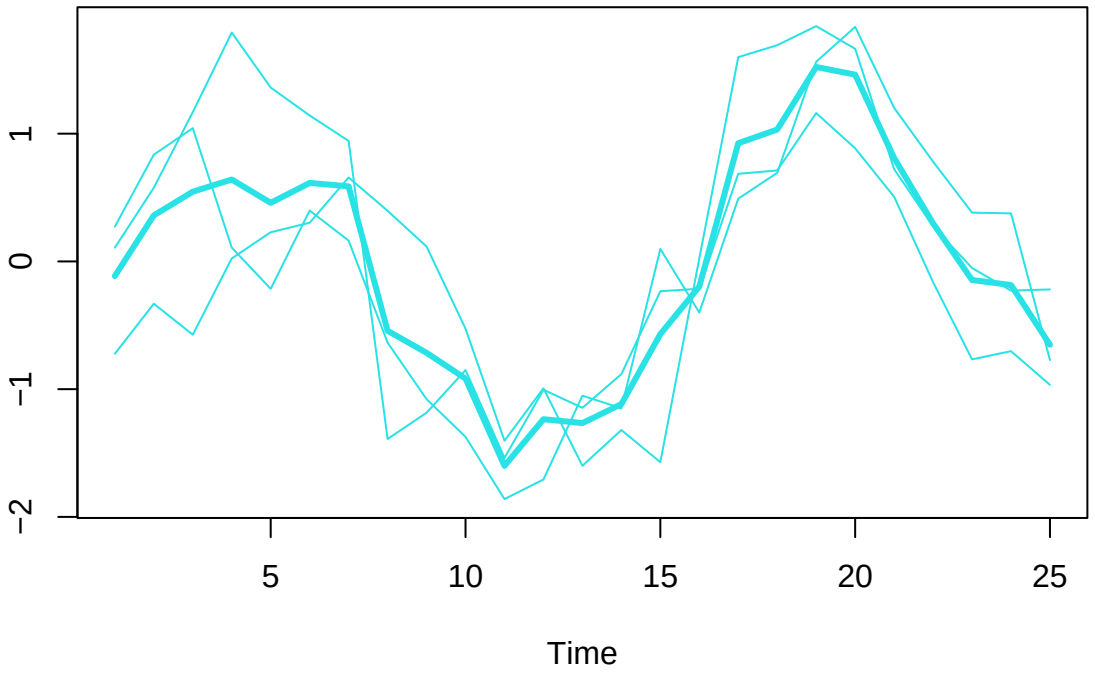

```
ts.plot(cbind(m3, stockstsScaled[, zDTW == 3]), lwd = c(3, rep(1,
    sum(zDTW == 3))), col = 3)
```




```
ts.plot(cbind(m4, stockstsScaled[, zDTW == 4]), lwd = c(3, rep(1,
    sum(zDTW == 4))), col = 4)
```




```
ts.plot(cbind(m5, stockstsScaled[, zDTW == 5]), lwd = c(3, rep(1,
    sum(zDTW == 5))), col = 5)
```



10. Identificar las diferencias entre los dos analisis

Pueden verse resultados similares.

Si bien el cluster 5, el nuevo, corresponde con valores que podrían haber sido considerados atípicos en el análisis anterior, correspondientes a las series: *INTC*, *PG* yy *WMT*.