



CURSO: Minería de Datos

Exercise Solution

Isaac Martín

mayo, 2018

Índice

1	Introducción	1
2	Datos	2
3	Ejercicio	4
3.1	Construir la matriz de series temporales	4
3.2	Representar las series con las que vamos a trabajar	4
3.3	Realizar un análisis cluster usando como variables de interés la media y desviación estándar de cada serie	5
3.4	Representar las series escaladas	8
3.5	Análisis Cluster para cada uno de los cuatrimestres	8
3.6	Análisis Cluster para todo el periodo	14
3.7	Representar gráficamente la media de cada cluster	15
3.8	Localizar atípicos en los clusters	19
3.9	Repetir el análisis, para todo el periodo, empleando la distancia DTW	19
3.10	Identificar las diferencias entre los dos análisis	23

1 Introducción

En este ejercicio vamos a trabajar con los datos del “Dow Jones Index Data Set” que podéis descargar aquí: [DOW JONES INDEX](#). Se trata de datos semanales del Dow Jones Industrial Index.

Attribute Information:

- quarter: the yearly quarter (1 = Jan-Mar; 2 = Apr-Jun).
- stock: the stock symbol (see above)
- date: the last business day of the work (this is typically a Friday)
- open: the price of the stock at the beginning of the week
- high: the highest price of the stock during the week
- low: the lowest price of the stock during the week
- close: the price of the stock at the end of the week
- volume: the number of shares of stock that traded hands in the week
- percent_change_price: the percentage change in price throughout the week
- percent_change_volume_over_last_wk: the percentage change in the number of shares of stock that traded hands for this week compared to the previous week
- previous_weeks_volume: the number of shares of stock that traded hands in the previous week
- next_weeks_open: the opening price of the stock in the following week
- next_weeks_close: the closing price of the stock in the following week
- percent_change_next_weeks_price: the percentage change in price of the stock in the following week
- days_to_next_dividend: the number of days until the next dividend
- percent_return_next_dividend: the percentage of return on the next dividend

2 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				
## 1				3.792670				NA
## 2				-4.428490			1.380223	
## 3				-2.470660			-43.024959	
## 4				1.638310			9.355500	
## 5				5.933250			1.987452	

```
## 6          0.230814          -25.712195
##  previous_weeks_volume next_weeks_open next_weeks_close
## 1              NA      $16.71      $15.97
## 2      239655616      $16.19      $15.79
## 3      242963398      $15.87      $16.13
## 4      138428495      $16.18      $17.14
## 5      151379173      $17.33      $17.37
## 6      154387761      $17.39      $17.28
##  percent_change_next_weeks_price days_to_next_dividend
## 1              -4.428490              26
## 2              -2.470660              19
## 3              1.638310              12
## 4              5.933250               5
## 5              0.230814              97
## 6              -0.632547              90
##  percent_return_next_dividend
## 1              0.182704
## 2              0.187852
## 3              0.189994
## 4              0.185989
## 5              0.175029
## 6              0.172712
```

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

```
##
##  AA  AXP  BA  BAC  CAT  CSCO  CVX  DD  DIS  GE  HD  HPQ  IBM  INTC  JNJ
##  25  25  25  25  25  25  25  25  25  25  25  25  25  25  25
##  JPM  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  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))
```

3 Ejercicio

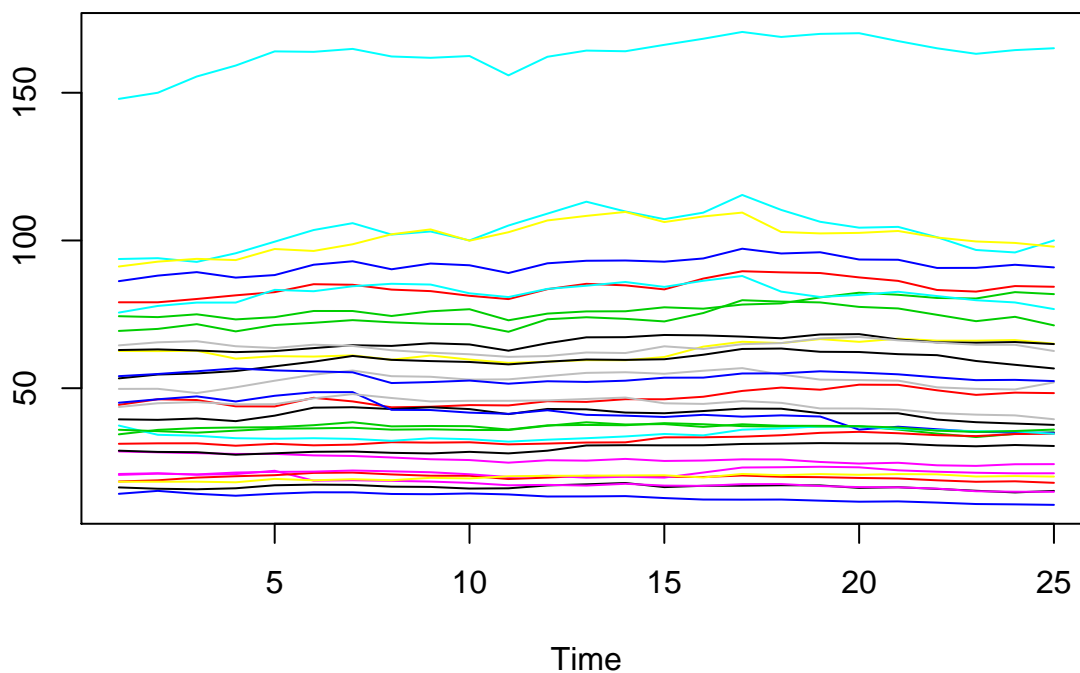
3.1 Construir la matriz de series temporales

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 = attributes(djidata[, "stock"])$levels
n = dim(djidata[stocks == "AA", ])[1]
stocksdata = matrix(0, n, length(stocks))
for (i in 1:length(stocks)) stocksdata[, i] = djidata[djidata$stock ==
  stocks[i], "close"]
colnames(stocksdata) = stocks
stocksts1 = as.ts(stocksdata[1:12, ])
stocksts2 = as.ts(stocksdata[13:25, ])
stocksts = as.ts(stocksdata)
```

3.2 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 basándonos, únicamente, en dos características de las series: su media y desviación típica.

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

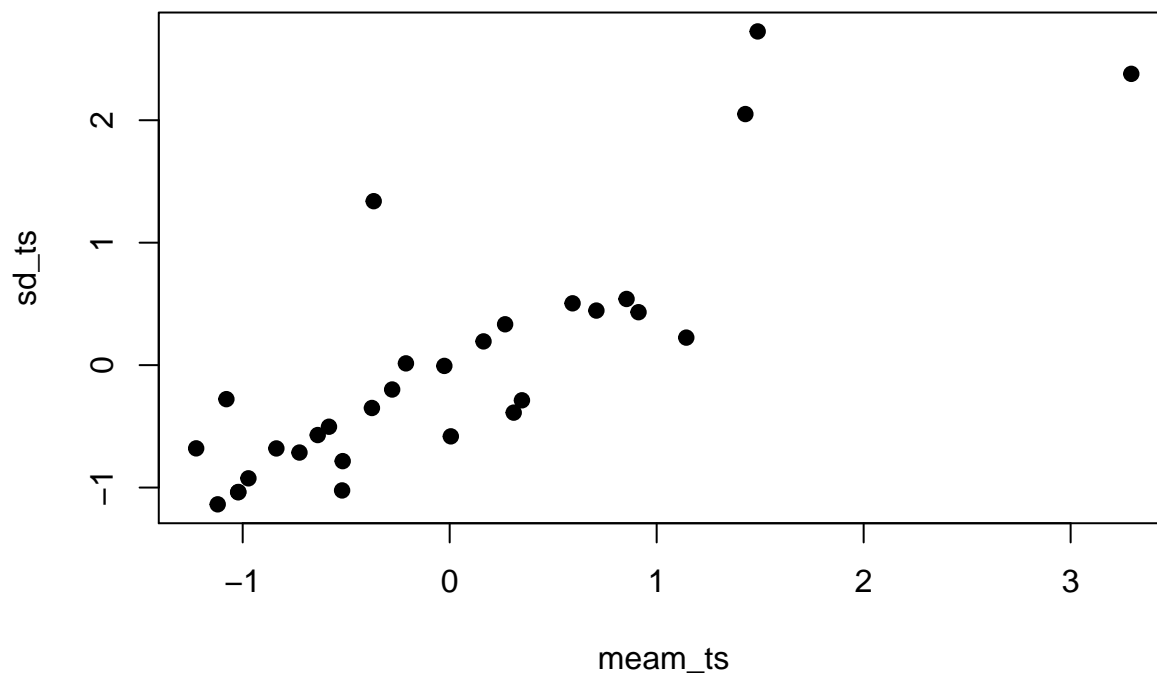
```
library(dplyr)

##
## Attaching package: 'dplyr'

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

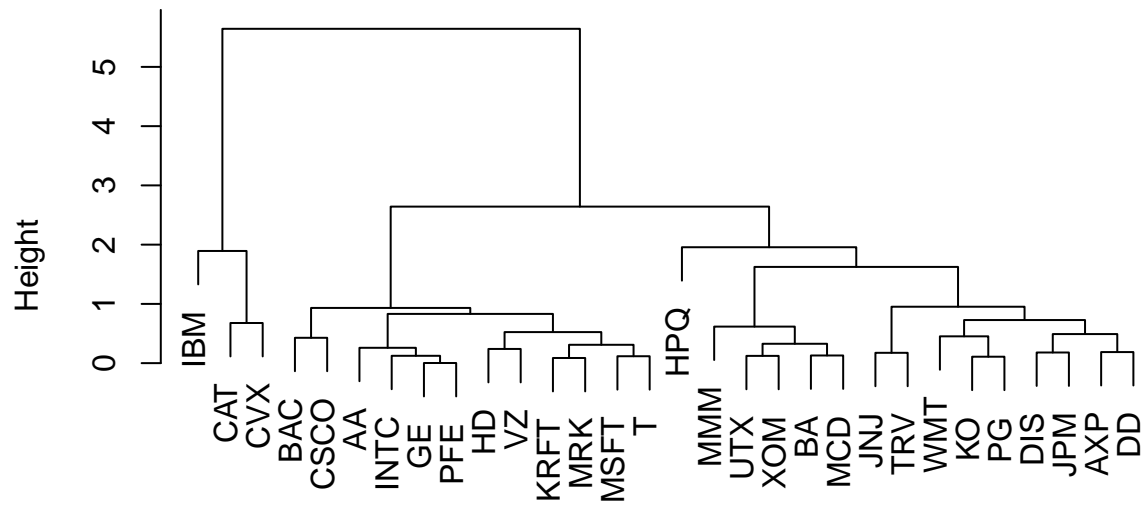
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

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)
```



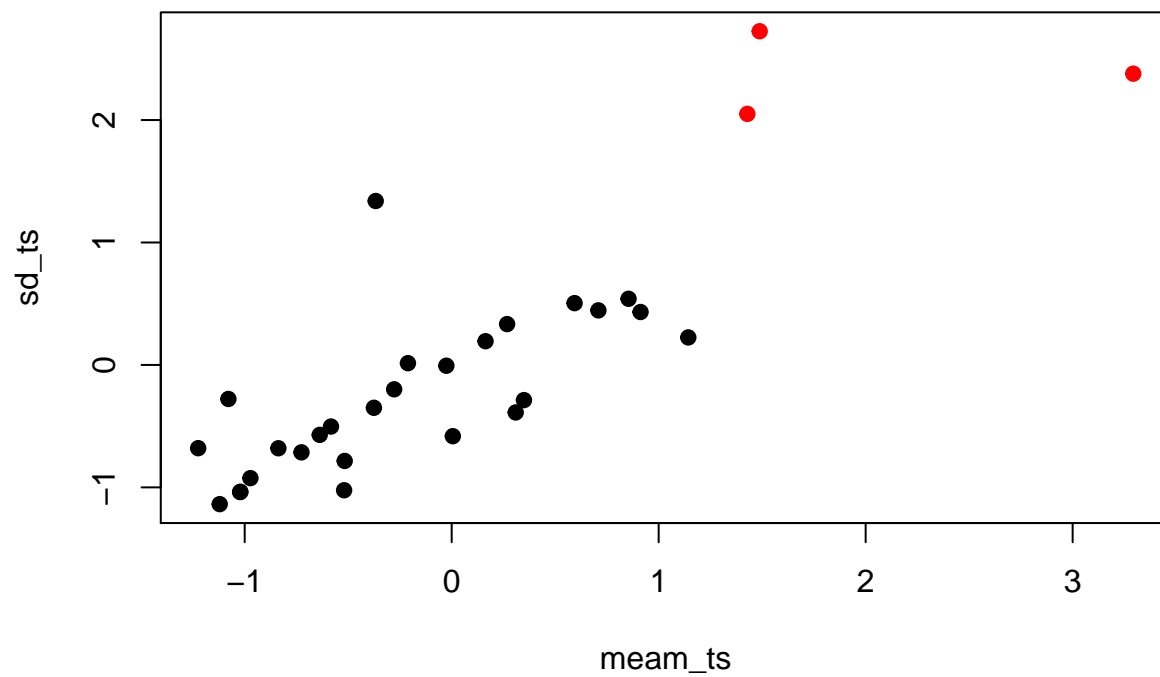
```
library(cluster)
djicluster = hclust(dist(newdata))
plot(djicluster)
```

Cluster Dendrogram



```
dist(newdata)
hclust (*, "complete")
```

```
djiccluster2 = cutree(djiccluster, k = 2)
plot(newdata, pch = 19, col = djiccluster2)
```



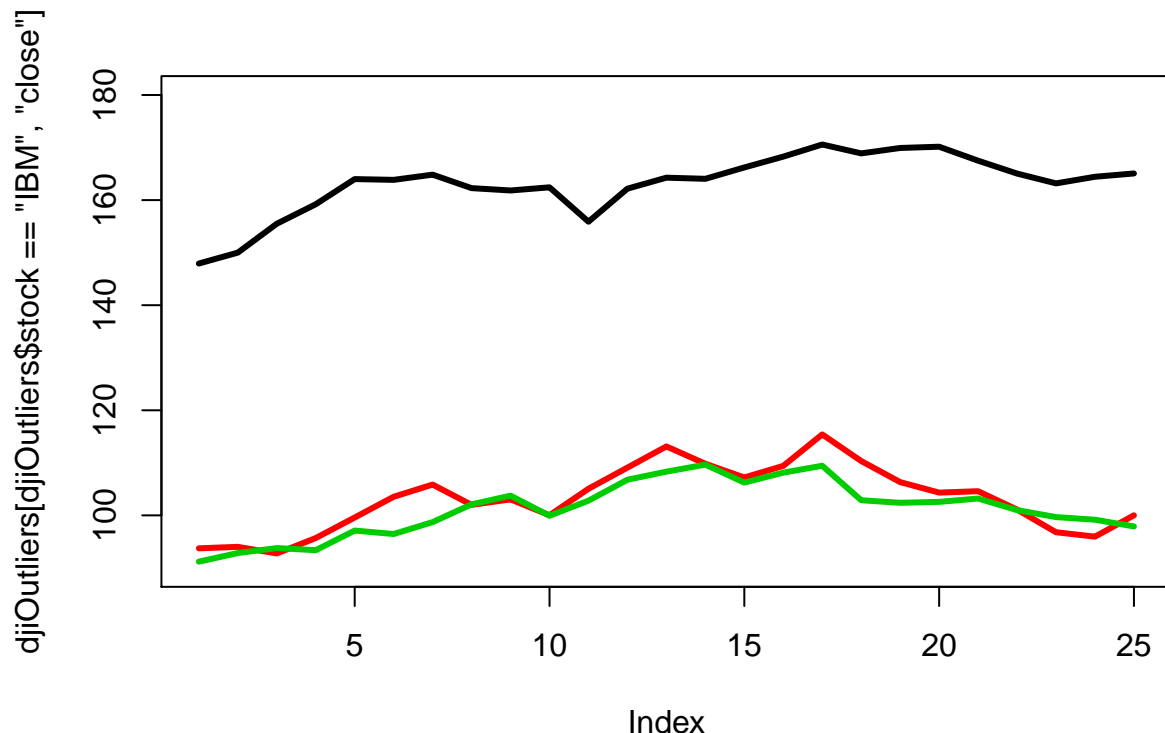
djiclust2

##	AA	AXP	BA	BAC	CAT	CSCO	CVX	DD	DIS	GE	HD	HPQ	IBM	INTC	JNJ
##	1	1	1	1	2	1	2	1	1	1	1	1	2	1	1
##	JPM	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	1

¿Pueden identificarse valores atípicos?

Todos los atípicos 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)
```

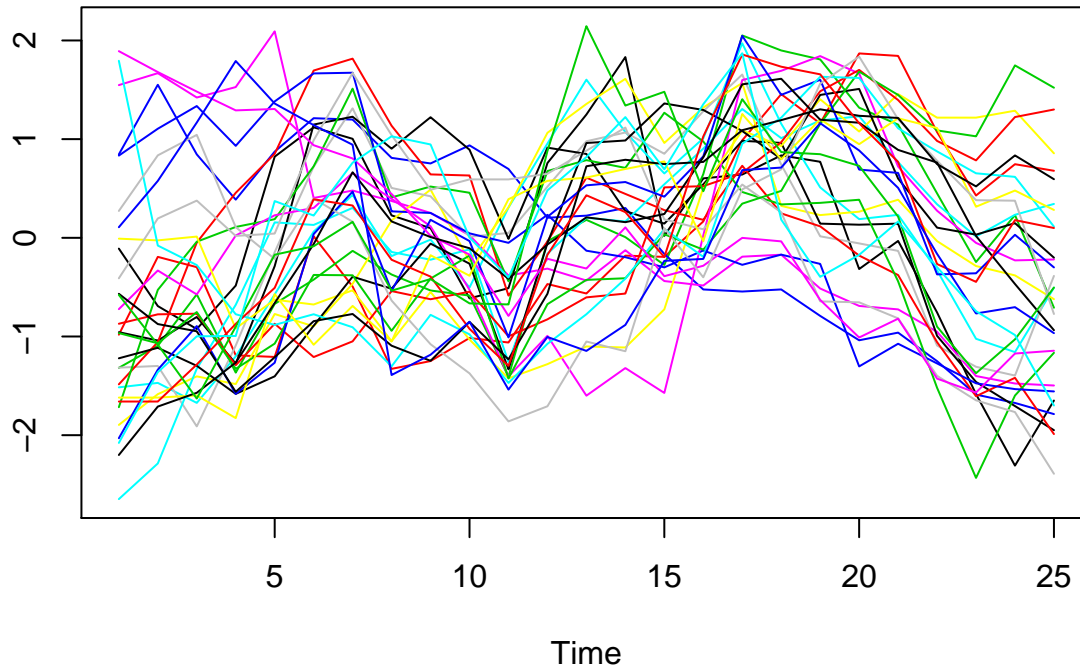


¿Existe relación entre las dos variables consideradas en el análisis? ¿Cómo interpretas este resultado?

Efectivamente, existe una relación lineal positiva entre media y desviación típica. Por lo tanto, lo más lógico sería escalar los datos para que todos tengan la misma media y desviación típica.

3.4 Representar las series escaladas

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

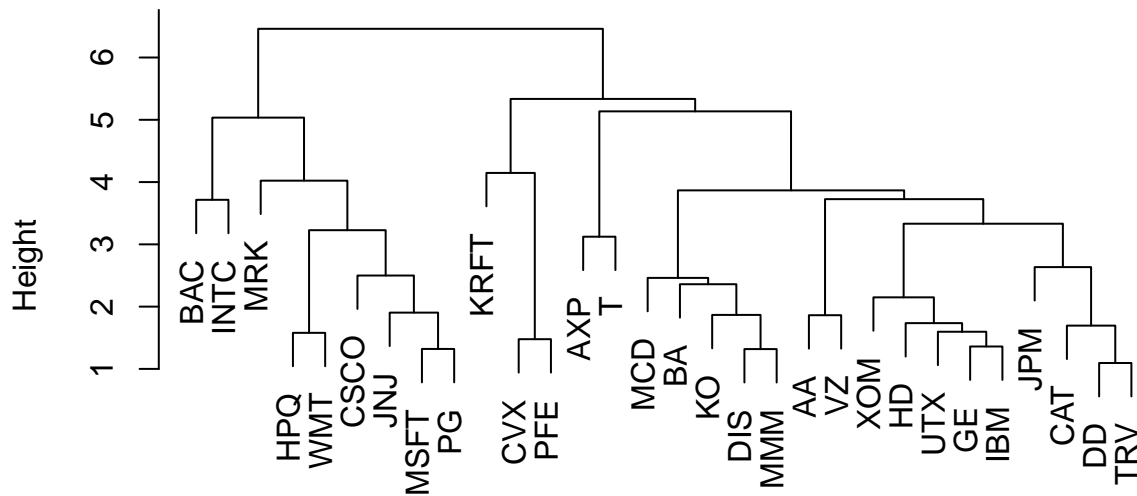


Una vez contruida la matriz con las series temporales, podemos pasar a analizar los datos. Separaremos los datos por cuatrimestres (*quarter*). Realizaremos un análisis cluster para cada uno de los cuatrimestres y otro empleando todo el periodo.

3.5 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)
```


Cluster Dendrogram



```
stocks1Dis
hclust (*, "complete")
```

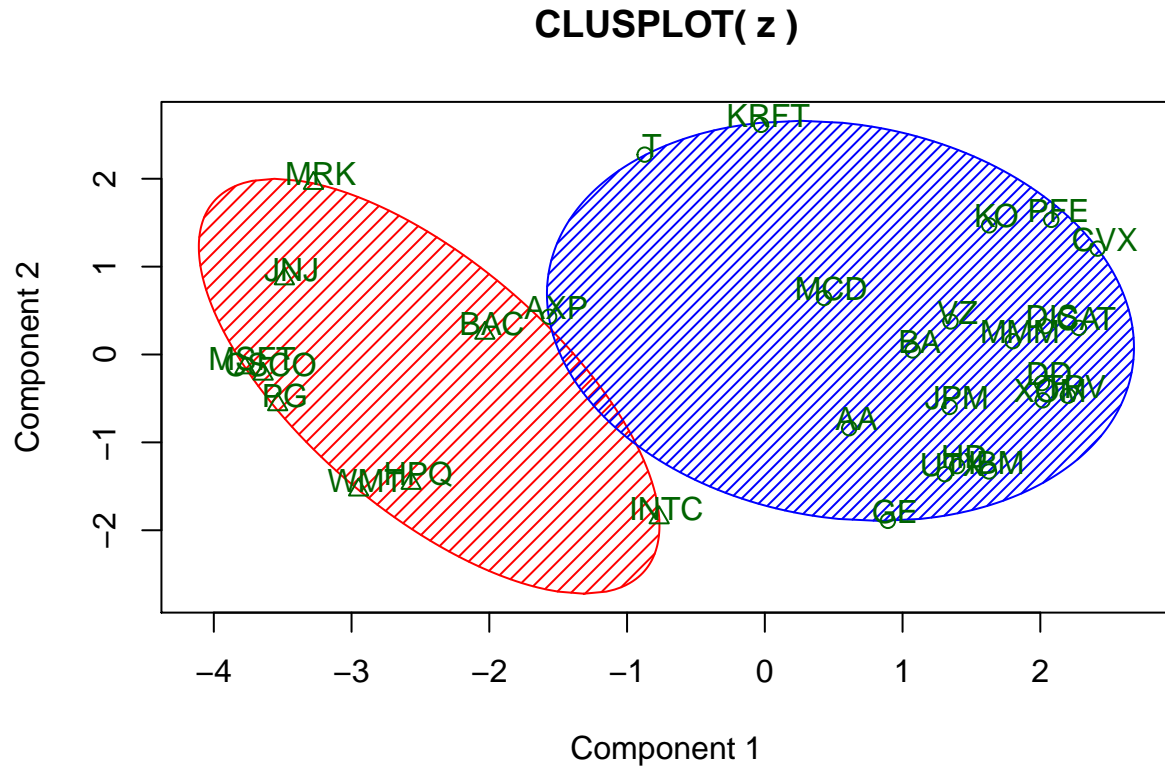
¿En cuantos grupos podemos dividir la muestra?

A la vista del dendrograma intuimos 2 grupos

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

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

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

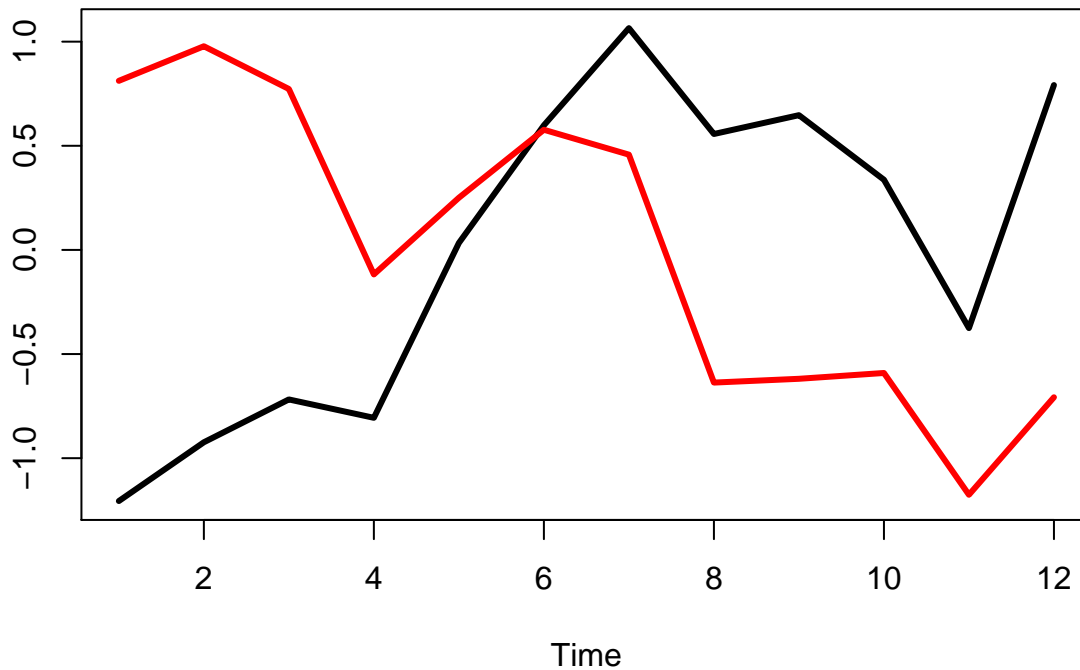


These two components explain 100 % of the point variability.

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)
```

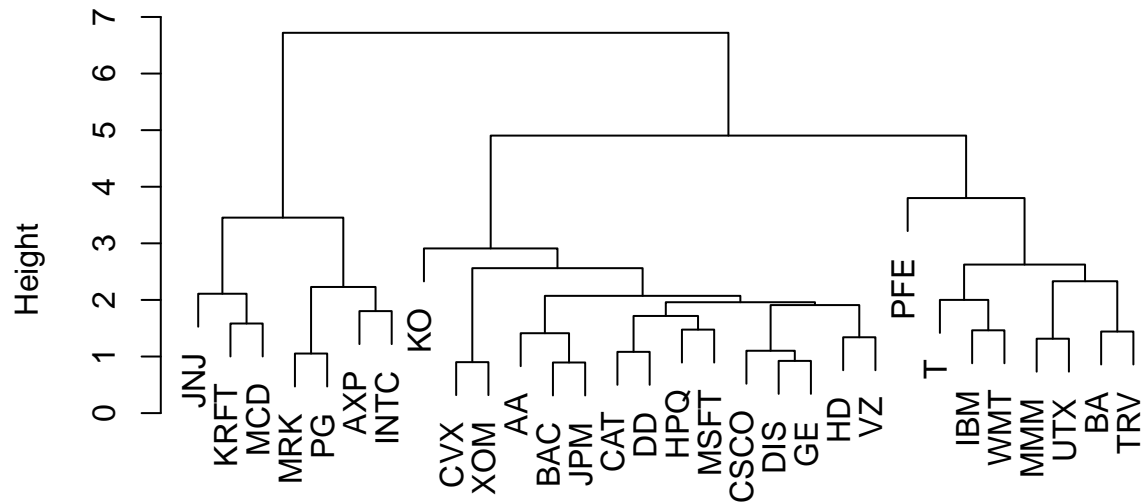


Podemos observar que el primer cluster corresponde a valores que crecen con el tiempo y el segundo cluster a valores que decrecen con el tiempo.

Pasamos a trabajar con el segundo cuatrimestre.

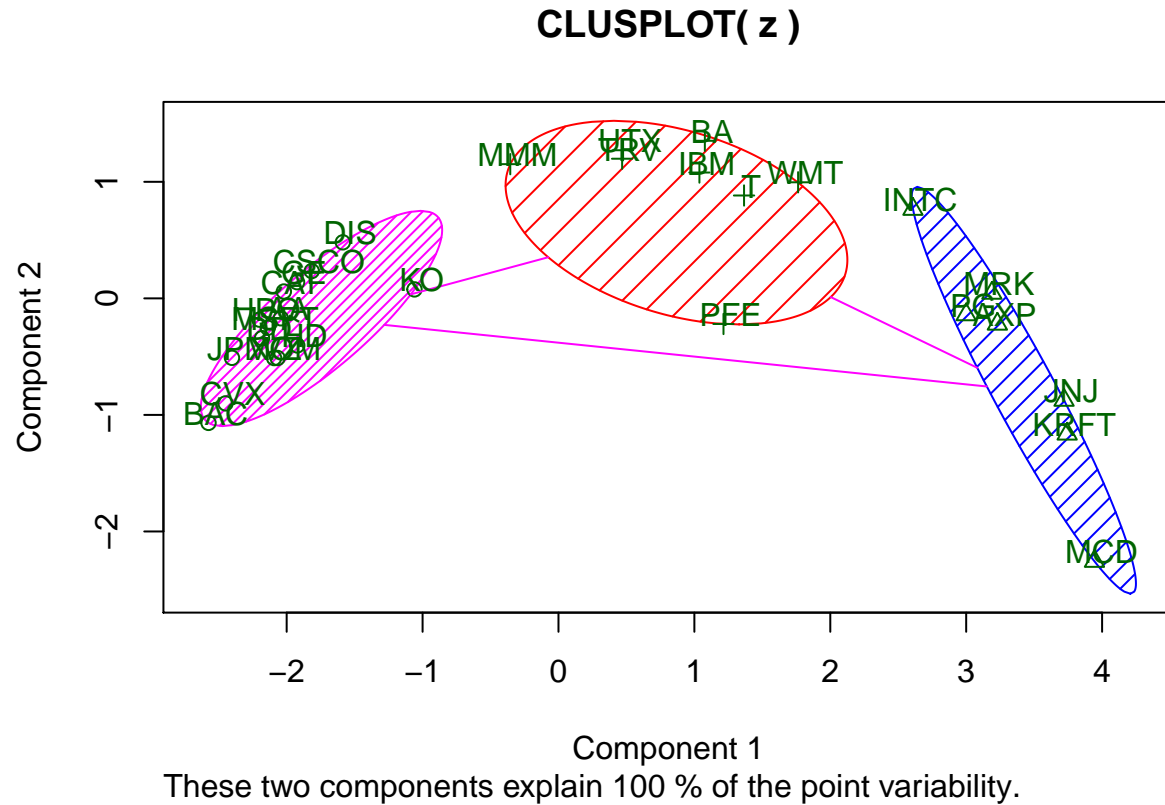
```
stocksts2Scaled = scale(stocksts2)
stocks2Dis = dist(t(stocksts2Scaled))
cluster2 = hclust(stocks2Dis)
plot(cluster2)
```

Cluster Dendrogram



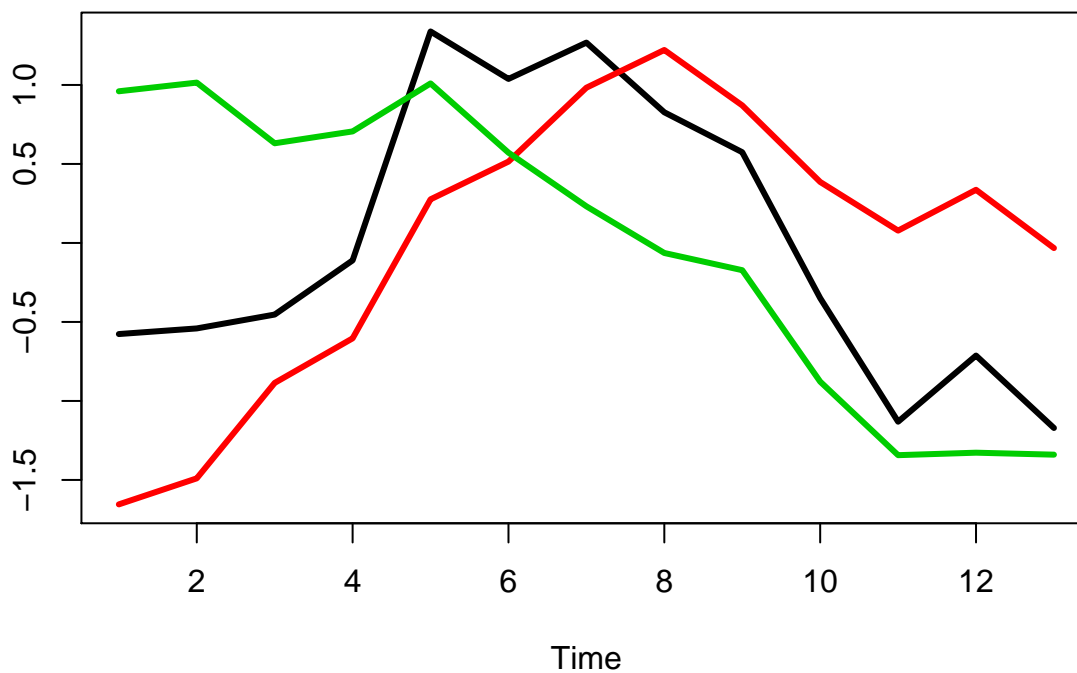
```
stocks2Dis
hclust (*, "complete")
```

```
# A la vista del dendrograma intuimos 3 grupos
z2 = cutree(cluster2, 3)
# z2=kmeans(t(stocksts2Scaled),3,nstart=25)
z = cmdscale(dist(t(stocksts2Scaled)))
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)
```



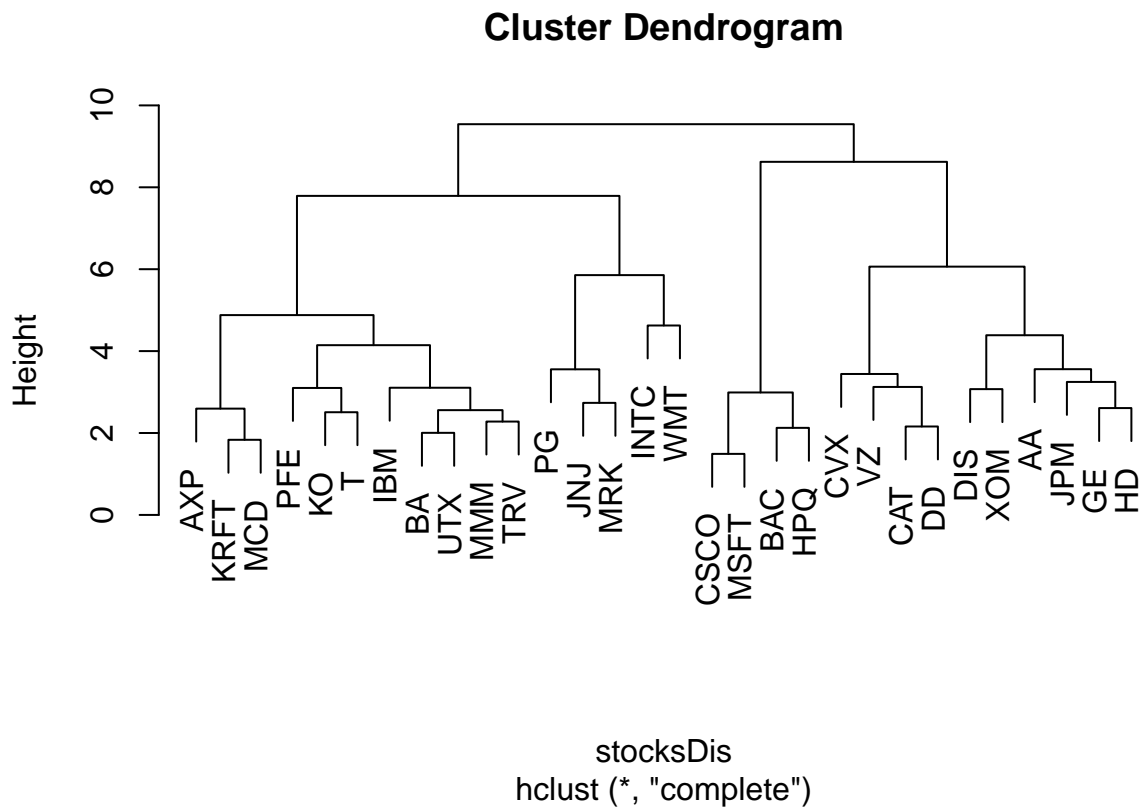
Podemos observar que el primer cluster corresponde a valores que decrecen con el tiempo. El segundo cluster

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.

3.6 Analisis Cluster para todo el periodo

Trabajamos con los datos de todo el periodo analizado.

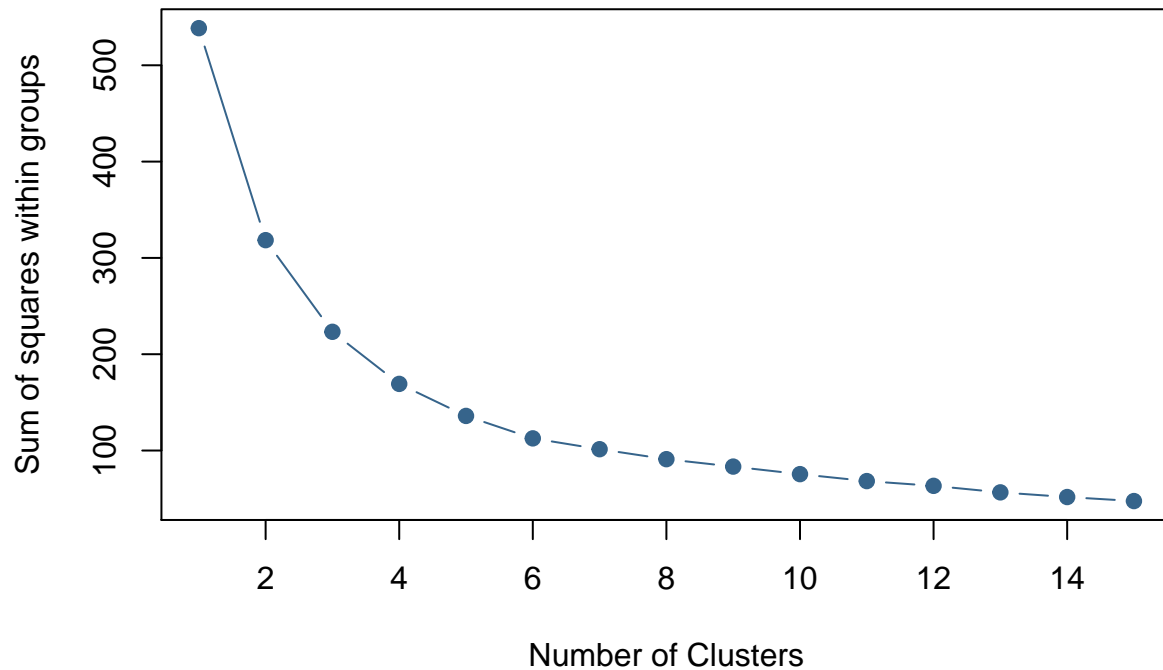
```
# todo el periodo
stockstsScaled = scale(stocksts)
stocksDis = dist(t(stockstsScaled))
clusterTotal = hclust(stocksDis)
plot(clusterTotal)
```



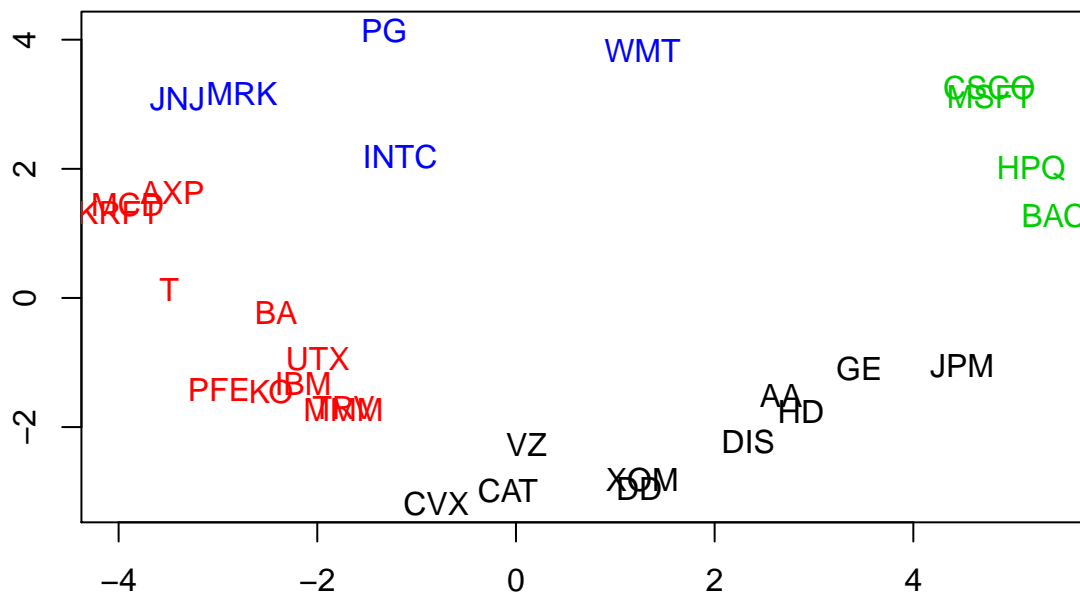
Elegir una

tecnic 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")
```



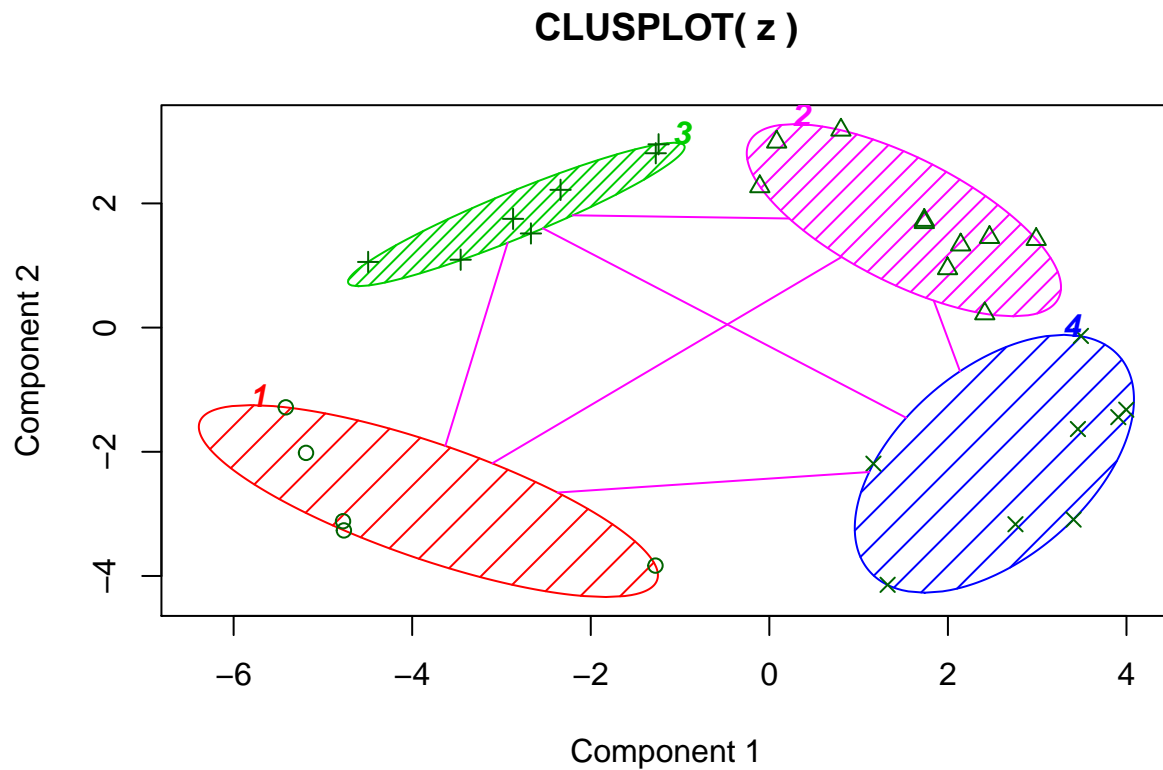
```
# A la vista del dendrograma intuimos 4 grupos
z3 = cutree(clusterTotal, 4)
z = cmdscale(dist(t(stockstsScaled)))
plot(z[, 1], z[, 2], type = "n", xlab = "", ylab = "")
text(z[, 1], z[, 2], rownames(z), cex = 1, col = z3)
```



3.7 Representar gráficamente la media de cada cluster

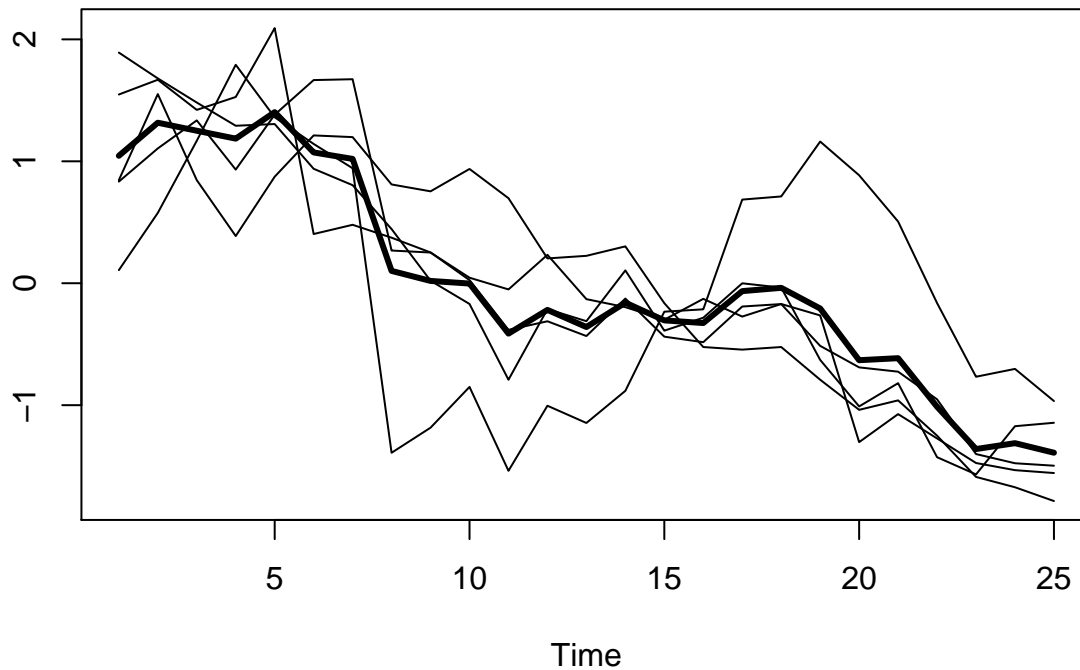
Vamos a ver, las medias de cada uno de los clusters.

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

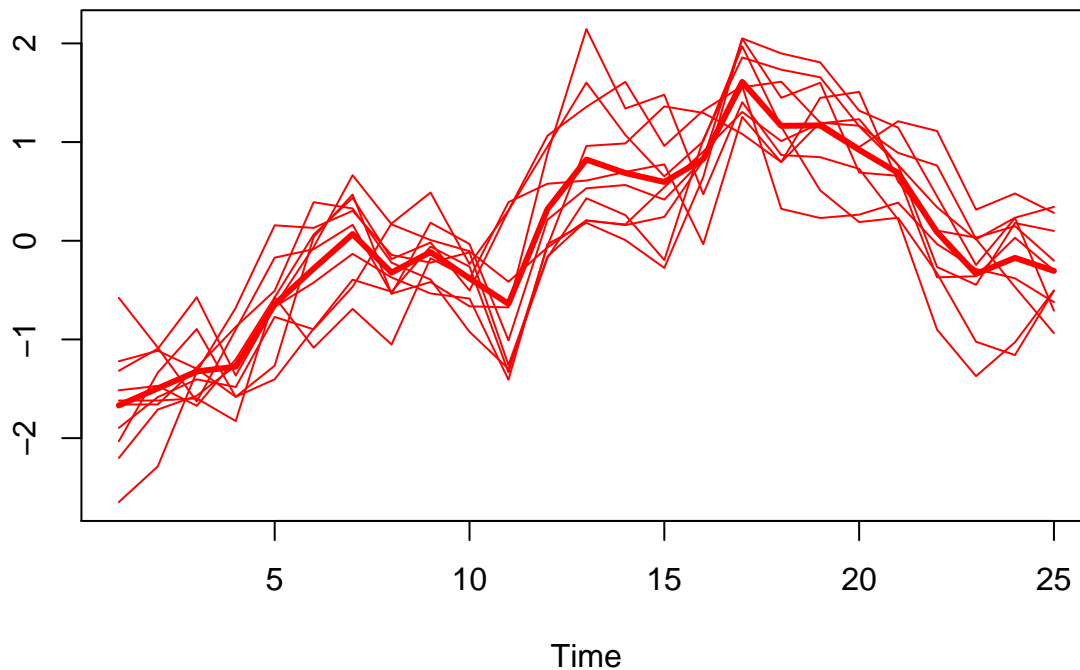


These two components explain 100 % of the point variability.

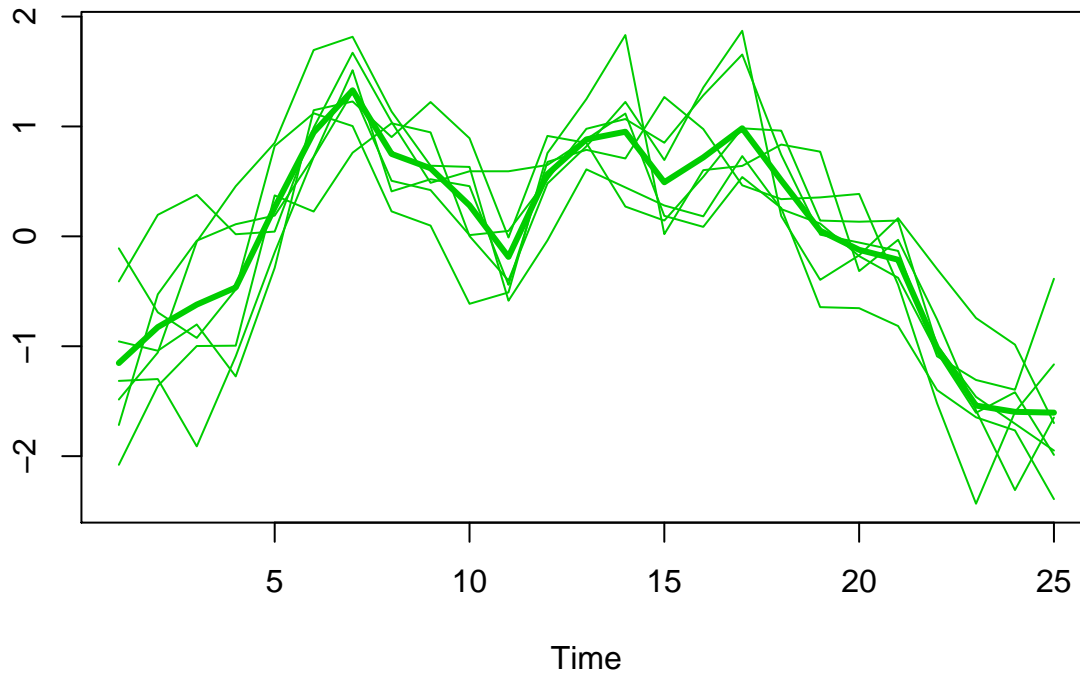
```
ts.plot(cbind((z3$centers[1, ]), stockstsScaled[, z3$cluster ==
  1]), lwd = c(3, rep(1, sum(z3$cluster == 1))), col = 1)
```

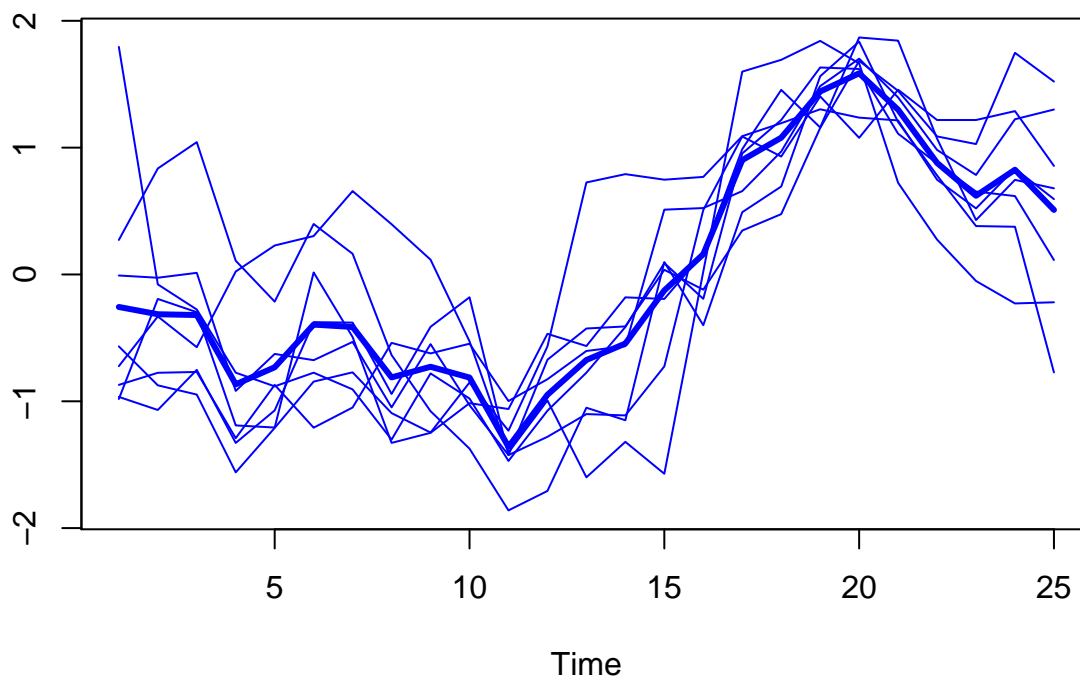
```
ts.plot(cbind((z3$centers[2, ]), stockstsScaled[, z3$cluster ==
  2]), lwd = c(3, rep(1, sum(z3$cluster == 2))), col = 2)
```



```
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)
```

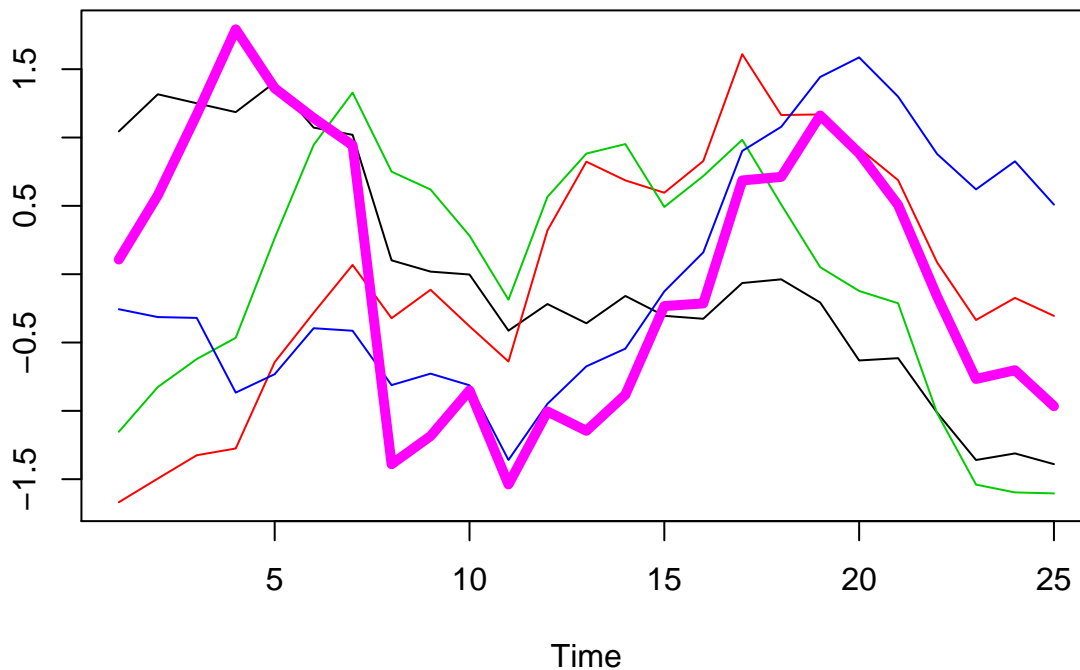


El primer cluster corresponde a valores decrecientes en el tiempo. El segundo cluster corresponde a valores que permanecen aproximadamente constantes hasta la mitad del periodo para crecer a partir de ese momento. El tercer cluster corresponde a valores crecientes en el tiempo. El cuarto cluster corresponde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

3.8 Localizar atípicos 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))
```



3.9 Repetir el análisis, para todo el periodo, empleando la distancia DTW

```
library(dtw)
```

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

```
##
```

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

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

```
##
```

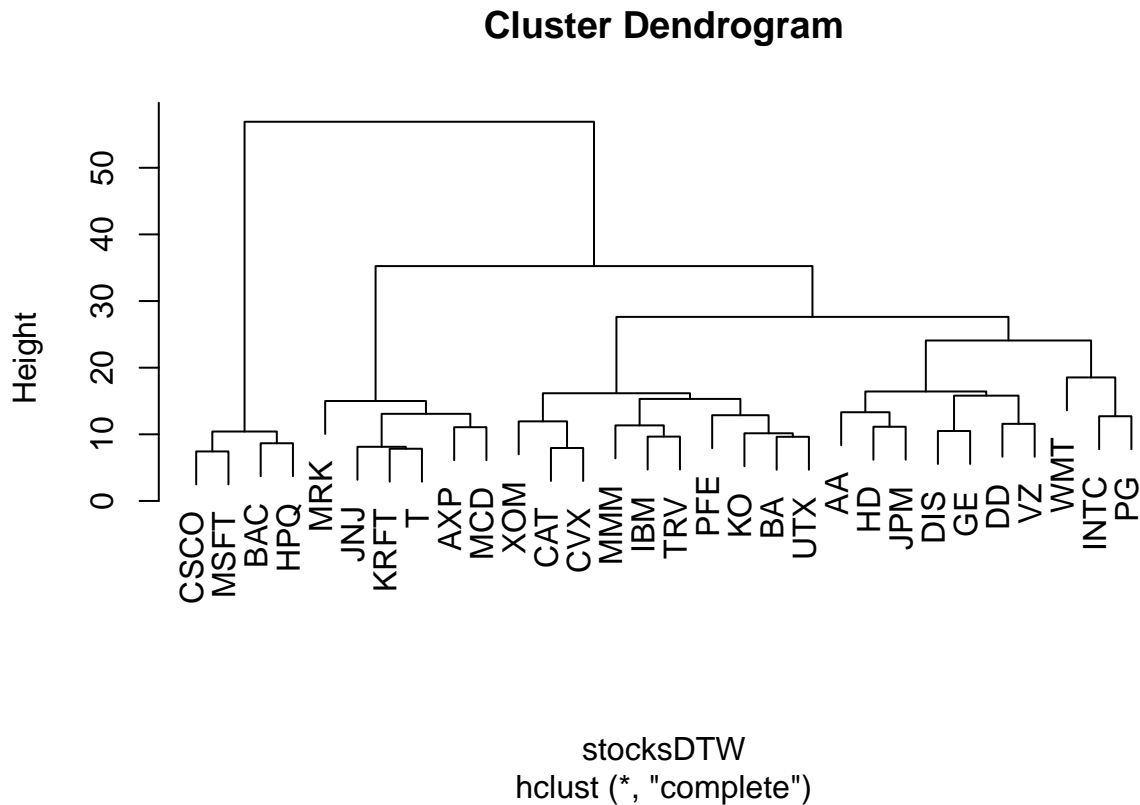
```
##      as.dist, dist
```

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

```
##
```

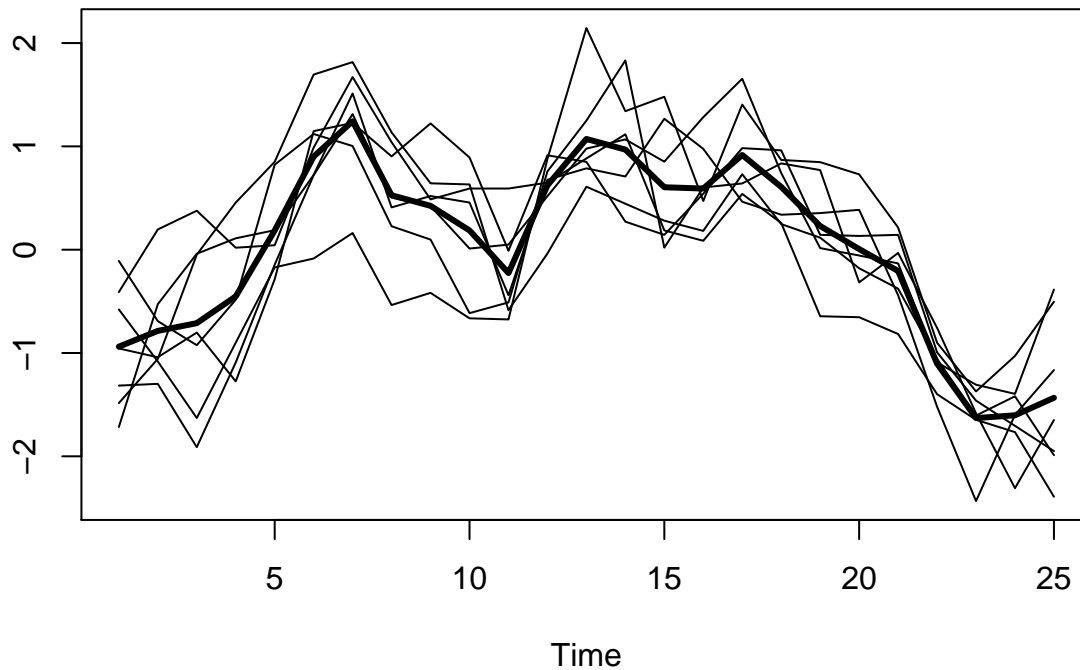
```
##      as.matrix
```

```
## Loaded dtw v1.18-1. See ?dtw for help, citation("dtw") for use in publication.
stocksDTW = dist(t(stockstsScaled), method = "DTW")
clusterTotalDTW = hclust(stocksDTW)
plot(clusterTotalDTW)
```

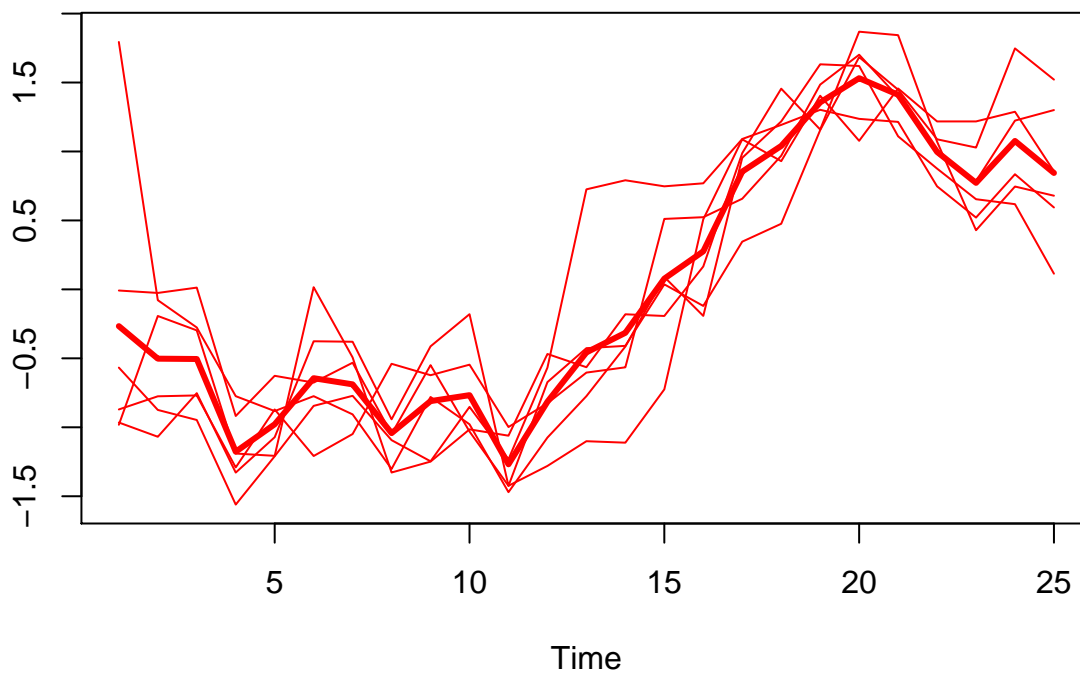


En este caso podríamos quedarnos con 3 o 5 grupos.

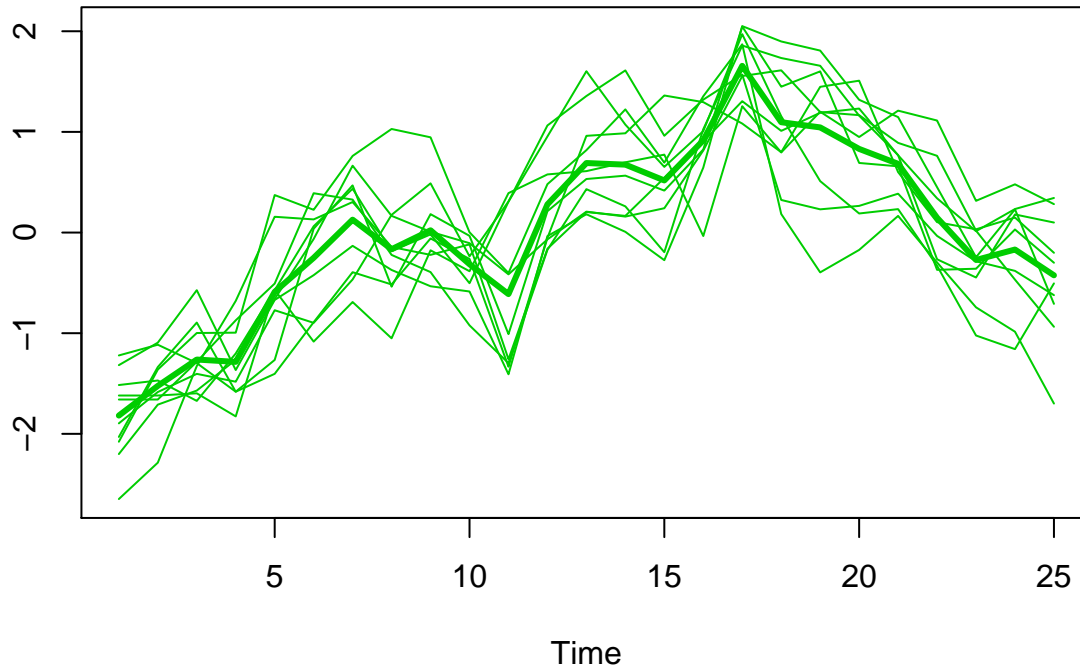
```
zDTW = cutree(clusterTotalDTW, 5)
m1 = apply(stockstsScaled[, zDTW == 1], 1, mean)
m2 = apply(stockstsScaled[, zDTW == 2], 1, mean)
m3 = apply(stockstsScaled[, zDTW == 3], 1, mean)
m4 = apply(stockstsScaled[, zDTW == 4], 1, mean)
m5 = apply(stockstsScaled[, zDTW == 5], 1, mean)
# A la vista del dendrograma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



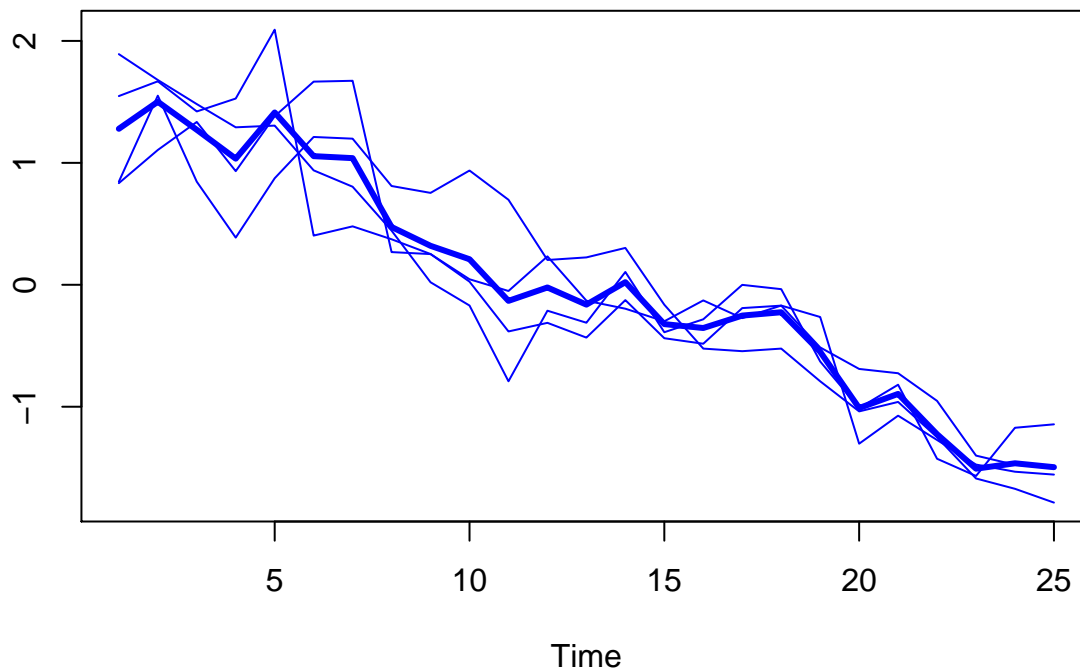
```
ts.plot(cbind(m2, stockstsScaled[, zDTW == 2]), lwd = c(3, rep(1,
  sum(zDTW == 2))), col = 2)
```



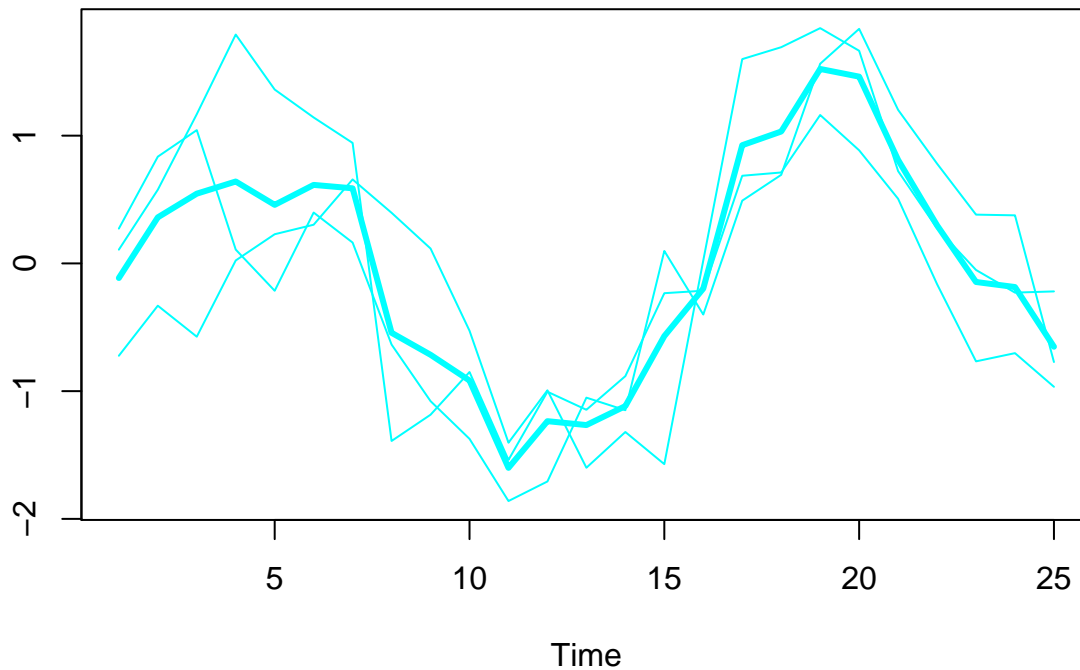
```
ts.plot(cbind(m3, stockstsScaled[, zDTW == 3]), lwd = c(3, rep(1,
  sum(zDTW == 3))), col = 3)
```



```
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)
```



3.10 Identificar las diferencias entre los dos análisisclear

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* y *WMT*.