



**CIMAT**

CENTRO DE INVESTIGACIÓN EN MATEMÁTICAS

## Análisis de Grupos en Series de Tiempo.

**Clase:** Series de Tiempo.

**Alumno:** Luis Esteban Lozano Marrujo.

**Profesores Titulares y Ayudante:**  
Graciela María de los Dolores González Farías,  
José Ulises Márquez Urbina, y  
Víctor de la Fuente.

# Índice

<b>1. Introducción</b>	<b>1</b>
1.1. Aprendizaje Supervisado y No Supervisado . . . . .	1
<b>2. Clustering de Series de Tiempo</b>	<b>2</b>
2.1. Medidas de Similaridad y Extracción de Características . . . . .	2
2.1.1. Extracción de Características . . . . .	3
2.1.2. Medidas de Similaridad . . . . .	3
2.2. <i>Clustering</i> de Series de Tiempo . . . . .	4
2.2.1. <i>Clustering</i> Jerárquico . . . . .	5
2.2.2. <i>Clustering</i> No-Jerárquico: <i>K-medias</i> . . . . .	5
2.2.3. <i>Clustering</i> No-Jerárquico: <i>K-medoids</i> . . . . .	6
2.2.4. <i>Clustering</i> No-Jerárquico: <i>K-Shape</i> . . . . .	7
2.2.5. Métodos de <i>clustering</i> con base en Observaciones . . . . .	7
2.2.6. Métodos de <i>clustering</i> con base en Características . . . . .	8
2.2.7. Métodos de <i>clustering</i> con base en Modelos . . . . .	9
2.2.8. Fuzzy <i>clustering</i> . . . . .	10
2.3. Prototipos de Series de Tiempo . . . . .	11
2.3.1. Media y Mediana . . . . .	11
2.3.2. Partición Alrededor de los Medoids (PAM) . . . . .	11
2.3.3. Extracción de <i>Shapes</i> . . . . .	11
<b>3. Clustering de Acciones del NASDAQ</b>	<b>12</b>
3.1. Visualizando Series de Tiempo Multivariadas . . . . .	12
3.2. <i>Clustering</i> Jerárquico . . . . .	14
3.3. <i>Clustering</i> No Jerárquico . . . . .	26
3.3.1. k-medias . . . . .	26
3.3.2. PAM . . . . .	27
3.3.3. K-Shape . . . . .	28
3.4. Fuzzy <i>clustering</i> . . . . .	29
<b>4. Conclusiones</b>	<b>30</b>

# 1. Introducción

El *clustering* de Series de Tiempo (ST) pretende obtener información relevante sobre los grupos que se pueden formar dado un conjunto de series de tiempo sin etiquetas predefinidas. El problema de agrupar objetos es un problema muy natural cuándo se tiene una colección de objetos con más de dos elementos, por lo que el problema de *clustering* puede surgir en todas las ciencias. A continuación, explicamos ambos tipos de aprendizaje y ubicamos el problema de *clustering* como un método de aprendizaje. Luego, presentamos el *clustering* de series de tiempo. Finalizamos con una aplicación de *clustering* del precio al cierre del día de *trading* de los valores del NASDAQ 100.

## 1.1. Aprendizaje Supervisado y No Supervisado

Los objetivos del aprendizaje estadístico suelen ser mejorar el entendimiento y la predicción. En un inicio, los métodos de aprendizaje se pueden dividir en dos: **aprendizaje supervisado y aprendizaje no supervisado** (Hastie y col., 2001; James y col., 2014).

1. El **aprendizaje supervisado** puede ocurrir cuándo el objetivo es predecir el valor de una variable respuesta  $y$  (también conocida como variable dependiente, etiqueta o *ground truth*) con base en distintas covariables de entrada  $\mathbf{x}$  (también conocida como variables independientes o características). Las dos principales áreas de aprendizaje supervisado son regresión y clasificación.
  - En un problema de regresión suponemos que la relación entre la respuesta  $y \in \mathbb{R}$  y las covariables  $\mathbf{x} \in \mathbb{R}^p$  se puede expresar como

$$y = f(\mathbf{x}) + \epsilon,$$

donde  $f(\cdot)$  es una función de  $\mathbf{x}$  fija pero desconocida que representa la información sistemática que  $\mathbf{x}$  nos da sobre  $y$ , y  $\epsilon$  es el error con media cero. La función  $f$  usualmente es desconocida y se debe estimar de alguna manera. Sea  $\hat{y} = \hat{f}(\mathbf{x})$ . Al conocer los valores de la variable respuesta, podemos estimar  $f$  al plantear una función de pérdida de mínimos cuadrados dada por

$$\begin{aligned}\mathbb{E}(y - \hat{y})^2 &= \mathbb{E}[f(\mathbf{x}) + \epsilon - \hat{f}(\mathbf{x})]^2 \\ &= \mathbb{E}[f(\mathbf{x}) - \hat{f}(\mathbf{x})]^2 + \mathbb{V}(\epsilon),\end{aligned}$$

donde  $\mathbb{E}$  y  $\mathbb{V}$  son los operadores de esperanza y varianza.

- Similarmente, en clasificación simple tenemos  $y \in \{0, 1\}$  y es común estimar  $\hat{f}$  con una función de pérdida de entropía cruzada dada por

$$-(y \ln(\hat{y}) + (1 - y) \ln(1 - \hat{y})).$$

Si alguien está aprendiendo a hacer algo con un maestro o supervisor, hay retroalimentación sobre si se está haciendo la tarea de manera correcta o incorrecta. Tal supervisión se simboliza en las funciones de pérdida, ya que estamos comparando la predicción  $\hat{y}$  con lo observado  $y$ .

2. **Aprendizaje no supervisado**, donde hay ausencia de etiquetas, y el objetivo es describir la asociación y patrones en un conjunto de datos de entrada  $x$ .

El **análisis de subgrupos** (también llamado **análisis de clústers** o *clustering*) es una tarea de aprendizaje no supervisado que busca agrupar **objetos** de tal manera que los objetos en el mismo **sub-grupo** (llamado **clúster**) son más **similares** (en algún sentido) entre sí, que comparados con aquellos en otros sub-grupos. Este tipo de análisis se suele utilizar como estadística descriptiva para evaluar si hay o no distintos sub-grupos en los datos, donde cada sub-grupo representa objetos con propiedades substancialmente diferentes (Hastie y col., 2001). El **análisis de sub-grupos en series de tiempo** ocurre cuando tenemos por objetos varias series de tiempo y queremos agrupar las series en categorías o clústers.

## 2. *Clustering* de Series de Tiempo

Las ST son datos dinámicos donde la variable de interés depende del tiempo, lo que quiere decir que las observaciones realizadas se hacen de manera cronológica. Los conjuntos de datos provenientes de ST frecuentemente tienen una naturaleza altamente dimensional, y son de mucho interés debido a que aparecen en las ciencias e ingenierías.

El *clustering* se utiliza inicialmente para datos transversales (de sección cruzada). Sin embargo, debido a la relevancia del análisis de grupos y de las series de tiempo, se vuelve necesario desarrollar métodos de *clustering* para el análisis de las series de tiempo. Así, el *clustering* de series de tiempo es ventajoso ya que nos permite descubrir patrones en datos que tienen una dimensión temporal (Aghabozorgi y col., 2015). TSCC, afirman que la clasificación y *clustering* de los datos con dependencias temporales es muy diferente a clasificar y agrupar estáticos. Los datos de series de tiempo pueden dividirse en cuatro categorías que contienen muchos tipos de datos con un componente temporal (Alqahtani y col., 2021):

1. Los datos de series de tiempo **univariados** se dan cuando sólo se observa una variable a lo largo del tiempo. En este caso no hay mucha agrupación a realizarse.
2. Los datos de series de tiempo **multivariados** ocurren cuando se tiene un conjunto de series de tiempo y se suele representar en forma matricial. Una matriz de dice ser un *array* 2-D (la dimensión de las filas y la dimensión de las columnas).
3. Los **campos tensoriales o tensores** son un *array* con más de dos dimensiones. Al lidiar con datos espacio-temporales, este tipo de datos se generaliza para considerar series de tiempo de redes y grafos, series de tiempo de posiciones espaciales de objetos en movimiento, y series de tiempo de distribuciones espaciales.
4. Los **multicampos** son un conjunto de campos, y nos dan mucha flexibilidad en la práctica. Por ejemplo, Maugeri, utilizan una combinación de indicadores: temporales, de intensidad, de tendencia, y regionales, para identificar grupos con patrones epidémicos similares en regiones de Italia.

En el análisis de ST es común encontrar la presencia de ruido, una alta dimensionalidad y fuertes correlaciones. Estos tres retos han llevado a la tubería del análisis convencional de ST que se puede apreciar en la Figura 1 (Alqahtani y col., 2021). Además, notemos que esta tubería también se puede utilizar en otros problemas de aprendizaje supervisado como regresión y clasificación. Vale la pena recordar que esta tubería ocasionalmente se utiliza en un proceso recursivo. Las áreas donde el *clustering* de ST se ha aplicado son: visión computacional, reconocimiento de voces, reconocimiento de caras, ingenierías, finanzas, economía, investigación de operaciones, biología, medicina, e incluso puede ser relevante para investigaciones multidisciplinarias.



Inspirado por: Alqahtani et al, 2021.

Figura 1: Tubería del análisis convencional de Series de Tiempo (ST).

### 2.1. Medidas de Similaridad y Extracción de Características

Recordemos que uno de los objetivos del análisis de grupos consiste en agrupar nuestros objetos tal que las unidades dentro de cada clúster sean similares entre sí, y los clústers sean distintos. Para este objetivo es esencial definir apropiadamente la medida de similaridad. Las medidas de similaridad se

deben elegir para que traten de manera apropiada los valores atípicos, las diferencias en amplitud, y la distorsión en el eje del tiempo, que pueden presentarse en los datos dinámicos de series temporales. Frecuentemente, es necesario complementar la elección de una medida de similaridad apropiada con un pre-procesamiento adecuado para obtener una aproximación apropiada de la representación subyacente de los datos que nos facilite obtener las agrupaciones.

### 2.1.1. Extracción de Características

La extracción de características es una forma de reducción de dimensión que ayuda a reducir el costo computacional cuando se tienen datos altamente dimensionales (Alqahtani y col., 2021). El objetivo es generar una abstracción que represente los datos y sea fiel a las características de los datos originales<sup>1</sup>. Además, se sabe que la elección de una medida de similaridad nos auxilia en el tratamiento de observaciones atípicas, diferencias en amplitud, y distorsiones en el eje de tiempo. Encima, Para la selección de características relevantes es indispensable comunicación con los expertos del dominio. La calidad de los grupos formados en *clustering* de las ST depende de la extracción de las características, la medida de similaridad y la técnica de análisis utilizadas. La selección de las características apropiadas suele ser compleja y es determinante en que los algoritmos de *clustering* sean exitosos para producir agrupaciones de calidad (Barandas y col., 2020). Algunos métodos son:

- **Logaritmo, Operador de Diferencias/Lags, Transformación de Box-Cox.**
- **Análisis de Componentes Principales (PCA).** Es una técnica clásica de reducción de dimensionalidad que se ha aplicado exitosamente a datos transversales y series de tiempo (Alqahtani y col., 2021; Hastie y col., 2001). El PCA computa las principales direcciones de variabilidad en los datos, y transforma el conjunto de datos original a un nuevo conjunto de variables no correlacionadas, donde las variables no correlacionadas son combinaciones lineales de las variables originales.  
Una extensión de interés es el análisis de componentes principales funcional (FPCA) de FPCA, que se utilizó en el área de finanzas para estimar la volatilidad implicada y la curva de retornos.
- ***n*-gramas.** Al transformar los datos de ST en un conjunto de características se puede perder parte de la naturaleza secuencial de los datos. Los *n*-gramas buscan capturar esta naturaleza secuencial de los datos y por lo tanto nos pueden ayudar a simplificar datos altamente dimensionales (Alqahtani y col., 2021; Hagiwara, 2021; Manning y Schütze, 1999; Patel y Arasanipalai, 2021).
- Otros: **Transformada de Fourier Discreta (DFT), Multidimensional Scaling (MDS), Continuous Bag of Words (CBOW), Discrete Wavelet Transform (DWT), Shapelets, Adaptive Piecewise Constant Approximation (APCA), Piecewise Aggregate Approximation (PAA), Chebyshev polynomials (CHEB), Symbolic Aggregate approXimation (SAX)**, etc (Alqahtani y col., 2021; Barandas y col., 2020).

### 2.1.2. Medidas de Similaridad

Las medidas de similaridad nos permiten cuantificar que tan parecidas son un par de ST. Una elección simple es utilizar alguna distancia, aunque existen medidas de similaridad hechas a la medida para tomar en cuenta los retos presentes en el *clustering* de ST. Yahyaoui, Wang, dtwclust, TSClust, dan una revisión de las medidas de similaridad para ST. A continuación discutimos con más detalle algunas posibles medidas de similaridad que nos serán útiles en el desarrollo de técnicas de *clustering*, pero existe una gran cantidad de opciones para evaluar la similaridad entre series

---

<sup>1</sup>Para técnicas reducción de dimensionalidad diseñadas para ST se recomienda consultar: Sorzano, C.O.S.; Vargas, J.; Montano, A.P. A survey of dimensionality reduction techniques. arXiv 2014.

justo como veremos. La elección de una medida de similaridad apropiada requiere conocimiento del dominio de aplicación y los retos inherentes al conjunto de datos de ST.

Sea  $X = \{x_{ij} : i = 1, \dots, I, j = 1, \dots, J\} = \{\mathbf{x}_i = (x_{i1}, \dots, x_{ij}, \dots, x_{iJ})^T : i = 1, \dots, I\}$  la matriz de datos observados donde  $x_{ij}$  representa la  $j$ -ésima variable observada de la  $i$ -ésima realización, y  $\mathbf{x}_i$  representa el vector de observaciones para la  $i$ -ésima realización.

### Medidas de Similaridad basadas en Distancias

Maharaj y col., 2019, con base en Everitt y col., 2011, afirma que la clase de distancias más utilizada en el análisis de clústers es la clase de distancias de Minkowski dada por

$${}_r d_{il} = \left( \sum_{j=1}^J |x_{ij} - x_{lj}|^r \right)^{1/r}, \quad \text{para } r \geq 1.$$

Algunos casos especiales son la distancia de taxista ( $r = 1$ ) y la distancia Euclídea ( $r = 2$ ). Una extensión interesante es la clase de **distancias de Minkowski**, donde la distancia se define de la siguiente manera:

$${}_r \tilde{d}_{il} = \left( \sum_{j=1}^J w_j^r |x_{ij} - x_{lj}|^r \right)^{1/r}, \quad r \geq 1,$$

tal que  $w_j$  representa un peso apropiado para la  $j$ -ésima variable, para  $j = 1, \dots, J$ .

Para utilizar las medidas de distancia en las técnicas de análisis de clústers es de gran utilidad recolectar todas las distancias para cada par de objetos en una matriz de distancias. La matriz de distancias de Minkowski se define como

$${}_r D = \left\{ {}_r d_{il} = \left( \sum_{j=1}^J |x_{ij} - x_{lj}|^r \right)^{1/r} : i, l = 1, \dots, I \right\}, \quad r \geq 1.$$

### Distancia de Fréchet

Montero y Vilar, 2014, afirma que Fréchet<sup>2</sup> introdujo esta distancia para medir la proximidad entre curvas continuas, pero se ha utilizado frecuentemente en el marco de series de tiempo. Sea  $M$  el conjunto de todas las posibles sucesiones de  $m$  pares que preservan el orden de observaciones de la forma

$$r = ((X_{a_1}, Y_{b_1}), \dots, (X_{a_m}, Y_{b_m})),$$

con  $a_i, b_j \in \{1, \dots, T\}$  tal que  $a_1 = b_1 = 1, \dots, a_m = b_m = T$ , y  $a_{i+1} = a_i$  ó  $a_i + 1$  junto con  $b_{i+1} = b_i$  ó  $b_i + 1$ , para  $i \in \{1, \dots, m-1\}$ . Entonces, la **distancia de Fréchet** está dada por

$$d_F(X_T, Y_T) = \min_{r \in M} \left( \max_{i=1, \dots, m} |X_{a_i} - Y_{b_i}| \right).$$

Esta distancia tiene la ventaja de tomar en cuenta el orden temporal de la ST y se puede utilizar para series de longitudes distintas.

## 2.2. Clustering de Series de Tiempo

Suponemos una estructura de datos proveniente de ST dada por una matriz de observaciones

$$X = \{x_{tm} : t = 1, \dots, T, m = 1, \dots, M\} = \{\mathbf{x}_i = (x_{i1}, \dots, x_{ij}, \dots, x_{iJ})^T : i = 1, \dots, T\},$$

donde  $t = 1, \dots, T$  son los tiempos de observación de las series y  $M$  es el número de series.

---

<sup>2</sup>Fréchet M. (1906). Wuelques Points du Calcul Fonctionnel. Rendiconti del Circolo Matematico di Palermo (18841940), 22(1), 1-72.

### 2.2.1. Clustering Jerárquico

El *clustering* jerárquico define una estructura de árbol para los datos sin etiquetas, por medio de agregar los objetos en un a un árbol de clústers. Usualmente, los métodos de *clustering* son de dos tipos: **aglomerativo** (de abajo a arriba o *bottom-up*) y **divisivo** (de arriba a abajo o *top-down*) (Alqahtani y col., 2021).

El *clustering* aglomerativo comienza con  $M$  clústers, cada uno incluye exactamente una serie que forma su propio clúster. El *clustering* aglomerativo luego une los clústers menos disimilares en clústers más grandes al utilizar la matriz de disimilaridades. Luego, se agrupan los objetos hasta que tenemos todos los objetos en un solo grupo (Xu y Brereton, 2005, cómo se citó en Maharaj y col., 2019). Concisamente, el método de *clustering* aglomerativo es:

1. Comenzamos con  $M$  clústers. Computamos la matriz de similaridades para los  $M$  clústers;
2. En la matriz de distancia, buscamos la distancia mínima

$$d(C_c, C_{c'}) = \min_{1 \leq p \neq q \leq M} d(C_p, C_q),$$

donde  $d(\cdot, \cdot)$  es la función de distancia o similaridad. Luego, combinamos los clústers  $C_c, C_{c'}$  en un nuevo clúster  $C_{cc'}$ ;

3. Actualizamos la matriz de distancias al computar las distancias entre el clúster  $C_{cc'}$  y todos los otros clústers; y
4. Repetimos los pasos 2 y 3 hasta que se obtiene un solo clúster.

Los clústers que se agrupan para formar un nuevo clúster dependen de la función de distancia que se utiliza para medir la similaridad. Algunos métodos para definir las funciones de distancia son (para otros métodos consultar Maharaj y col., 2019):

- **Single linkage** (método de vecino más cercano): la distancia entre par de clústers se determina por las dos series más cercanas a los diferentes clústers.  
Al utilizar este método de agrupación, se tienden a generar clústers grandes que causa un efecto encadenador (Maharaj y col., 2019, con base en Everitt y col., 2011). Por lo tanto, dos clústers con distintas propiedades pueden terminar agrupándose debido a la existencia de ruido. Pero, si los grupos están bien separados, este método suele funcionar bien.
- **Complete linkage**: se utiliza la distancia más lejana entre par de series para definir una distancia interclúster.
- **Método de Ward (método de mínima varianza)**: el objetivo del método de Ward es minimizar el incremento de la suma de cuadrado de errores dentro de cada clase.

El resultado del *clustering* jerárquico se puede visualizar con un **dendrograma** que es una visualización en forma de árbol que busca representar la jerarquía de los clústers según el sea la similitud.

### 2.2.2. Clustering No-Jerárquico: $K$ -medias

Como lo dice su nombre, el **clustering no jerárquico** asigna a un conjunto de objetos en  $K$  clústers sin alguna estructura jerárquica. Este método de agrupación usualmente se plantea como un problema de optimización, donde la función objetivo representa la variabilidad dentro de los clústers. Uno de los métodos de *clustering* no jerárquico más populares es  $k$ -medias (Xu y Brereton,

2005, cómo se citó en Maharaj y col., 2019). Matemáticamente, el problema de *clustering* por  $k$ -medias es:

$$\min \sum_{i=1}^I \sum_{k=1}^K u_{ik} d_{ik}^2 = \sum_{i=1}^I \sum_{k=1}^K u_{ik} \|\mathbf{x}_i - \mathbf{h}_k\|^2,$$

tal que  $\sum_{k=1}^K u_{ik} = 1, \quad u_{ik} \geq 0, u_{ik} \in \{0, 1\},$

donde  $u_{ik}$  indica la pertenencia del  $i$ -ésimo objeto al  $K$ -ésimo clúster, y  $d_{ik}^2 = \|\mathbf{x}_i - \mathbf{h}_k\|^2$  indica la distancia Euclídea al cuadrado entre el  $i$ -ésimo objeto y el centroide del  $K$ -ésimo clúster. Este método, busca una partición óptima de los datos al minimizar el criterio de suma de errores al cuadrado dentro del clúster (Maharaj y col., 2019 con base en MacQueen, 1967, y Xu y Brereton, 2005). Algorítmicamente, el método de agrupación de  $K$ -medias es:

1. Iniciar una  $K$ -partición aleatoriamente o con conocimiento previo. Computamos los centroides que son medias dentro de cada clúster considerando sólo las observaciones que pertenecen a cada clúster.
2. Asignamos cada unidad en el conjunto de datos al clúster más cercano utilizando una medida de distancia entre las series y el centroide.
3. Volvemos a computar los prototipos con base en la partición actual.
4. Repetimos el paso 2 y 3 hasta que no hay cambio para cada clúster.

### 2.2.3. Clustering No-Jerárquico: $K$ -medoids

El método de *clustering* de  $K$ -medoids (también conocido como Partición Alrededor de Medoids, PAM) agrupa los objetos por medio de los medoids, donde los medoids representan las características prototípicas de cada clúster (Kaufman y Rousseeuw, 1990). El algoritmo de *clustering* por  $k$ -medoids es (Maharaj y col., 2019):

1. Esta fase secuencialmente selecciona  $K$  unidades 'centrales' que se utilizan como los medoids iniciales.
2. Si la función objetivo se puede reducir al intercambiar una unidad seleccionada con una unidades sin seleccionar, entonces el cambio se hace. Esto continua hasta que la función objetivo no se puede hacer más chica. Entonces, al considerar un conjunto de  $I$  unidades en el conjunto de observaciones  $\mathbf{x}$  y el conjunto de medoids  $\tilde{\mathbf{X}}$  que es un subconjunto de  $\mathbf{x}$  con  $K$  unidades. Podemos formalizar el modelo como sigue:

$$\min \sum_{i=1}^I \sum_{k=1}^K u_{ik} \tilde{d}_{ik}^2 = \sum_{i=1}^I \sum_{k=1}^K u_{ik} \|\mathbf{x}_i - \tilde{\mathbf{x}}_k\|^2,$$

tal que  $\sum_{k=1}^K u_{ik} = 1, \quad u_{ik} \geq 0, u_{ik} \in \{0, 1\},$

donde  $u_{ik}$  indica la pertenencia del  $i$ -ésimo objeto al  $k$ -ésimo clúster, y  $\tilde{d}_{ik}^2$  indica la distancia Euclídea al cuadrado entre el  $i$ -ésimo objeto y el medoid del  $k$ -ésimo clúster.

#### 2.2.4. Clustering No-Jerárquico: *K-Shape*

El *clustering* por *K-Shape* tiene una distancia propia, llamada SBD, y una función de extracción de formas propia. Este algoritmo requiere *z*-estandarización. La distancia con base en formas (SBD del inglés *Shape-Based Distance*) se propone como parte del algoritmo de *K-Shape* que es una alternativa más rápida que DTW. La distancia utiliza la correlación cruzada con normalización de coeficientes (NCCc) entre dos series para cuantificar su similaridad y por lo tanto, es sensible a la escala lo que hace necesaria la *z*-estandarización. Este algoritmo fue desarrollado por Paparrizos y Gravano, 2015 como se citó en Sardá-Espinosa, 2018.

##### *Shape-Based Distance*

La distancia SBD se computa con la siguiente fórmula

$$SBD(x, y) = 1 - \frac{\max\{NCCc(x, y)\}}{\|x\| \|y\|},$$

donde  $\|\cdot\|$  es la norma en  $l_2$ . Esta distancia soporta series con longitudes distintas similarmente al DTW.

#### 2.2.5. Métodos de *clustering* con base en Observaciones

En los métodos de *clustering* con base en observaciones usualmente se utiliza alguna técnica de pre-procesamiento y luego se utiliza alguna distancia para evaluar la similaridad entre los objetos. Estos métodos son útiles para conjuntos de datos de tamaño no muy grande, las series pueden tener longitudes distintas. Nos permite evaluar que tan parecidas son las series en un sentido geométrico.

##### Dynamic Time Warping

El *Dynamic Time Warping* (DTW) es una clase de algoritmos que nos permiten comparar series de tiempo al dar medidas de distancia que son insensibles a compresiones y alargamientos locales, el *warping* deforma óptimamente una serie de tiempo a la otra. La lógica detrás del DTW es que, dadas dos series de tiempo, se puede estirar o comprimir localmente una serie para hacer que se parezca a la otra lo más posible. Despues del *warping*, se computa la distancia al sumar las distancias individuales de los componentes alineados (Giorgino, 2009; Maharaj y col., 2019).

El DTW se propone en los 70s en el contexto de reconocimiento de voz, para tomar en cuenta las tasas de habla entre las personas que hablan el lenguaje. Esto es muy útil cuando se puede encontrar un puntaje de distancia. Por ejemplo, entre las pronunciaciones *no* y *nooo*, entonces el DTW sería insensible a la duración del sonido de la *o*. De hecho, hoy en día el uso de DTW va más allá de sólo distorsiones en el tiempo, por ejemplo: puede tomar en cuenta distorsiones en ángulos (Giorgino, 2009). Montero y Vilar, 2014 mencionan algunos puntos a notar del DTW:

- Es superior a la distancia Euclídea, ya que la distancia Euclídea es muy sensible a distorsiones en el tiempo. Además que la distancia Euclídea se utiliza sólo en series de tiempo de longitud igual.
- Posteriormente a su amplio uso en el reconocimiento de palabras aisladas, se ha utilizado extensamente en: bioinformática, robótica, física, finanzas, reconocimiento de voz y señales, análisis de electrocardiogramas, *clustering* de expresión de genes, monitoreo de procesos.

En el DTW, las series de tiempo se deforman (*warp*) no linealmente para que haya cierta concordancia. Ahora ilustramos la idea subyacente del DTW. Suponemos dos series de tiempo  $\mathbf{x}_i$  y  $\mathbf{x}'_i$  con longitudes respectivas  $T$  y  $T'$  no necesariamente iguales. Tal idea subyacente consiste en encontrar la transformación óptima  $\phi$  tal que

$$d(\mathbf{x}_i, \mathbf{x}'_i) = \min_{\phi} d_{\phi}(\mathbf{x}_i, \mathbf{x}'_i).$$

En palabras, encontramos la deformación de los ejes de tiempo de las ST  $\mathbf{x}_i$  y  $\mathbf{x}'_i$  que nos lleva las dos series a estar lo más cerca posible, es decir, la distancia total entre  $\mathbf{x}_i$  y  $\mathbf{x}'_i$  se computa por medio de la ‘curva deformada’ que asegura que cada dato en  $x_i$  se compara al punto más cercano en  $x'_i$ . Para definir la curva deformada, sea

$$\phi_l = (\varphi_l, \psi_l), \quad l = 1, 2, \dots, L,$$

bajo las condiciones de frontera:

$$\phi_1 = (1, 1), \phi_L = (T, T');$$

y monotonicidad:

$$\varphi_1 \leq \dots \leq \varphi_l \leq \dots \leq \varphi_L; \text{ y } \psi_1 \leq \dots \leq \psi_l \leq \dots \leq \psi_L.$$

El efecto de aplicar el *warping* a ambas series consiste en alinear los índices de tiempo de ambas series por medio de las funciones  $\varphi_l, \psi_l$ . La disimilaridad total entre las series de tiempo después del *warp* es:

$$\sum_{l=1}^L d(\mathbf{x}_{i,\varphi_l}, \mathbf{x}'_{i,\psi_l}) m_{l,\phi},$$

donde  $d(\cdot, \cdot)$  es alguna función de distancia, usualmente la distancia Euclídea. Ya que hay varios *warplings*, la distancia DTW es aquella que corresponde al *warping óptimo*  $\hat{\phi}_l = (\hat{\varphi}_l, \hat{\psi}_l)$  que minimiza la disimilaridad total entre  $\mathbf{x}_i$  y  $\mathbf{x}'_i$ :

$$d_{DTW}(\mathbf{x}_i, \mathbf{x}'_i) = \sum_{l=1}^L d(\mathbf{x}_{i,\hat{\varphi}_l}, \mathbf{x}'_{i,\hat{\psi}_l}) m_{l,\hat{\phi}}.$$

A pesar del gran espacio de búsqueda, el tiempo de computación del DTW es  $O(T \cdot T')$  (Maharaj y col., 2019; Sardá-Espinosa, 2018). Para llevar a cabo el *clustering* computamos la matriz de disimilaridad y alimentamos esta matriz a los algoritmos de *clustering*.

## 2.2.6. Métodos de *clustering* con base en Características

Los resultados del análisis de grupos, dependen de la métrica de similaridad. La distancia euclídea es una elección que puede resultar demasiado simple. El DTW tiene por objetivo identificar perfiles geométricos similares. Ahora, exploramos métricas que son más apropiadas para discriminar entre las estructuras de dependencia subyacentes. Maharaj y col., 2019, llaman a este tipo de análisis de grupos el ***clustering* con base en características**, y recomiendan que se debe ser consciente del proceso generador para poder determinar qué características se considerarán. Particularmente, las características extraídas de series de tiempo estacionarias no van a ser necesariamente iguales de aquellas extraídas de ST no estacionarias.

### Autocorrelación y Autocorrelación Parcial

Galeano y Peña, 2000, introducen una métrica para ST estacionarias de segundo orden con base en la **función de autocorrelación** (ACF) estimada. Dada una serie de tiempo  $\mathbf{x}_t = \{x_t : t = 1, 2, \dots, T\}$ , sea  $\hat{\rho}_{\mathbf{x}r} = (\hat{\rho}_1, \dots, \hat{\rho}_R)$  el estimador de función de correlación de  $\mathbf{x}_t$  con lag  $R$  tal que  $\hat{\rho}_i \approx 0$  para  $i > R$ . Una distancia entre dos series de tiempo  $\mathbf{x}_t$  y  $\mathbf{x}'_t$  se puede definir como:

$$d_{ACF}(\mathbf{x}_t, \mathbf{x}'_t) = \sqrt{(\hat{\rho}_{\mathbf{x}r} - \hat{\rho}_{\mathbf{x}'_t r})^T W (\hat{\rho}_{\mathbf{x}r} - \hat{\rho}_{\mathbf{x}'_t r})},$$

donde  $W$  es una matriz de pesos. Caiado, 2006, implementó tres maneras posibles de computar  $d_{ACF}$ :

1. La opción más simple es utilizar pesos uniformes  $W = I$  y denotamos  $d_{ACFU}$ .
2. Una opción simple es utilizar pesos con decaimiento geométrico  $W = D$ , donde  $D$  es una matriz de pesos geométricos y denotamos  $d_{ACFG}$ .
3. La tercera opción consiste en utilizar la distancia de Mahalanobis entre los coeficientes de correlación dados por la fórmula truncada de Bartlett, y denotamos  $d_{ACFM}$ .

Además, Caiado, 2006 propone una medida basada en la función de autocorrelación parcial (PACF) estimada  $\hat{\gamma}_{xr} = (\hat{\gamma}_1, \dots, \hat{\gamma}_r)$ . La métrica  $d_{PACF}$  está dada por

$$d_{PACF}(x_t, y_t) = \sqrt{(\hat{\gamma}_{xr} - \hat{\gamma}_{yr})^T W (\hat{\gamma}_{xr} - \hat{\gamma}_{yr})},$$

donde  $W$  también es una matriz de pesos.

#### 2.2.7. Métodos de *clustering* con base en Modelos

Las medidas de disimilaridad con base en modelos suponen que los modelos se generan con ciertas estructuras paramétricas. Lo más común actualmente es suponer procesos ARIMA invertibles. Luego, ajustamos un modelo ARIMA a cada serie y medimos la disimilaridad entre los modelos ajustados. La estructura se supone conocida o se estima utilizando algún criterio de selección de modelos (e.g. AIC, BIC). La estimación suele ser por mínimos cuadrados.

#### Distancia de Piccolo

Piccolo, 1990, define una métrica para la clase de modelos SARIMA invertibles. Sea  $z_t$  un proceso  $ARIMA(p, d, q)(P, D, Q)_s$  invertible definido por

$$\varphi(B)\nabla^d\nabla_s^D z_t = \nu(B)e_t,$$

donde  $e_t \sim WN(0, \sigma^2)$ ,  $B$  es el operador *backshift* o *lag* tal que  $B^k z_t = z_{t-k}$ , los polinomio  $\varphi(B) = \phi(B)\Phi(B^s) = (1 - \phi_1 B - \dots - \phi_p B^p)(1 - \Phi_1 B^s - \dots - \Phi_p B^{sP})$  y  $\nu(B) = \phi(B)\Phi(B^s) = (1 - \theta_1 B - \dots - \theta_p B^p)(1 - \Theta_1 B^s - \dots - \Theta_p B^{sP})$ , para  $s \geq 0$ , no tienen factores comunes, y todas las raíces de  $\varphi(B)\nu(B) = 0$  yacen fuera del círculo unitario. Encima, se supone que los valores atípicos y el componente determinístico se ha eliminado previamente (Maharaj y col., 2019; Montero y Vilar, 2014).

La invertibilidad nos garantiza que  $Z_t$  se puede representar en términos de sus valores pasados con una representación  $AR(\infty)$  dada por  $\pi(B)z_t = e_t$ , donde  $\pi(B) = (1-B)^d(1-B^s)^D\varphi(B)\nu^{-1}(B) = 1 - \sum_{j=1}^{\infty} \pi_j B^j$  y  $\sum_{j=1}^{\infty} |\pi_j| < \infty$ . Así, se introduce la distancia de Piccolo que busca medir la disimilaridad estructural de dos procesos ARIMA  $x_t, y_t$  con ordenes dados. Suponiendo unos  $\pi$ -pesos, la distancia se define como la distancia Euclídea entre los  $\pi$ -pesos estimados,

$$d_{PIC} = \sqrt{\sum_{j=1}^{\infty} (\hat{\pi}_{j,x} - \hat{\pi}_{j,y})^2}.$$

También, Piccolo, 1990, deriva la distribución asintótica y una aproximación computacionalmente rápida. Posteriormente, utiliza esta medida para un problema de clasificación.

#### Distancia LPC Cepstrum

Kalpakis y col., 2001, proponen utilizar el *cepstrum* para el *clustering* de ST ARIMA. El *cepstrum* se define como la transformada de Fourier inversa del espectro de amplitud en el corto plazo. Más precisamente, utilizamos el LPC cepstrum qué se construye utilizando los coeficientes de autoregresión de la codificación lineal predictiva de la señal.

Considerando una ST  $x_t$  con estructura  $AR(p)$ ,

$$x_t = \sum_{r=1}^p \phi_r X_{t-r} + e_t,$$

donde  $\phi_r$  son los coeficientes de autoregresión y  $e_t$  es un proceso de media cero y varianza constante positiva. Entonces, los **coeficientes LPC cepstrum** se derivan de los coeficientes autoregresivos  $\phi_r$  y están dados por

$$\eta_h = \begin{cases} \phi_1 & \text{si } h = 1 \\ \phi_h + \sum_{m=1}^{h-1} (\phi_m - \eta_{h-m}) & \text{si } 1 < h \leq p \\ \sum_{m=1}^p \left(1 - \frac{m}{h}\right) \phi_m \eta_{h-m} & \text{si } p < h. \end{cases}$$

Para medir la distancia entre dos ST  $x_t, y_t$ , se considera la distancia Euclídea entre los coeficientes LPC cepstrum estimados

$$d_{LPC.Cep}(x_t, y_t) = \sqrt{\sum_{i=1}^T (\hat{\eta}_{i,x_t} - \hat{\eta}_{i,y_t})^2}.$$

La idea es que los coeficientes LPC cepstrum mantienen un gran grado de información sobre el proceso ARIMA subyacente y nos permiten discriminar entre procesos ARIMA (Sardá-Espinosa, 2018).

#### 2.2.8. Fuzzy clustering

Los procedimientos de *clustering* anteriores se conocen como particiones duras, aunque el *clustering* jerárquico solo tiene una partición dura cuando se corta el dendrograma en una altura fija. En las **particiones duras**, cada miembro de los datos pertenece exclusivamente a un clúster y se obtiene una partición exhaustiva, disjunta a pares, y con clústers no vacíos. La manera dura de asignar la pertenencia a los clústers puede no ser deseable, por ejemplo cuando se tienen dos objetos igual de distantes a varios centroides de clústers. Tradicionalmente, el *clustering* obliga arbitrariamente la asignación de objetos ‘ambiguos’ a uno de los grupos, aunque se podría decir que pertenecen de manera casi igual a todos. El **fuzzy clustering** relaja el requisito de que los objetos se tienen que asignar a sólo un grupo (Maharaj y col., 2019; Sardá-Espinosa, 2018).

Para ilustrar cómo funciona el fuzzy *clustering*, supongamos que tenemos datos de  $N$  series de tiempo y se tienen  $k$  clústers. Para cada serie, el grado de pertenencia se normaliza tal que la suma es 1 sobre todos los clústers. Así, se puede crear una matriz de dimensión  $N \times k$  que se denomina **matriz de pertenencia**, donde todos los renglones suman 1 (Sardá-Espinosa, 2018). Esta asignación gradual puede reflejar la estructura de los grupos en una manera más natural, especialmente cuando los clústers se sobreponen. Los métodos fuzzy son apropiados para las siguientes consideraciones (Maharaj y col., 2019):

- Es un procedimiento libre de distribuciones.
- Debido a la dificultad de identificar una frontera clara entre los clústers.
- Es eficiente computacionalmente ya que los cambios dramáticos en la asignación de clústers son menos probables al momento de la estimación. También se sabe que es menos afectado por problemas de óptimos locales.
- Permite descubrir si un objeto pertenece a un segundo clúster, con un grado de pertenencia muy cercano al primer clúster.
- Mayor sensibilidad para capturar detalles importantes en las series de tiempo.

Además, se puede pasar de un agrupamiento fuzzy a un agrupamiento duro al tomar el máximo por renglón de la matriz de pertenencia (Sardá-Espinosa, 2018).

### Fuzzy k-medoids

Al utilizar **fuzzy k-medoids** como función de centroides, los centroides se seleccionan al resolver

$$\begin{aligned} \mu_k &= x_q \\ q &= \arg \min \sum_{p=1}^N u_{p,k}^m d_{p,k}, \\ \text{donde } \sum_{k=1}^K u_{p,k} &= 1, \quad u_{p,k} \geq 0, \end{aligned}$$

además que  $d_{p,k}$  representa la distancia entre el  $p$ -ésimo miembro de los datos y el  $k$ -ésimo centroide fuzzy. En este algoritmo, el centroide pertenece a los datos originales, y se pueden considerar series de distinta longitud (siempre que la función de distancia pueda aplicarse a series de distinta longitud). La función objetivo se minimiza de manera iterada al resolver la ecuación

$$u_{p,k} = \frac{1}{d_{p,k}^{\frac{2}{m-1}} \sum_{q=1}^K \left( \frac{1}{d_{p,k}} \right)^{\frac{2}{m-1}}},$$

hasta la convergencia.

## 2.3. Prototipos de Series de Tiempo

La otra parte importante en la tarea de *clustering* de series de tiempo es obtener lo que se conoce como el prototipo de las series de tiempo. Se espera que las series dentro de un clúster sean parecidas y puede ser de interés construir una ST que resuma las características más importantes de todas las series dentro del grupo, esto es lo que se conoce como obtener prototipos, centroides o promediar series (Sardá-Espinosa, 2018).

### 2.3.1. Media y Mediana

Una propuesta simple y muy útil es la media aritmética. Suponiendo series de tiempo con la misma frecuencia de muestreo y total de observaciones, tomamos el promedio de cada valor de las series dentro del clúster en  $t$  y repetimos para todo  $t = 1, \dots, T$ . Más precisamente, sea  $x_{c,t}$  el  $t$ -ésimo elemento de serie  $c$  dentro del clúster  $C$ ,

$$\mu_t = \frac{1}{N} \sum_c x_{c,t}, \quad \text{para todo } t = 1, \dots, T; \quad c \in C,$$

y donde  $N$  es el número de series en el clúster. Aunque, también se puede utilizar la mediana muestral en vez de la media (Sardá-Espinosa, 2018).

### 2.3.2. Partición Alrededor de los Medoids (PAM)

Un **medoid** es un objeto representativo de un clúster, cuya distancia promedio a todos los objetos dentro del mismo clúster es mínima. Los medoids son elementos en la muestra y por lo tanto la PAM es preferida a la media o mediana para no alterar la estructura de la serie de tiempo.

### 2.3.3. Extracción de *Shapes*

Un método para hacerse con un prototipo de las series se llama **extracción de shapes** y es parte del algoritmo *k*-Shape propuesto por Paparrizos y Gravano, 2015. El algoritmo depende de la NCCc y se utiliza para ligar dos series de manera óptima. Para el centroide de la serie, usualmente se utiliza uno al azar de los datos, excepto cuando todas las series en consideración tienen la misma longitud en tal caso no se necesita un centroide. La serie que resulta del algoritmo debe estar *z*-estandarizada al igual que las series de entrada.

### 3. Clustering de Acciones del NASDAQ

El NASDAQ (del inglés *National Association of Securities Dealers Automated Quotation*) es una bolsa de valores donde cotizan empresas tecnológicas. Es la segunda bolsa de valores automatizada más grande de EE.UU y la bolsa que tiene mayor volumen de intercambio por hora en el mundo. Utilizamos los activos en el índice NASDAQ 100 que incluye los 100 valores de las empresas más grandes en la industria tecnológica, dado que estén listadas en el NASDAQ. Precisamente, utilizamos los valores listados en el NASDAQ-100 al 9 de Mayo de 2022. Recolectamos los datos del precio al cierre del día de intercambio desde el inicio de 2021 hasta el 20 de Mayo de 2022, para un total de  $T = 348$  observaciones de 101 activos. Ilustramos el reto de visualización de series de tiempo multivariadas cuando se tienen muchas series, y hacemos uso el m-TSNE para poder visualizar series de tiempo altamente dimensionales. Posteriormente, ilustramos la tubería de análisis convencional de series de tiempo para agrupar los valores en el NASDAQ 100. Utilizamos la mayoría de las distancias y métodos de *clustering* presentados anteriormente y evaluamos algunas diferencias.

#### 3.1. Visualizando Series de Tiempo Multivariadas

El desarrollo tecnológico de las sociedades post-industriales genera cantidades masivas de información hoy en día. Un reto común y actual en el análisis de ST es la gran dimensión de los datos. Incluso la visualización de múltiples ST se complica cuando se tienen muchas series, y este problema se acentúa cuando se tienen ST en forma de multicampos. En la Figura 2 se visualizan 3, 6, 15 y todos los valores en el NASDAQ 100. Dentro de la Figura 2, donde se tienen 3 valores se puede apreciar la similaridad en la forma de los valores de AMD y NVDA, en contraste con INTC. Sin embargo, al aumentar el número de series de tiempo en la visualización, se satura de información y no es fácil realizar la exploración.

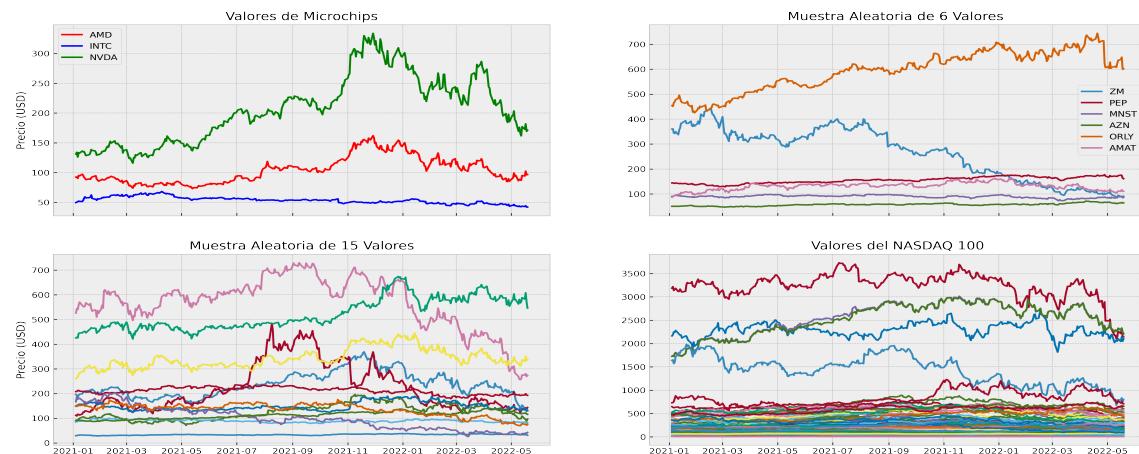


Figura 2

El método propuesto por van der Maaten y Hinton, 2008, conocido como *t-SNE*, permite visualizar datos altamente dimensionales, pero supone realizaciones independientes. Sin embargo, el **m-TSNE** de Nguyen y col., 2017, (m-TSNE del inglés *multivariate Time Series t-distributed Stochastic Neighbor Embedding*). En el m-TSNE, primero obtiene la similaridad entre cada objeto utilizando una norma de Frobenius extendida y es común utilizar algún tipo de preprocesamiento. Luego, se computa la proyección 2D ó 3D. Este método es capaz de descubrir grupos en los datos, pero también uno puede sobre-interpretar un gráfico<sup>3</sup>. El *t-SNE* se ha utilizado frecuentemente en problemas de clasificación dentro del área de visión computacional. Por lo tanto,

<sup>3</sup><https://distill.pub/2016/misread-tsne/>

el m-TSNE es una generalización del t-SNE que permite incorporar situaciones que se presentan en el análisis de ST y es una herramienta para visualizar series de tiempo en un espacio de dimensión menor.

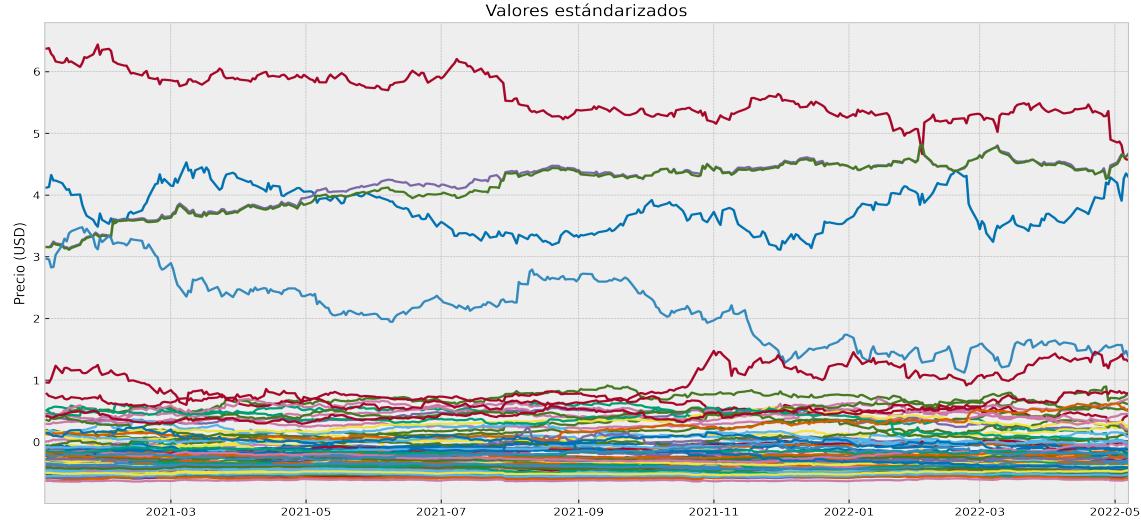


Figura 3

Primero, se estandarizaron los precios de los valores tal que su media fuera cero y tuviesen desviación estándar <sup>4</sup>, tales valores se pueden observar en la Figura 3. Luego, se aplicó t-SNE y m-TSNE. El resultado del t-SNE se puede ver en la Figura 4 y en la Figura 5 está el resultado del m-TSNE y en ambas figuras se puede apreciar la formación de grupos en los precios de los valores del Nasdaq-100 dentro de nuestro rango de observación.

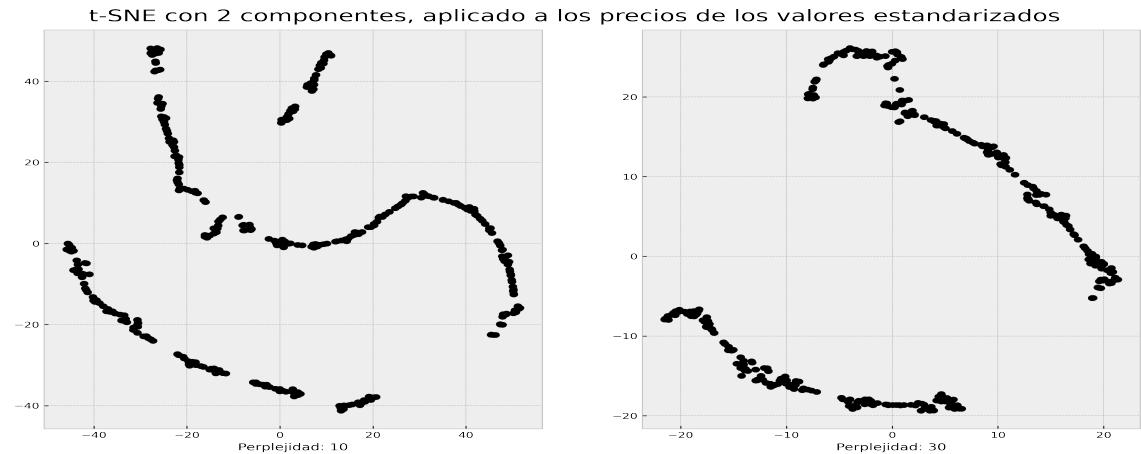


Figura 4

---

<sup>4</sup>[https://tslearn.readthedocs.io/en/stable/gen\\_modules/preprocessing/tslearn.preprocessing.TimeSeriesScalerMeanVariance.html](https://tslearn.readthedocs.io/en/stable/gen_modules/preprocessing/tslearn.preprocessing.TimeSeriesScalerMeanVariance.html)

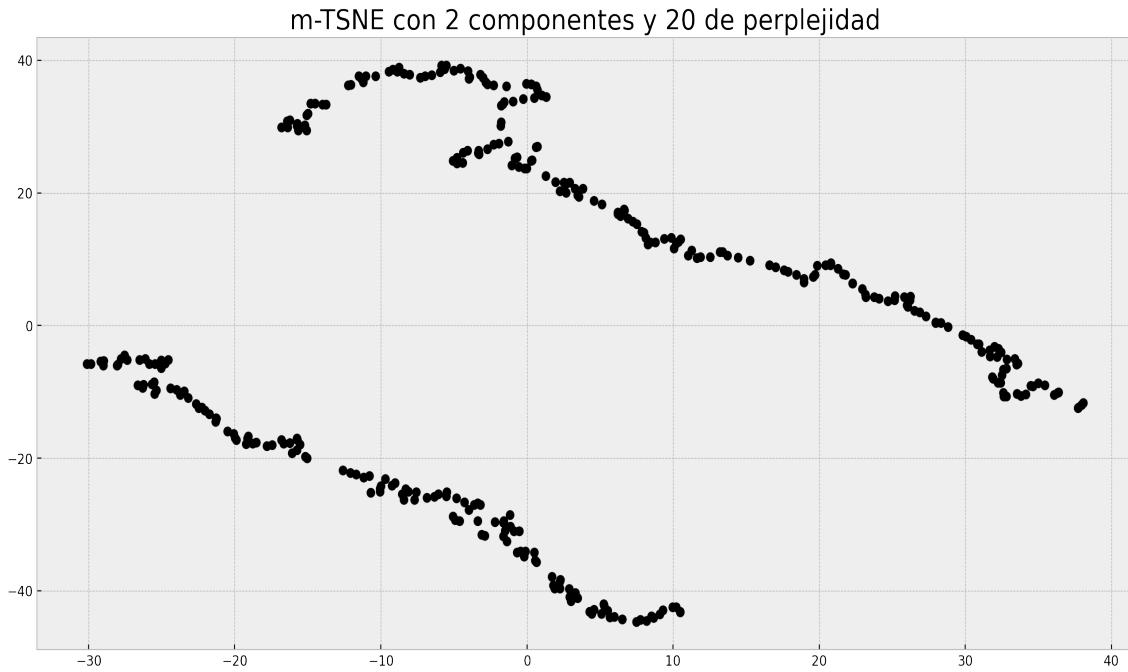


Figura 5: m-TSNE aplicado a los precios de los valores estandarizados.

### 3.2. *Clustering* Jerárquico

#### *Clustering* Jerárquico con Distancia Euclídea y DTW

Para ilustrar la tarea de *clustering* con los valores del NASDAQ-100 como series de tiempo iniciamos con el *clustering* más simple que es el *clustering* jerárquico con distancia euclídea. En la Figura 6 tenemos los resultados del *clustering* jerárquico con *complete linkage* y distancia euclídea. Al partir el dendrograma a una altura apropiada se pueden obtener un número de clústers  $K \leq M$ , donde  $M = 101$  es el número de ST en la muestra. Para obtener  $K = 4$  clústers, el dendrograma se puede cortar a una altura de 20,000 y resulta la partición presente en las figuras 7 y 8, donde se puede apreciar que, al utilizar esta técnica de *clustering* muy simple, se agrupan los valores por niveles de precio al cierre dados por: menos de 1000 USD, de 1000 a 1800 USD; de 2000 a 3000 USD; más de 3000 USD.

Ya que las series tienen todas la misma longitud, el DTW tiene un *warp* simple y el resultado del *clustering* jerárquico con DTW termina siendo muy similar a los grupos formados con la distancia Euclídea tal y como se puede apreciar en ambos dendrograma en las figuras 6 y 9. Al obtener una partición con  $K = 4$  clústers, podemos apreciar que se forman varios niveles de precios: menos de 450 USD, de 450 a 1200 USD; de 800 a 1900 USD; más de 1900 USD, tal y como se puede apreciar en las figuras 10 y 11.

Clústering Jerárquico con Distancia Euclídea de los valores en el NASDAQ-100

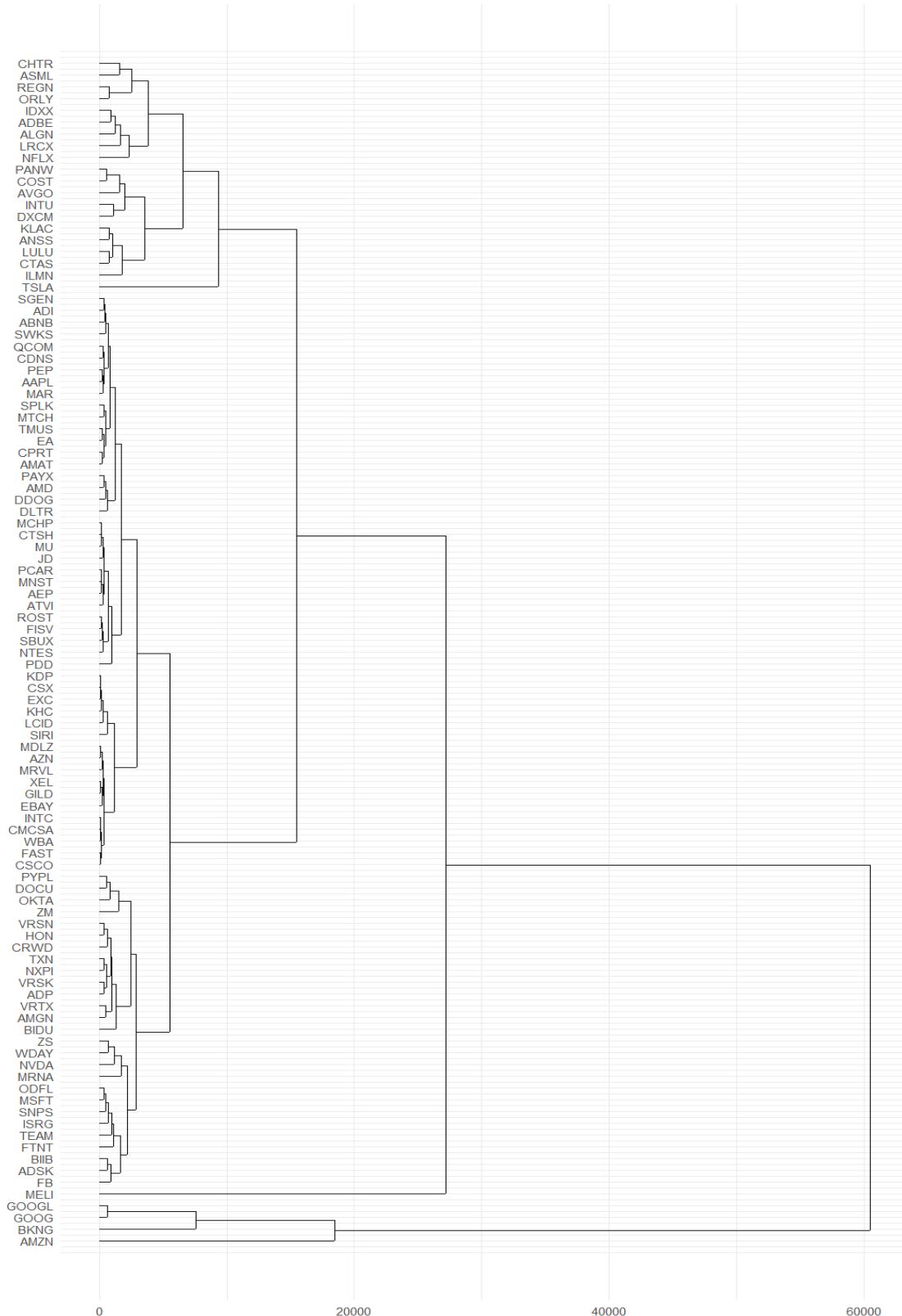


Figura 6: Dendrograma para *clustering* Jerárquico con Distancia Euclídea.

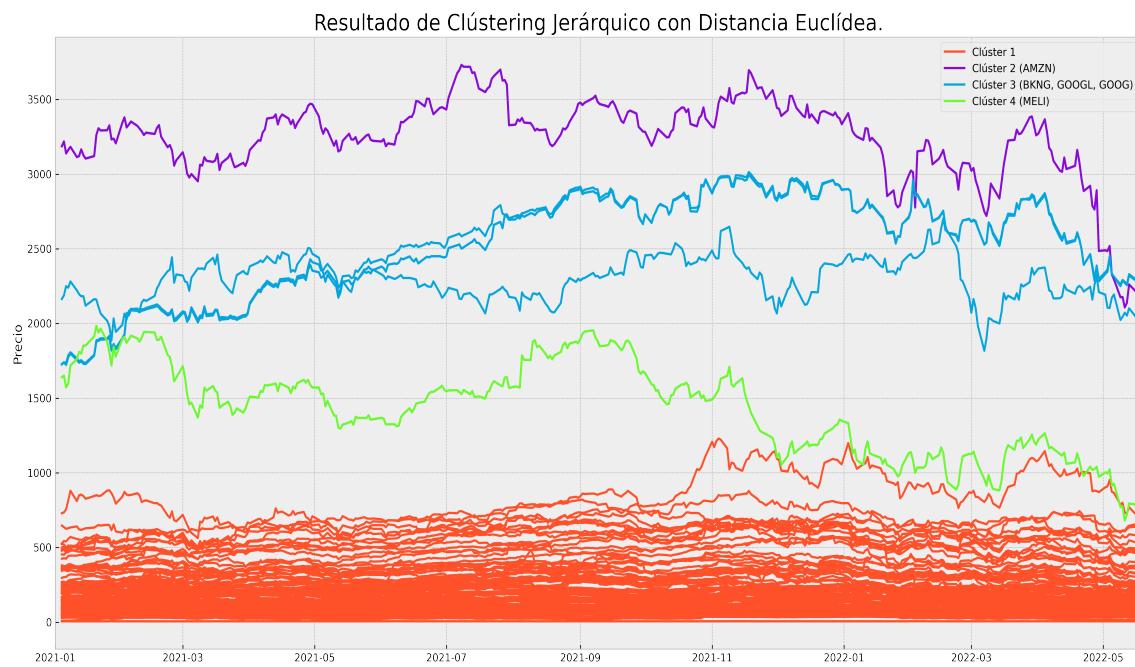


Figura 7



Figura 8

Clústering Jerárquico con Distancia DTW de los valores en el NASDAQ-100

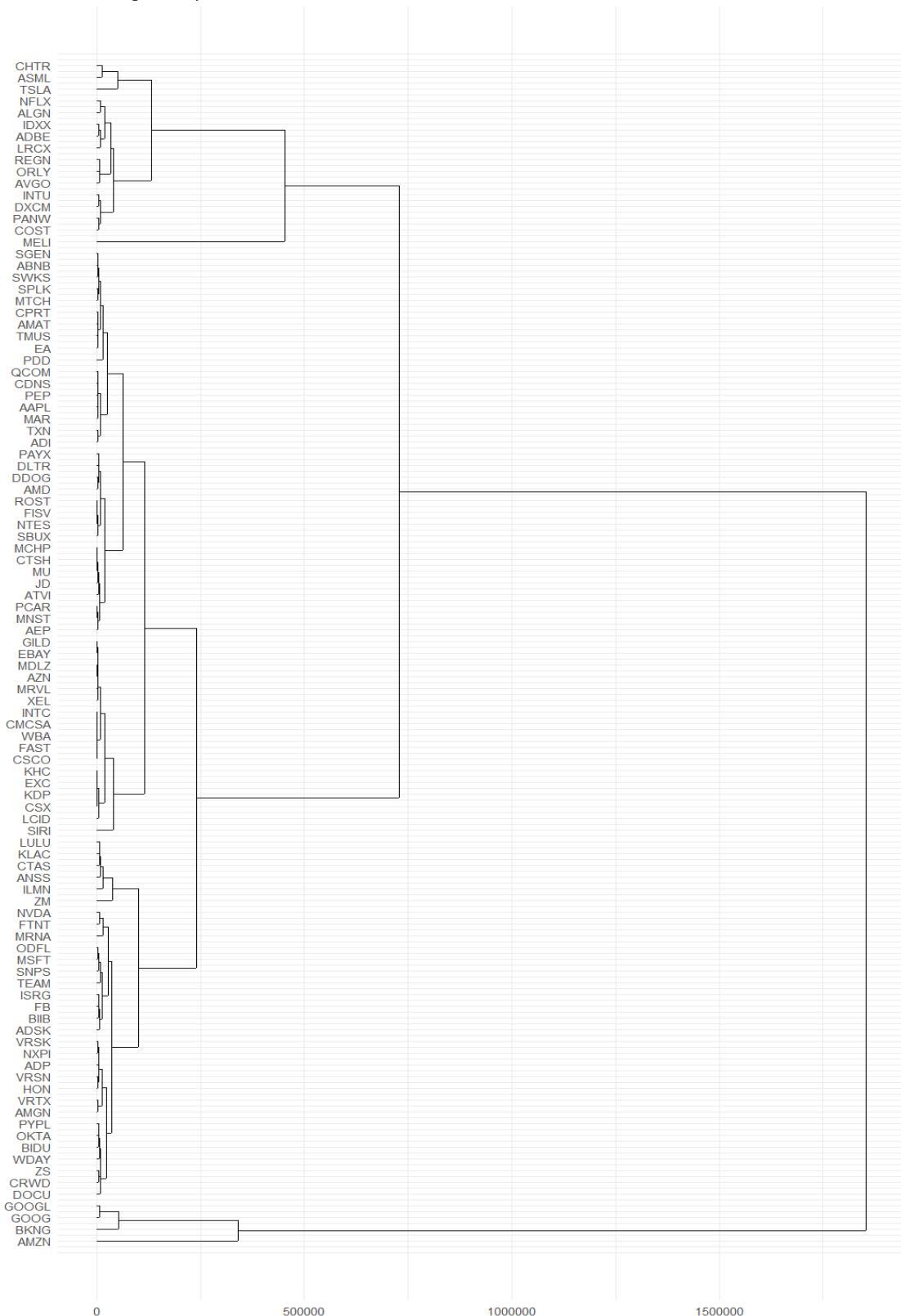


Figura 9: Dendrograma para *clustering* Jerárquico con Distancia DTW.

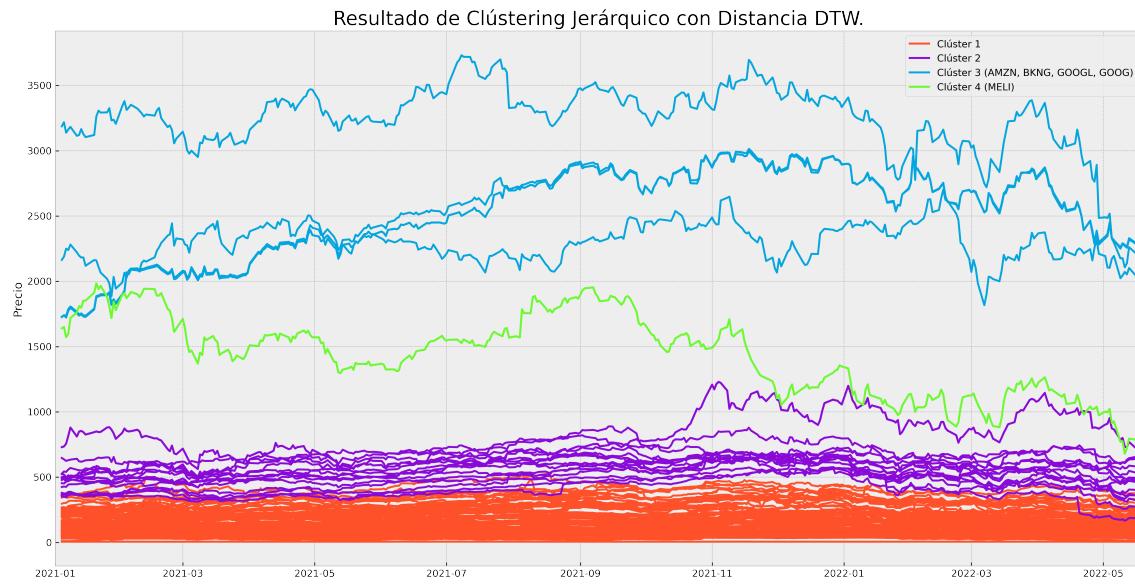


Figura 10

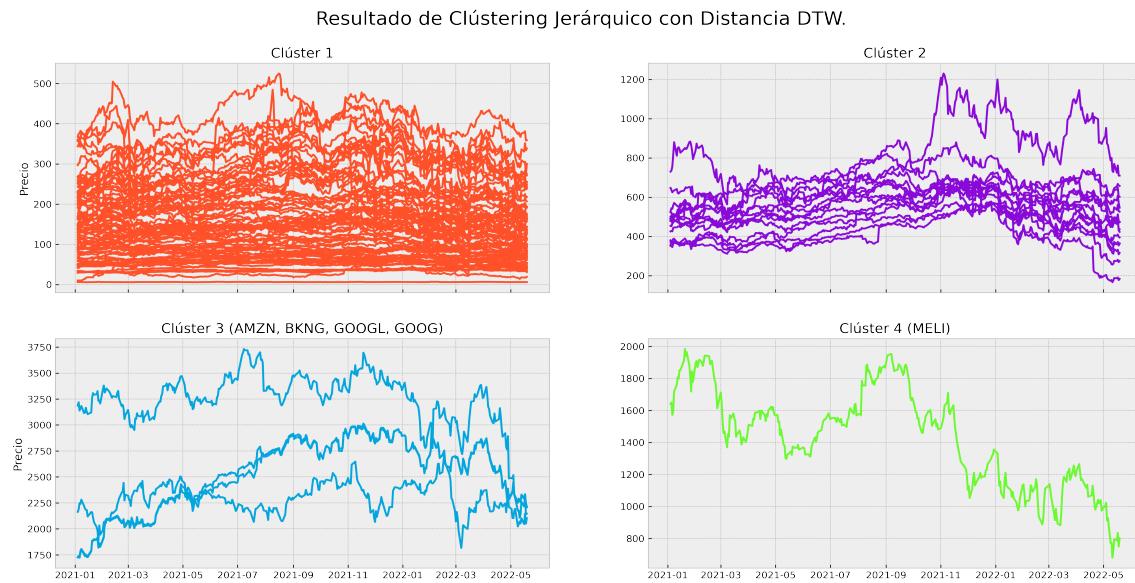


Figura 11

### **Clustering Jerárquico con Distancia ACF y PACF**

Los resultados del *clustering* jerárquico al utilizar distancias ACF y PACF se pueden observar en las figuras 12, 17, 13, 15. Se puede notar que los resultados de las agrupaciones son muy distintos, comparados con *clustering* jerárquico con distancia euclídea y dtw, ya que ahora no se dividen los valores por nivel de precio sino que se agrupan por medio de las funciones de autocorrelación. Sin embargo, no es sorprendente que la mayoría de los precios de los valores estén correlacionados debido a que todos forman parte del mismo índice.

Clústering Jerárquico con Distancia ACF de los valores en el NASDAQ-100

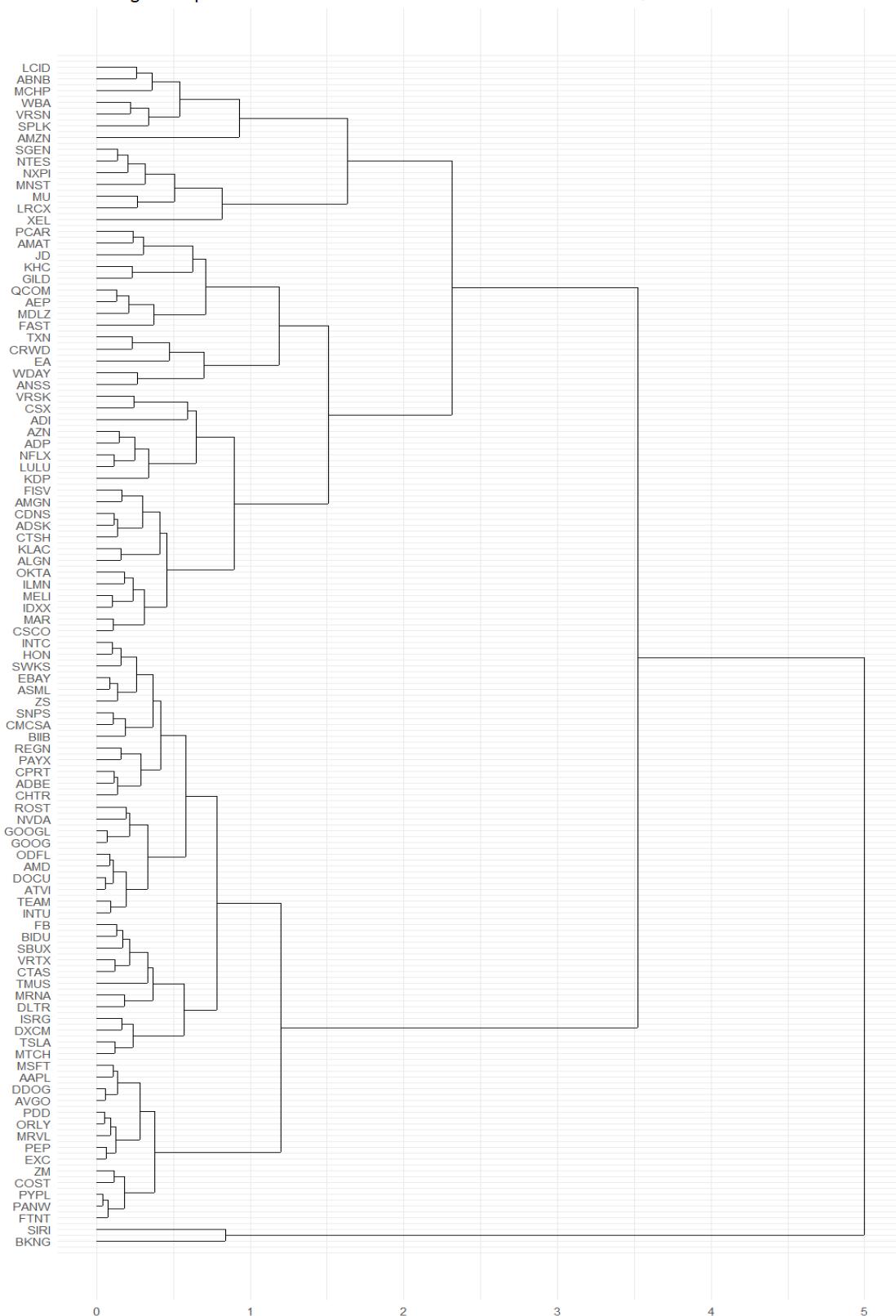


Figura 12: Dendrograma para *clustering* Jerárquico con Distancia ACF.

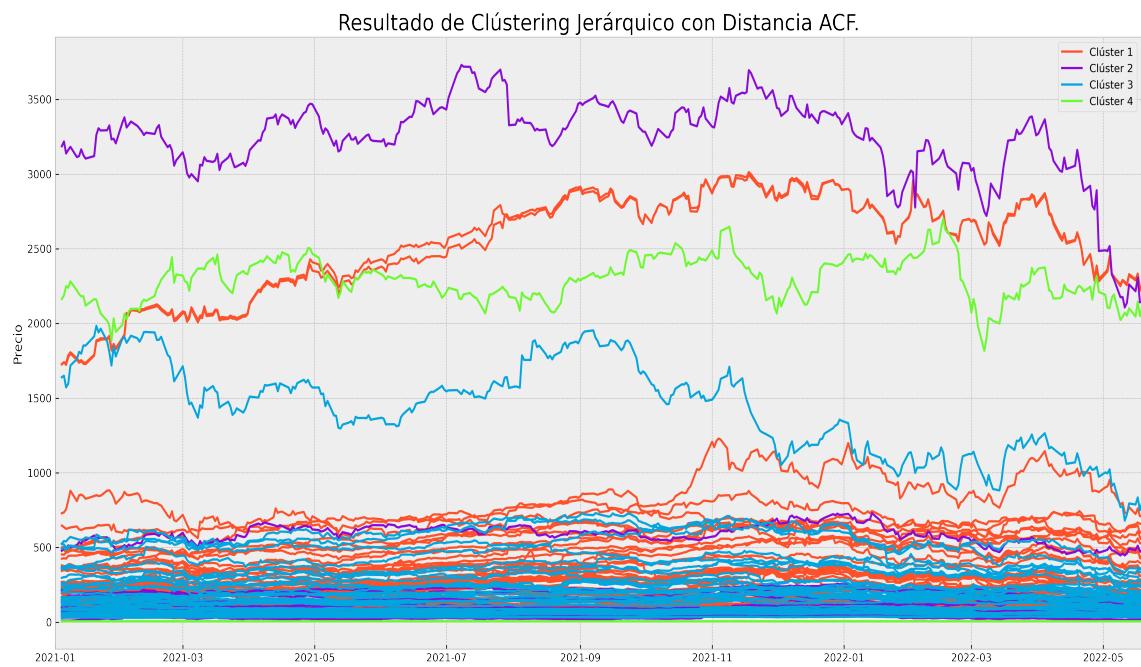


Figura 13

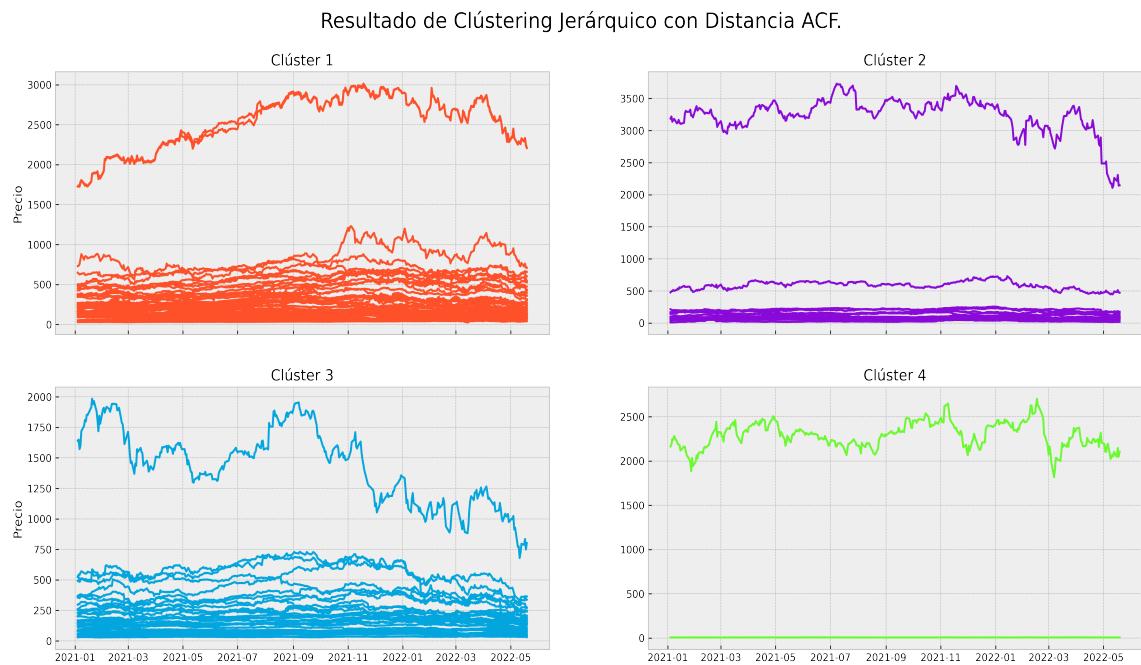


Figura 14

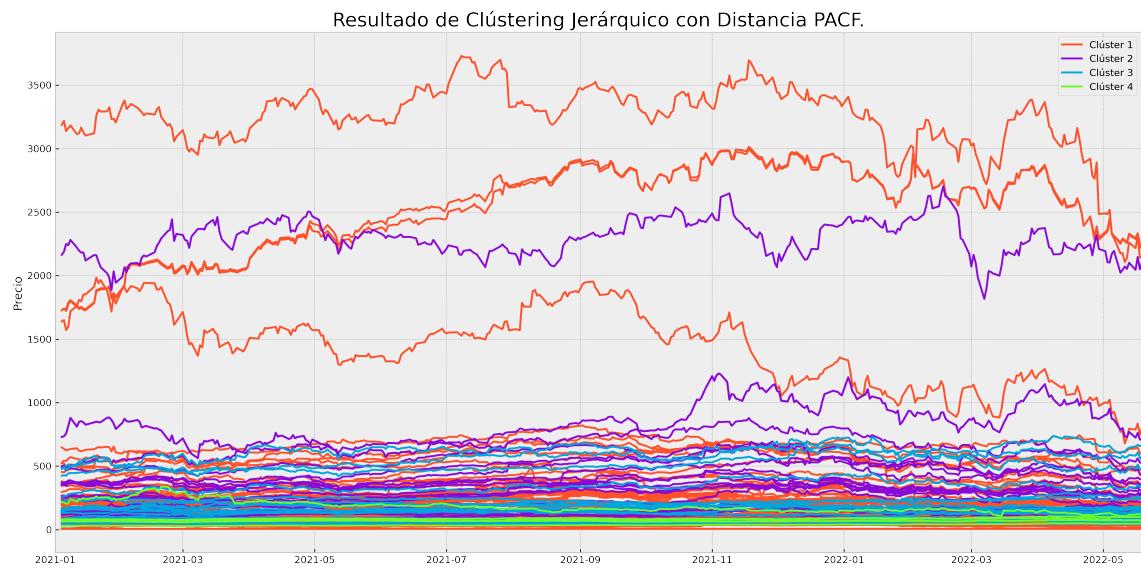


Figura 15

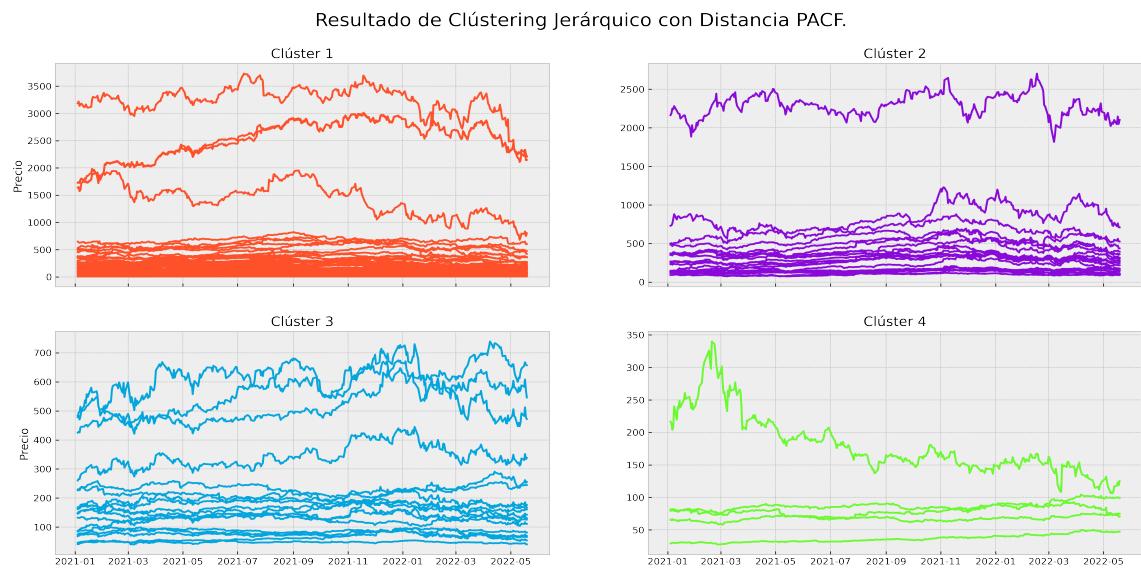


Figura 16

Clústering Jerárquico con Distancia PACF de los valores en el NASDAQ-100

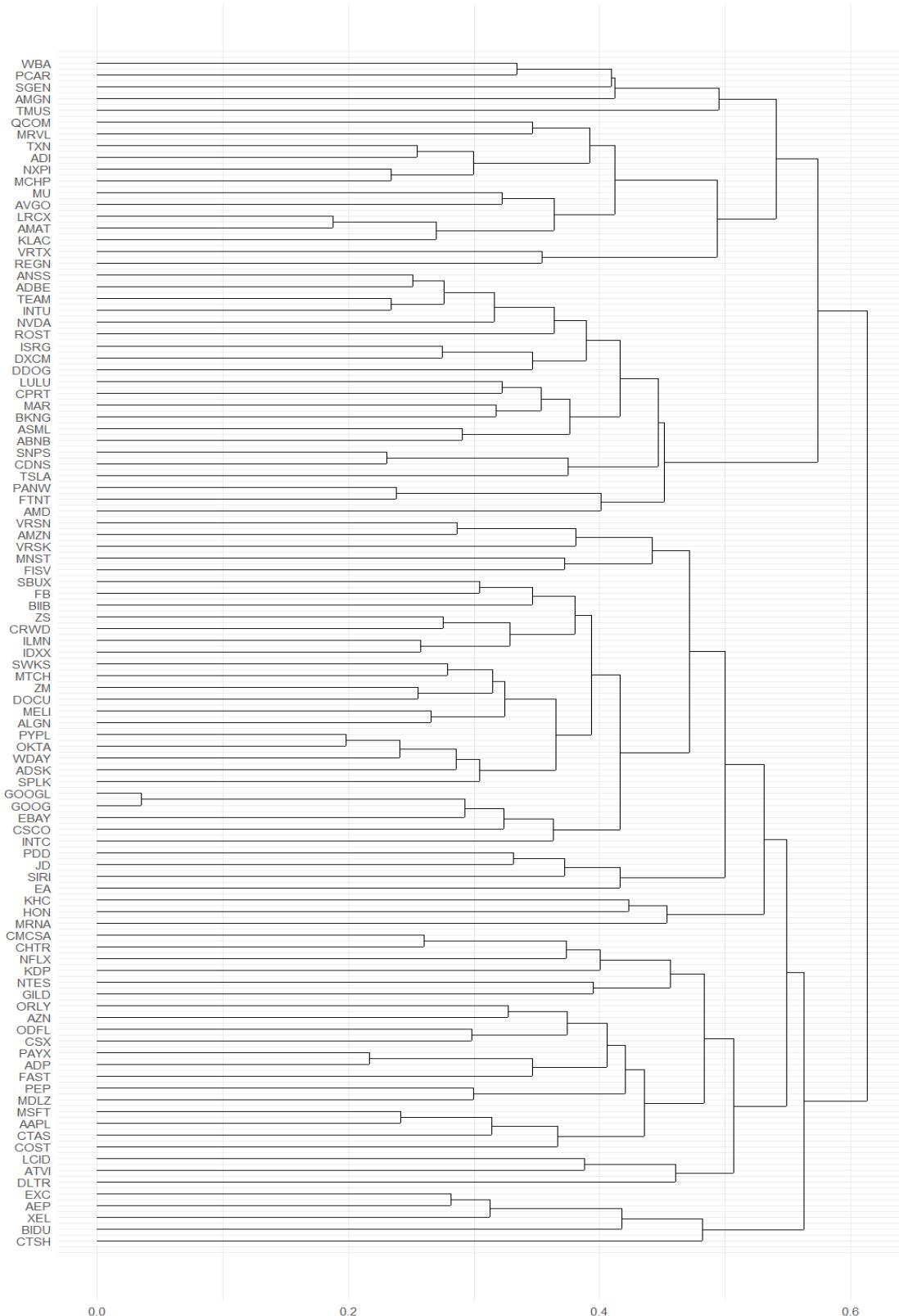


Figura 17: Dendrograma para *clustering* Jerárquico con Distancia PACF.

### Clustering Jerárquico con Distancia de Piccolo y LPC Cepstrum

Debido a la importancia de los modelos ARIMA en el área de finanzas, es natural querer realizar las agrupaciones al utilizar medidas de similaridad con base en estos modelos. En las figuras 20 y 18, notemos que la mayoría de los valores se concentran en el grupo 1 y 2. Asimismo, los clústers 3 y 4 parecen ser procesos no similares ya que forman clústers solitarios.

En las figuras 21 y 19, tenemos la salida del algoritmo de *clustering* jerárquico con base en la distancia LPC Cepstrum. También, podemos apreciar que los valores se agrupan de manera distinta que al usar distancia euclídea o basada en autocorrelaciones y los clústers 3 y 4 parecen ser procesos no similares al seguir formando clústers solitarios.

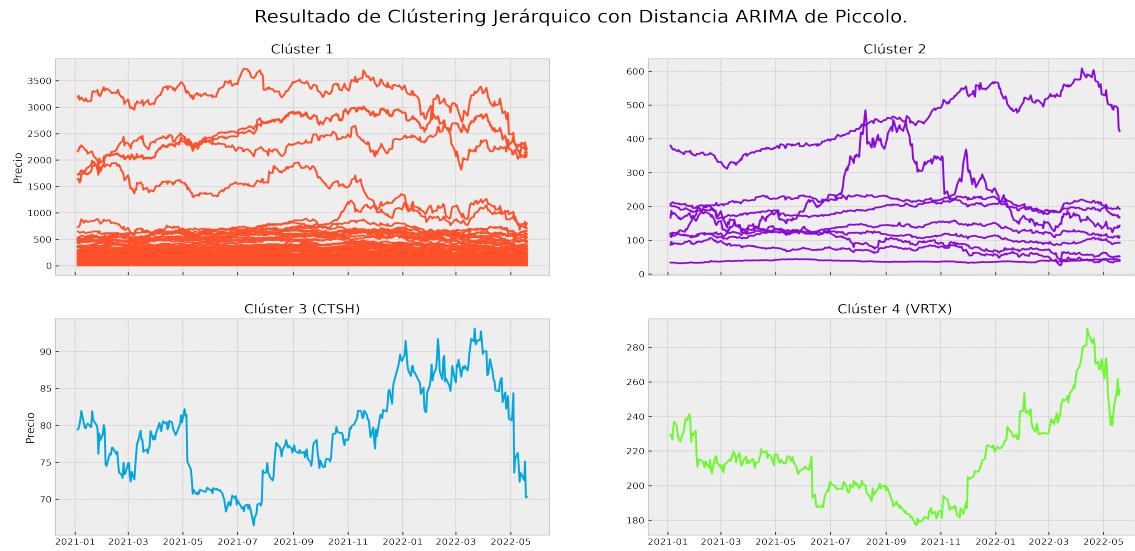


Figura 18

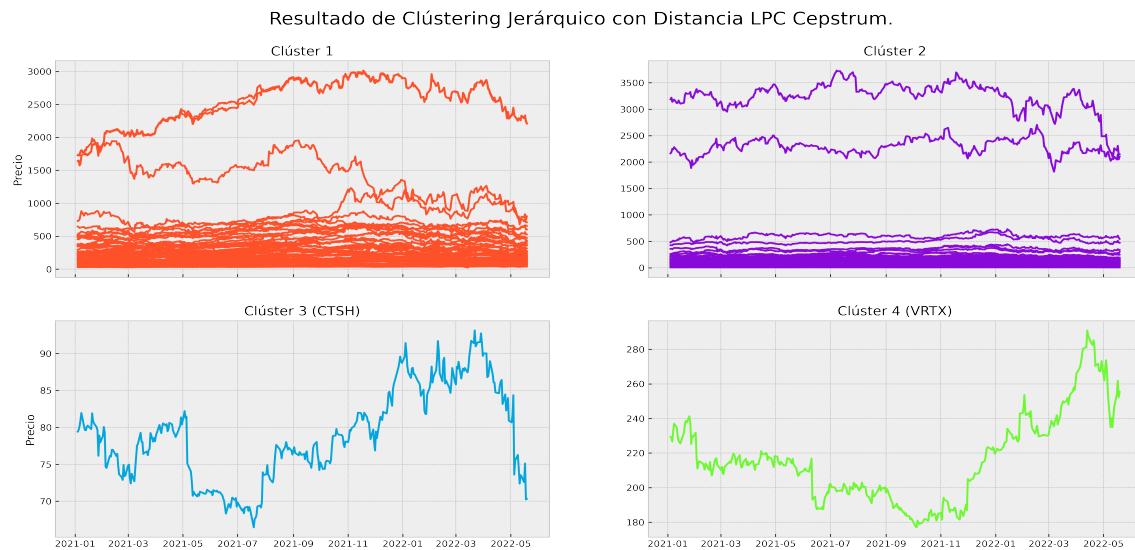


Figura 19

Clústering Jerárquico con Distancia ARIMA de Piccolo de los valores en el NASDAQ-100

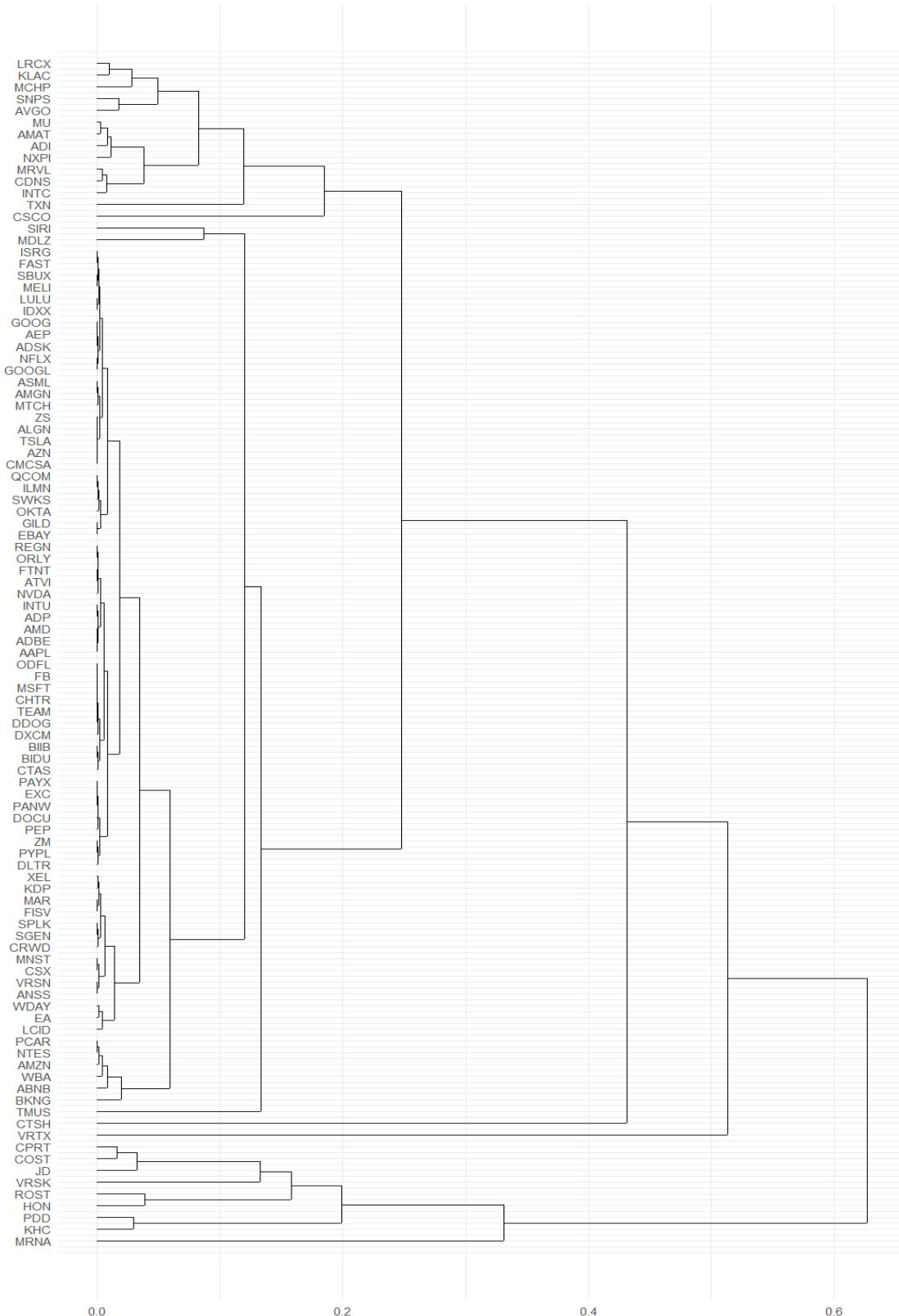


Figura 20  
24

Clústering Jerárquico con Distancia LPC Cepstrum de los valores en el NASDAQ-100

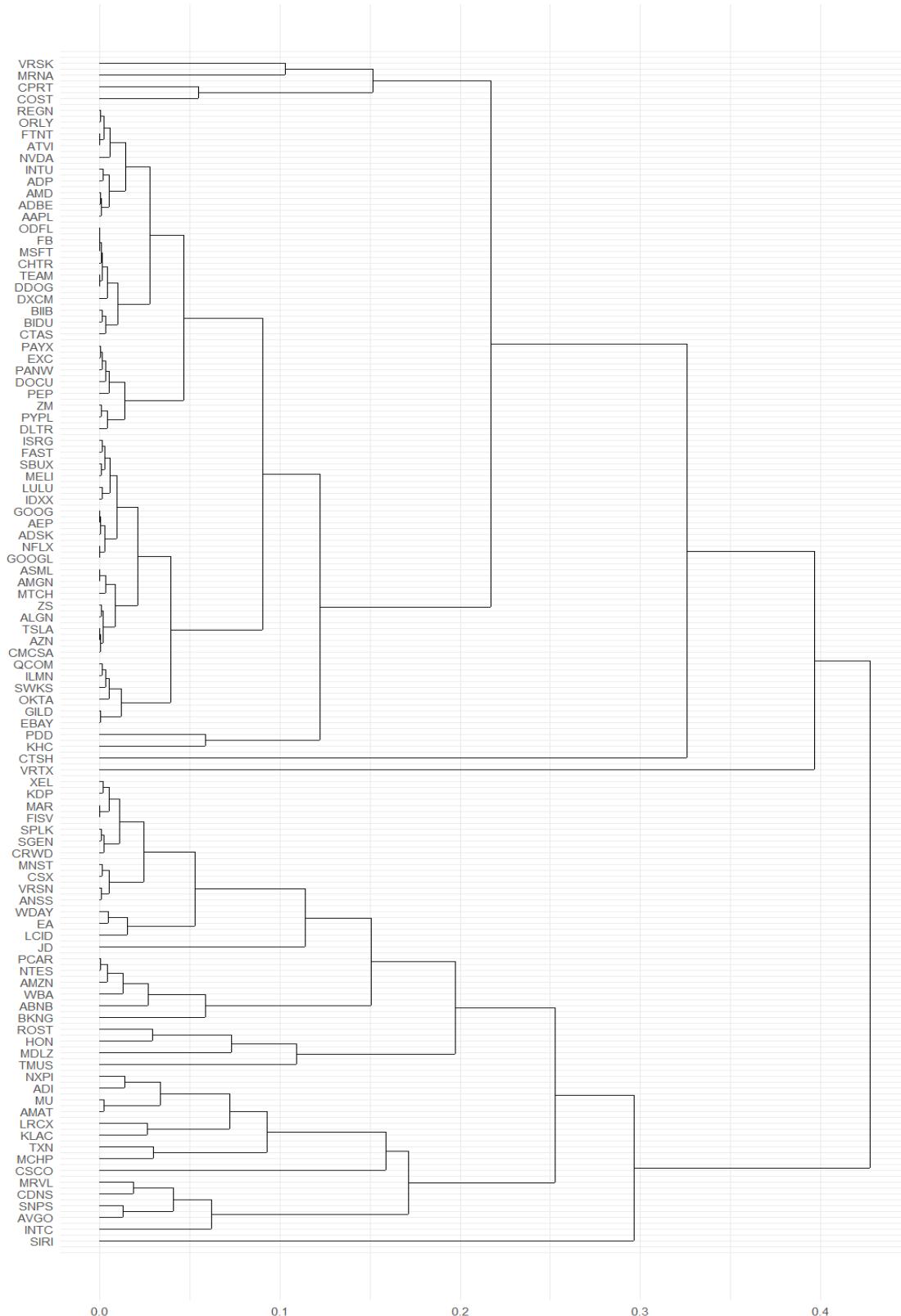


Figura 21  
25

### 3.3. Clustering No Jerárquico

#### 3.3.1. k-medias

Para determinar el número de  $k$  en  $k$ -medias, utilizamos el método del codo que involucra llevar a cabo el *clustering* de  $k$ -medias de manera iterada incrementando  $k$  y graficar el *score* del *clustering* como función de  $k$ . El *score* es una manera de medir la distancia entre los clústers relativo a la distancia dentro de los clústers.

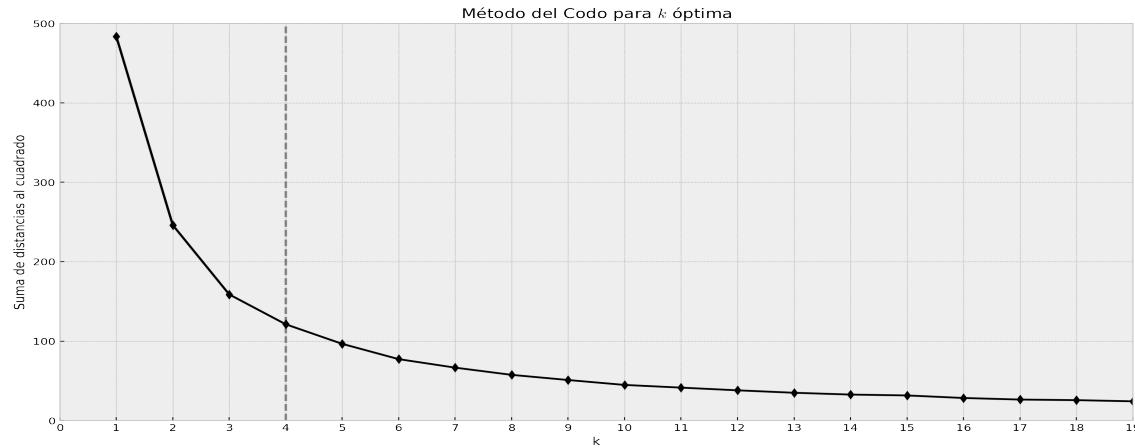


Figura 22

El resultado del *clustering* por  $k$ -medias,  $k = 4$ , se puede apreciar en la Figura 23. Podemos apreciar los 4 niveles de precios distintos establecidos, aproximadamente: menos de 200 USD, de 100 a 400 USD; de 250 a 1000 USD; más de 1000 USD. Es fácil ver que se tiene una partición de los valores distinta que cuándo se utilizó la distancia euclídea en el contexto de agrupación jerárquica. Además, hay bastante sobreposición de las series por lo que podría ser útil un enfoque fuzzy para llevar a cabo las divisiones en niveles de precios.

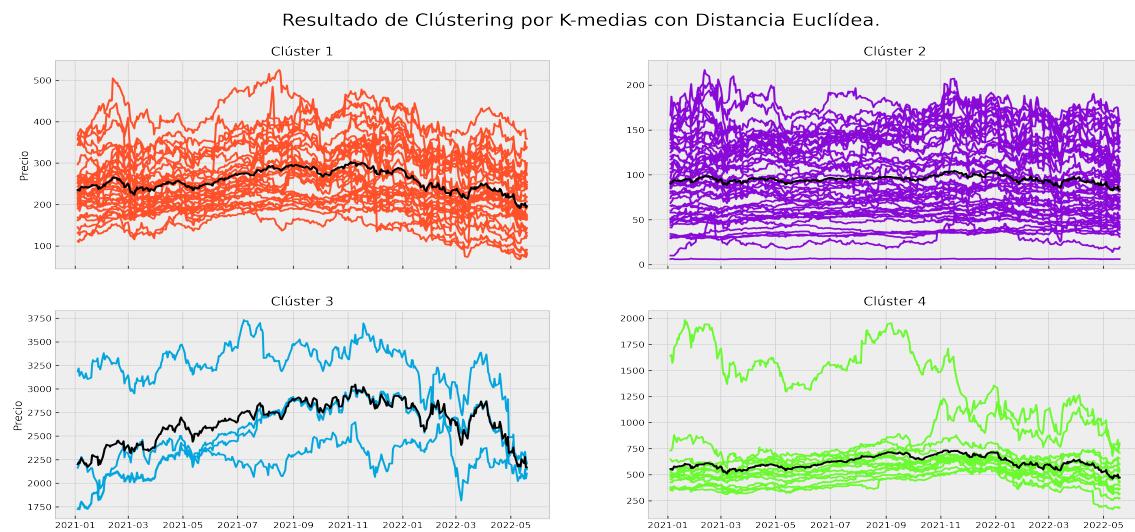


Figura 23: *Clustering* por  $k$ -medias con Distancia Euclídea.

### 3.3.2. PAM

#### *Clustering no Jerárquico con distancia DTW y PAM*

En la salida del *clustering* no Jerárquico con distancia DTW y PAM, que aparece en la Figura 24, se puede ver que el centroide ahora es un elemento de las series y caracterizan un precio promedio de las series dentro del clúster.

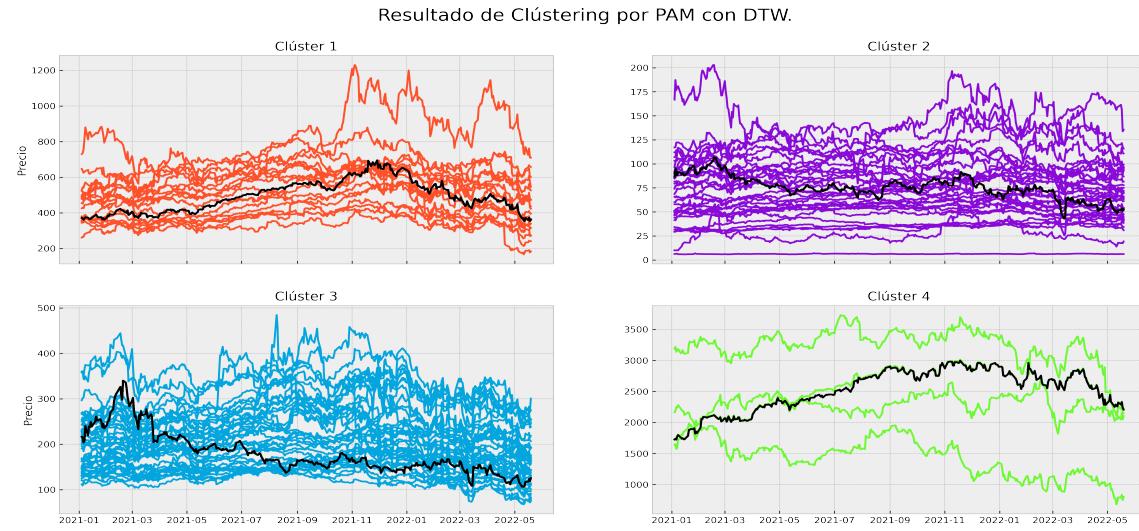


Figura 24: Al utilizar PAM, el prototipo del clúster (la linea negra) es el medoid.

#### *Clustering no Jerárquico con distancias basadas en autocorrelaciones*

Con base en las figuras 25 y 26, podemos ver el *clustering* no jerárquico con distancias basadas en autocorrelaciones. También, se puede apreciar que los centroides que nos podrían permitir seleccionar una ST (o su función de autocorrelación) representativa de cada clúster. En la Figura 25, podemos notar que parece haber 2 ST con una ACF atípica a las otras, particularmente aquellas ST en el clúster 3.

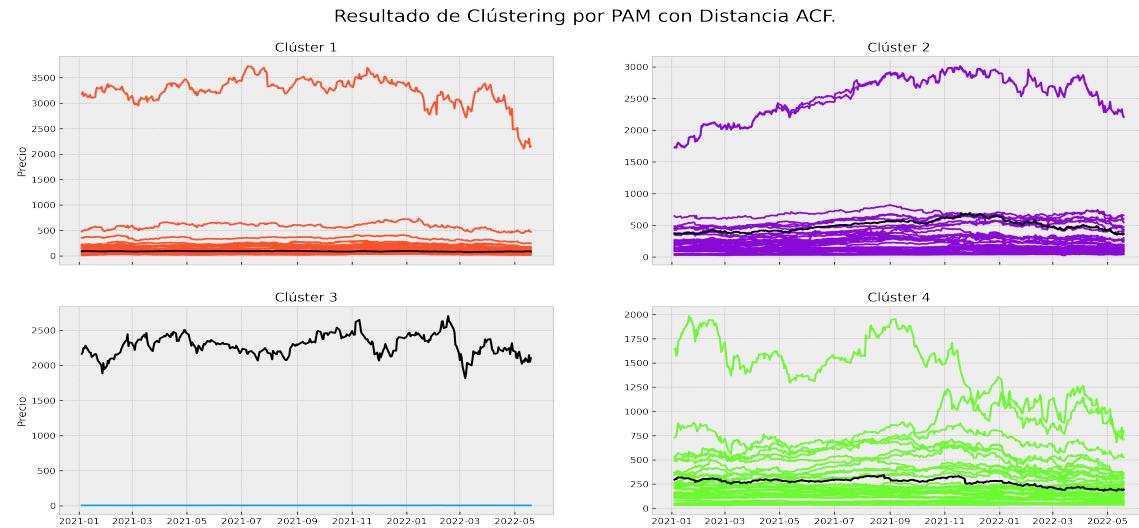


Figura 25: *Clustering* por PAM con Distancia ACF.

Al utilizar la distancia PACF, en la Figura 26, los grupos formados se vuelven más balanceados a comparación con la distancia ACF.

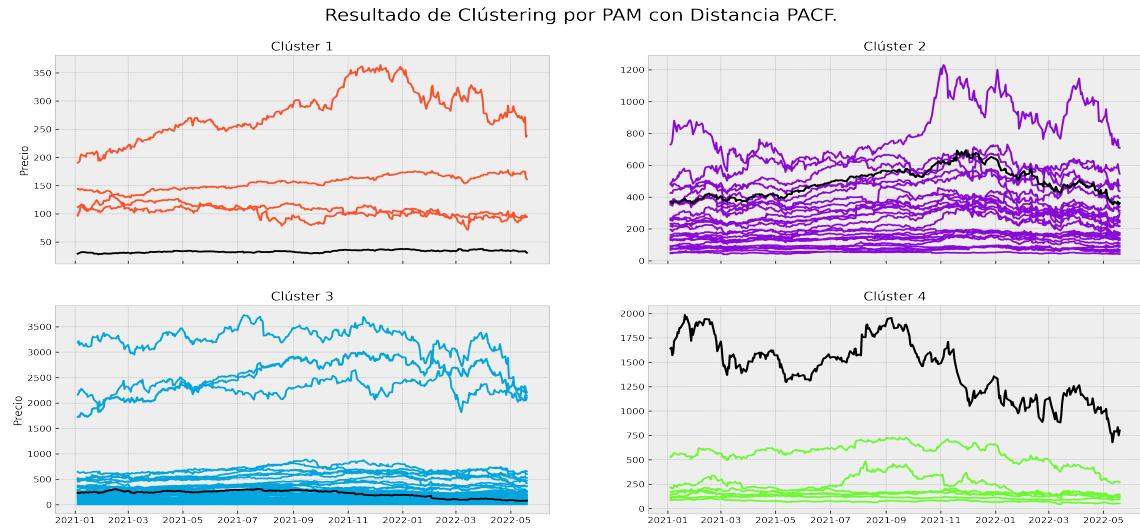


Figura 26: *Clustering* por PAM con Distancia PACF.

### 3.3.3. K-Shape

Para utilizar el *K-Shape* primero estandarizamos el conjunto de valores para que tengan media cero y desviación estándar unitaria. En los clústeres formados que se pueden ver en Figura 27 podemos ver que los valores parecen agruparse en 4 formas: un clúster con forma creciente y una corrección considerable al final del periodo, luego tenemos dos formas de clústeres que comienzan por encima del precio promedio y luego parecen converger al precio promedio. El último clúster parece tener una forma creciente después de permanecer alrededor del precio promedio.

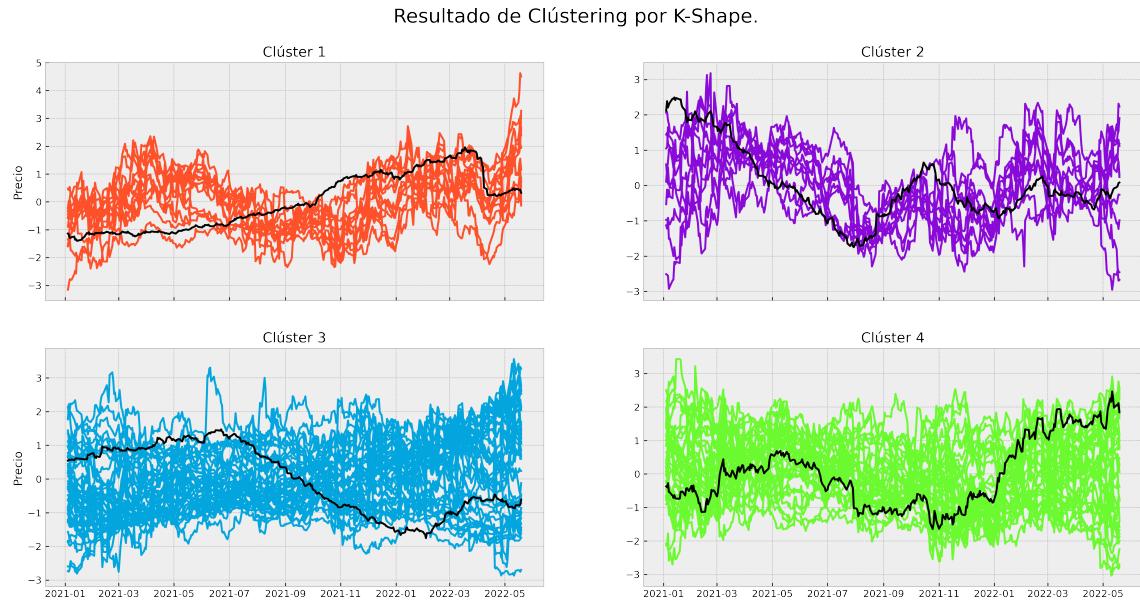


Figura 27: *Clustering* por *K-Shape*.

### 3.4. Fuzzy clustering

Comenzamos nuestros fuzzy clústers con una distancia ACF y  $K = 4$  medoids. En la Tabla 1 tenemos cinco entradas de la matriz de pertenencia. Dentro de los 5 valores en la tabla, no tenemos clústers que parezcan pertenecer a más de 1 grupo, ya que si se marca un clúster al que pertenecen. Una vez que se obtiene la partición dura, se obtiene la partición presente en la Figura 28.

Símbolo del valor	Clúster 1	Clúster 2	Clúster 3	Clúster 4
AAPL	0.07	0.90	0.01	0.02
ABNB	0.03	0.01	0.69	0.27
ADBE	0.281	0.686	0.15	0.018
ADI	0.59	0.13	0.12	0.16
ADSK	0.97	0.021	0.004	0.005

Tabla 1: Cinco entradas de la matriz de pertenencia bajo Fuzzy clustering con Distancia ACF.

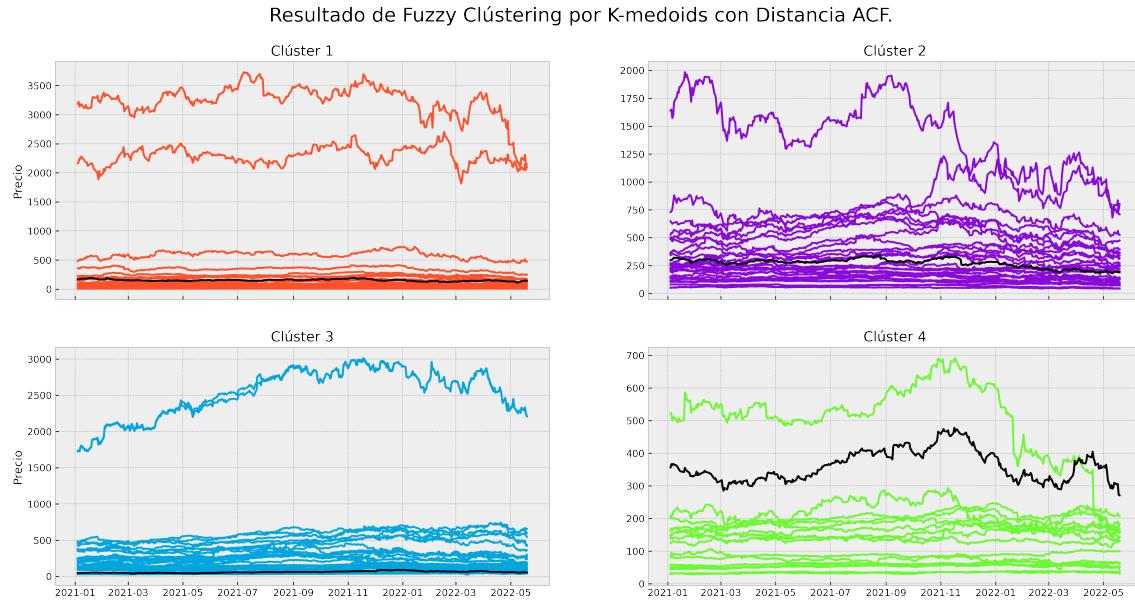


Figura 28

Al realizar el fuzzy clustering con la distancia ARIMA de Piccolo y  $K = 4$  medoids se obtiene una matriz de similaridad de donde se obtienen 5 entradas que están en la Tabla 2. Bajo esta técnica de clustering se han separado los valores de GOOG y GOOGL lo cuál no tiene mucho sentido ya que el precio de ambas acciones es muy cercano, tal y como se puede apreciar en la Figura 29.

Símbolo del valor	Clúster 1	Clúster 2	Clúster 3	Clúster 4
AAPL	0.99	0.001	0.001	0.008
ABNB	0.26	0.003	0.02	0.69
ADBE	0.99	0.0004	0.0006	0.009
ADI	0.003	0.993	0.0007	0.004
ADSK	0.92	0.002	0.008	0.07

Tabla 2: Primeras 5 entradas de la matriz de pertenencia bajo Fuzzy *clustering* con Distancia ARIMA de Piccolo.

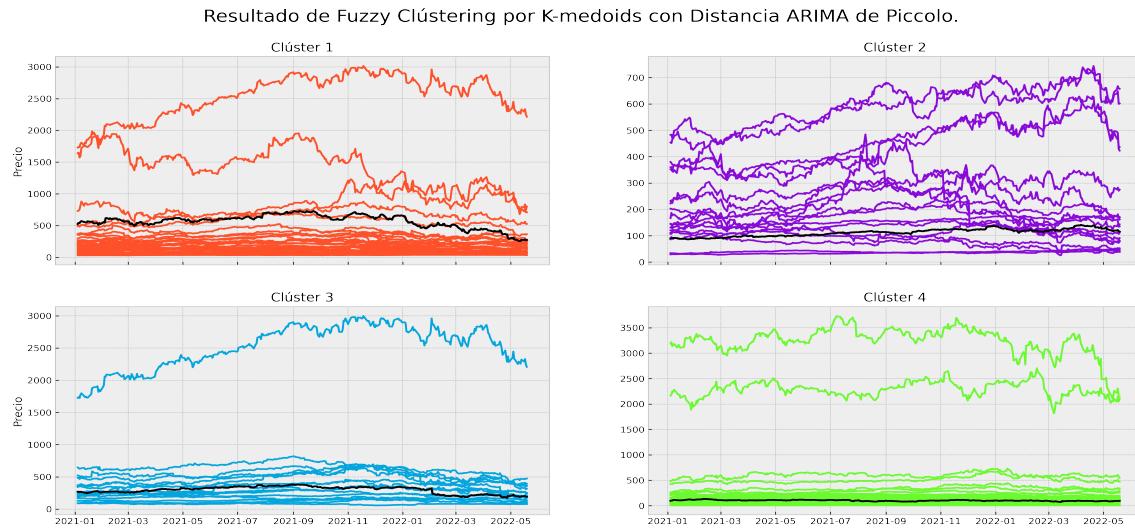


Figura 29

## 4. Conclusiones

Hemos visto una introducción al análisis de grupos para series de tiempo. Ya que el *clustering* es un método de aprendizaje no supervisado, nos permite mejorar nuestro entendimiento de una gran cantidad de fenómenos, incluyendo fenómenos con carácter temporal. Para llevar a cabo el *clustering* y que los grupos sean de calidad, es necesario utilizar extracción de características, medidas de similaridad y un método de *clustering* que tomen en cuenta las condiciones particulares del objetivo de agrupación y retos presentes en los datos.

Para el análisis de grupos en los precios al cierre del día de *trading* de los valores del NASDAQ 100, se puede concluir que la distancia euclídea es muy simple y sólo agrupa las ST por niveles de precios. El DTW simple no parece tener mucho efecto debido a que las series de tiempo tienen longitud igual y en el fondo utiliza una distancia euclídea. Dos de los enfoques más relevantes para el análisis de los valores son aquellos que toman en cuenta la estructura de autocorrelación o una estructura paramétrica, pero el método paramétrico puede ser intensivo computacionalmente debido a la estimación de muchos parámetros de manera iterada y no optimizada. Además, hemos visto que el método de fuzzy *clustering* puede ser demasiado flexible para realizar las particiones duras.

## Referencias

- Aghabozorgi, S., Seyed Shirkhorshidi, A. & Ying Wah, T. (2015). Time-series clustering – A decade review. *Information Systems*, 53, 16-38.
- Alqahtani, A., Ali, M., Xie, X. & Jones, M. W. (2021). Deep Time-Series Clustering: A Review. *Electronics*, 10(23).
- Barandas, M., Folgado, D., Fernandes, L., Santos, S., Abreu, M., Bota, P., Liu, H., Schultz, T. & Gamboa, H. (2020). TSFEL: Time Series Feature Extraction Library. *SoftwareX*, 11.
- Caiado, J. (2006). *Distance-Based Methods for Classification and Clustering of Time Series* (Tesis doctoral). H University of Lisbon.
- Everitt, B., Landauand, S., Leese, M. & Stahl, D. (2011). *Cluster Analysis* (5.<sup>a</sup> ed.). Wiley.
- Galeano, P. & Peña, D. (2000). Multivariate analysis in vector time series. *Resenhas*, 4, 1384,404.
- Giorgino, T. (2009). Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package. *Journal of Statistical Software*, 31.
- Hagiwara, M. (2021). *Real-World Natural Language Processing*. Manning.
- Hastie, T., Tibshirani, R. & Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2.<sup>a</sup> ed.).
- James, G., Witten, D., Hastie, T. & Tibshirani, R. (2014). *An Introduction to Statistical Learning: With Applications in R* (2.<sup>a</sup> ed.). Springer.
- Kalpakis, K., Gada, D. & Puttagunta, V. Distance measures for effective clustering of ARIMA time-series. En: *Proceedings 2001 IEEE International Conference on Data Mining*. 2001.
- Kaufman, L. & Rousseeuw, P. (1990). Partitioning Around Medoids (Program PAM). *Finding Groups in Data* (pp. 68-125). John Wiley Sons, Ltd.
- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*.
- Maharaj, E. A., D'Urso, P. & Caiado, J. (2019). *Time Series Clustering and Clasification*. CRC.
- Manning, C. D. & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.
- Montero, P. & Vilar, J. A. (2014). TSclust: An R Package for Time Series Clustering. *Journal of Statistical Software*, 26.
- Nguyen, M., Purushotham, S., To, H. & Shahabi, C. (2017). m-TSNE: A Framework for Visualizing High-Dimensional Multivariate Time Series. *CoRR*, abs/1708.07942.
- Paparrizos, J. & Gravano, L. (2015). k-Shape: Efficient and Accurate Clustering of Time Series. *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, 1855 -1870.
- Patel, A. A. & Arasanipalai, A. U. (2021). *Applied Natural Language Processing in the Enterprise Teaching Machines to Read, Write Understand*. O'REILLY.
- Piccolo, D. (1990). A Distance Measure For Classifying Arima Models. *Journal of Time Series Analysis*, 11(2), 153-164.
- Sardá-Espinosa, A. (2018). Comparing Time-Series Clustering Algorithms in R Using the dtwclust Package.
- van der Maaten, L. & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9, 2579-2605.
- Xu, Y. & Brereton, R. G. (2005). A comparative study of cluster validation indices applied to genotyping data. *Chemometrics and Intelligent Laboratory Systems*, 78, 30-40.