Universidad de Valladolid

FACULTAD DE CIENCIAS

TRABAJO FIN DE GRADO

Grado en Estadística

TÉCNICAS DE CLUSTERING APLICADAS A SERIES DE TIEMPO

Autora: Irene Portela Cameselle

Tutor: José Enrique Puente Domínguez

Junio 2023

Agradecimientos

A mi familia, por vuestra paciencia, sabios consejos y amor incondicional. A mi madre, fuente de inspiración y ejemplo de perseverancia, a mi padre por su optimismo contagioso, a mi hermana por ser mi eterna compañera y modelo a seguir.

A mis compañeros de carrera, por habernos apoyado y querido tanto durante estos años, a Mario, por haber sido un gran apoyo.

A los profesores, en especial a mi tutor José Enrique Puente, por transmitirme sus conocimientos.

Resumen

El presente trabajo aborda el estudio de técnicas de clustering aplicadas a series de tiempo. Se ha realizado una revisión detallada de los diferentes enfoques, algoritmos y técnicas existentes en la literatura en este campo de estudio.

Una vez revisado el estado del arte, se han aplicado los conocimientos adquiridos al análisis de los datos recopilados por las estaciones de monitoreo de óxidos de nitrógeno en Galicia. Este análisis ha permitido identificar grupos de estaciones con comportamientos anuales o semanales similares en cuanto a la concentración de óxidos de nitrógeno.

Con los resultados obtenidos del análisis clustering, se ha propuesto una zonificación de las estaciones de monitoreo de Galicia adaptada a los niveles del óxido de nitrógeno. Esta zonificación facilita la toma de decisiones en gestión y control de la calidad del aire, ya que se pueden identificar áreas con niveles o comportamientos similares de contaminación.

Abstract

This study focuses on the exploration of clustering techniques applied to time series data. A detailed review of the different approaches, algorithms, and techniques existing in the literature of this field has been conducted.

Once the state of the art has been reviewed, the acquired knowledge has been applied to the analysis of data collected from nitrogen oxide monitoring stations in Galicia. This analysis has allowed identifying groups of stations with similar annual or weekly behaviors in terms of nitrogen oxide concentration.

Based on the results obtained from the clustering analysis, a zoning of the monitoring stations in Galicia, adapted to the nitrogen oxide levels, has been proposed. This zoning facilitates decision-making in the management and control of air quality, as it enables the identification of areas with similar pollution levels or behaviors.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1. Introducción

Actualmente nos encontramos en la era de la digitalización, caracterizada por la masiva generación de datos y su respectivo almacenamiento. A medida que se acumulan los datos, cobra mayor importancia encontrar formas eficientes de analizar y extraer información.

En vista de la creciente cantidad de información disponible, se presentan nuevas oportunidades para comprender y abordar problemas complejos en múltiples áreas. En este contexto, las técnicas de análisis de datos de series temporales juegan un papel muy importante debido a la dependencia temporal inherente en los datos.

Uno de los desafíos más importantes a los que se enfrenta la sociedad actual es la contaminación del aire, un problema que afecta a nivel global. La contaminación del aire consiste en la presencia de sustancias o elementos tóxicos en la atmósfera. Estas sustancias, en determinades cantidades, pueden resultar nocivas para la salud. Se ha probado que la contaminación atmosférica está directamente vinculada con graves problemas de salud como son las enfermedades cardíacas, el cáncer de pulmón o las infecciones respiratorias. Según la Organización Mundial de la Salud (OMS), la contaminación del aire ambiental provoca cada año alrededor de 4 millones de muertes prematuras en el mundo [43]. Por otro lado, como consecuencia de la toxicidad de dichas sustancias, también se ve afectado el medio ambiente, ya que la polución produce múltiples fenómenos que amenazan tanto a los ecosistemas como a la biodiversidad.

Reducir los efectos de la contaminación del aire ambiental se ha convertido en uno de los grandes retos a los que se enfrenta la sociedad actual. Por este motivo, la regulación de la calidad del aire y la adopción de medidas preventivas para minimizar la contaminación están cada vez más presentes en las agendas políticas y sociales de todo el mundo.

En el ámbito de esta problemática, las técnicas de análisis de series de tiempo juegan un papel fundamental. El objetivo principal de esta investigación es realizar una revisión detallada de la literatura existente en el campo del clustering de series de tiempo y aplicar los conocimientos adquiridos al análisis de la calidad del aire con respecto a los óxidos de nitrógeno (NOx) en Galicia.

El análisis clustering de las estaciones de monitoreo de NOx repartidas por el territorio gallego es una valiosa herramienta para comprender y abordar la contaminación del aire de manera más efectiva. La identificación de estaciones similares en términos de magnitud o patrones de comportamiento permite establecer zonas con un comportamiento similar, lo que facilita la implementación de medidas específicas de gestión y control adaptadas a cada área.

De esta manera, se espera contribuir al control de los efectos adversos de la contaminación del NOx, promoviendo así un entorno más saludable y sostenible en Galicia, tanto para los seres humanos como para la biodiversidad.

Capítulo 2. Contexto

2.1. Aprendizaje automático

El aprendizaje automático es una rama de la Inteligencia Artificial que se centra en el desarrollo de algoritmos o sistemas que aprenden continuamente y de forma autónoma de los datos con el fin de mejorar su rendimiento en una tarea específica. Hay tres categorías de aprendizaje automático:

- **Aprendizaje supervisado**: utiliza un conjunto de datos etiquetados para entrenar un modelo que pueda predecir la salida correcta para nuevas entradas de datos no etiquetadas, es decir, de las cuales no se conoce la salida correcta. Puede resultar de gran ayuda para resolver problemas de clasificación y de regresión.
- **Aprendizaje no supervisado:** trabaja con conjuntos de datos no etiquetados. Los métodos no supervisados tienen como objetivo identificar patrones, características o estructuras existentes en los datos sin la ayuda de una salida previamente definida o esperada.
- **Aprendizaje semisupervisado:** se ubica entre el aprendizaje supervisado y no supervisado. Trabaja con un conjunto de entrenamiento que combina una pequeña cantidad de datos etiquetados con un gran número de datos no etiquetados.

El presente trabajo se enfoca en los desafíos que presenta el aprendizaje no supervisado.

2.2. Clustering

La agrupación en clústeres es un método de aprendizaje no supervisado que se encarga de revelar patrones o estructuras ocultas en un conjunto de datos. Los clústeres o conglomerados resultantes deben ser internamente lo más homogéneos posible y heterogéneos entre sí.

El proceso de agrupamiento se puede dividir en cuatro pasos [1]:

1. **Selección de las características:** cuando se trabaja con grandes conjuntos de datos, se puede dar la posibilidad de que algunas características sean ruidosas y que el proceso de clustering se vea sesgado. Es por ello, que es importante realizar una preparación previa de los datos consistente en elegir qué características son las que pueden resultarnos útiles para agrupar los datos. Esto significa tamaños de entrada más pequeños y un mejor rendimiento.

2. **Selección de algoritmos clustering**: una de las etapas cruciales es determinar cómo medir la similitud entre observaciones. Esto afecta directamente a la calidad y la interpretación de los resultados del agrupamiento. Una vez se ha elegido una medida de similitud adecuada, se debe elegir el algoritmo de clustering que se ajuste bien a los datos. La selección del número de clústeres adecuado también es una etapa crítica. Se debe elegir un número que permita una buena separación entre grupos, pero que a su vez también sea coherente con la tarea de interés y la interpretación de los resultados. Algunos métodos requieren el número de

clústeres como entrada, mientras que otros métodos permiten al usuario decidir sobre este número después de generar el clustering.

3. **Validación de resultados:** debido a que el clustering es una práctica de aprendizaje supervisado, es difícil conocer cuando el resultado de un agrupamiento es correcto. Es por ello por lo que existen diversos métodos y técnicas para evaluar los resultados obtenidos.

4. **Interpretación de resultados:** a consecuencia de ser un método de aprendizaje no supervisado, la interpretación de los datos es un proceso fundamental en el que se debe que acompañar a los resultados del agrupamiento con otras pruebas experimentales y análisis para obtener una comprensión completa del problema y tomar decisiones informadas.

2.3. Serie temporal

Definición 1. Una serie temporal X se define como $X = (x_1, x_2, \dots, x_n)$, donde x_i es el valor de la i-ésima observación de la variable aleatoria X_i, siendo n el número total de observaciones.

Una serie temporal es una secuencia de observaciones numéricas o medidas registradas en momentos sucesivos, es decir, siguiendo un orden cronológico. Por lo general, los datos se recogen en intervalos de tiempo regulares.

Para poder analizar o modelar una serie temporal univariante es importante comprender sus componentes:

- **Componente tendencial o tendencia $T(t)$:** dirección general de crecimiento o decrecimiento que sigue la serie temporal a largo plazo.
- **Componente estacional $S(t)$:** patrones repetitivos que se presentan en periodos regulares de tiempo. Variación periódica presente en toda la serie.
- **Componente cíclica $C(t)$:** oscilaciones o fluctuaciones alrededor de la tendencia de forma irregular.
- **Componente irregular o residuo $\epsilon(t)$:** error aleatorio que no puede ser explicado por las otras tres componentes, se asume que sigue una distribución normal.

La estacionaridad es una propiedad importante de las series de tiempo. Una serie se considera estacionaria si su media, varianza y autocorrelación son constantes a lo largo del tiempo. La estacionaridad permite utilizar modelos de series de tiempo para predecir patrones futuros en la serie.

Para las series de tiempo con componentes estructurales de tendencia y estacionalidad, se utilizan los siguientes modelos:

- Modelo aditivo: $X_t = T_t + S_t + \epsilon_t$
- Modelo multiplicativo: $X_t = T_t * S_t * \epsilon_t$
- Modelo mixto: $X_t = T_t * S_t + \epsilon_t$

Capítulo 3. Estado del arte

3.1. Taxonomía

Una forma de clasificar los enfoques de clustering de series de tiempo es mediante la taxonomía. En la literatura, se distinguen tres categorías principales [2]:

- **Clustering de series de tiempo completas**: consiste en agrupar un conjunto de series temporales con respecto a su similitud. Para ello, se considera toda la serie temporal un elemento a partir del cual se calculan las similitudes con las demás series temporales.

- **Clustering de subsecuencias:** consiste en agrupar subsecuencias de una sola serie temporal. Dichos fragmentos de la serie temporal se extraen a través de una ventana deslizante, "sliding window" [3].

- **Clustering de puntos de tiempo:** combina la proximidad temporal de los puntos de tiempo con la similitud entre sus valores correspondientes. Es un enfoque similar al clustering de subsecuencias con la diferencia de que no todos los puntos tienen que ser asignados a los clústeres, algunos de ellos se consideran ruido [4].

3.2. Representación

Asiduamente, a la hora de analizar series de tiempo, nos enfrentamos a un problema de alta dimensionalidad de los datos. Aplicar algoritmos sobre estas series resultaría muy costoso y poco eficaz, por lo que se han estudiado múltiples métodos de representación que ayudarán a reducir la dimensionalidad de los datos. Como consecuencia, al reducir la dimensionalidad, se consigue resaltar las características fundamentales de las series de tiempo, además de reducir su ruido. En la literatura, la representación se divide en tres tipos principales [5]:

- **Representación no adaptativa de datos:** los parámetros de la transformación se mantienen iguales para todas las series de tiempo, independientemente de su naturaleza. Las primeras representaciones no adaptativas se basan en descomposiciones espectrales, como Discrete Fourier Transform (DFT) y la Discrete Wavelet Transform (DWT). La DFT proyecta la serie de tiempo en una base de funciones seno y coseno, mientras que la DWT utiliza versiones escaladas y desplazadas de una función wavelet madre. También se han propuesto otros enfoques más específicos para las series de tiempo como Piecewise Aggregate Approximation (PAA) [5].

- **Representación adaptativa de datos:** los parámetros de la transformación se ajustan o modifican en función de los datos disponibles. Es decir, la representación se adapta a las características de los datos. Además, al agregar un proceso de selección sensible a los datos que guía el ajuste de los parámetros, se puede lograr que los métodos que originalmente no eran adaptativos a los datos

se vuelvan adaptativos. Ejemplos de este tipo de representación son Singular Value Decomposition (SVD), Adaptive Piecewise Constant Approximaton (APCA) o Symbolic Aggregate Approximation (SAX).

- **Representación basada en modelos:** este enfoque se basa en la idea de que los datos de una serie de tiempo son generados por un modelo subyacente. El objetivo principal es encontrar los parámetros de ese modelo para poder representar la serie. Algunos de estos algoritmos son los modelos Autorregresivos de Media Móvil (ARMA) o los Modelos Ocultos de Markov [6].

3.2.1. Aproximación agregada por partes (PAA)

Definición 2. Suponiendo una secuencia de tiempo $X = (x_1, x_2, \dots, x_n)$, la cual se transforma utilizando la técnica PAA. La secuencia resultante se define como $\bar{X}_i = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_k)$. Formalmente [7]:

$$\bar{x}_i = \frac{k}{n} \sum_{j=\frac{n}{k}(i-1)+1}^{\frac{n}{k}i} x_j \qquad (1)$$

PAA [8] es un método de transformación que consiste en dividir una serie temporal en k segmentos de igual longitud y calcular le media de cada segmento. Cada una de estas medias es usada como dimensión de un vector de características k-dimensional. De esta forma, se consigue una reducción de la dimensionalidad y, por tanto, una reducción de la complejidad computacional [9].

A continuación, se muestra un gráfico que ilustra el comportamiento de la técnica de aproximación agregada por partes.

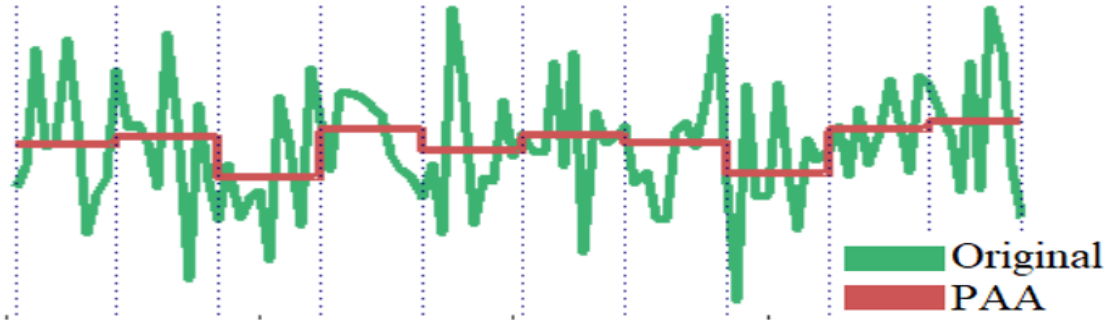

Figura 3.1. Serie de tiempo original y representada con el método PAA. Fuente: elaboración propia.

3.2.2. Aproximación agregada simbólica (SAX)

La técnica SAX [10] consiste en convertir una serie de tiempo de longitud n en k símbolos. Para ello, primero se normaliza la serie de tiempo, después se divide la serie en k segmentos de igual longitud mediante la ecuación (1) del método PAA. Por último, se calculan los "puntos de corte", que dividen el espacio de distribución en α regiones equiprobables. Estos puntos de corte son una lista ordenada de números $\beta = \beta_1, \dots, \beta_{\alpha-1}$, de manera que el área bajo la curva gaussiana $N(0,1)$ desde β_i hasta β_{i+1} sea igual a $\frac{1}{\alpha}$ [11].

Ejemplo:

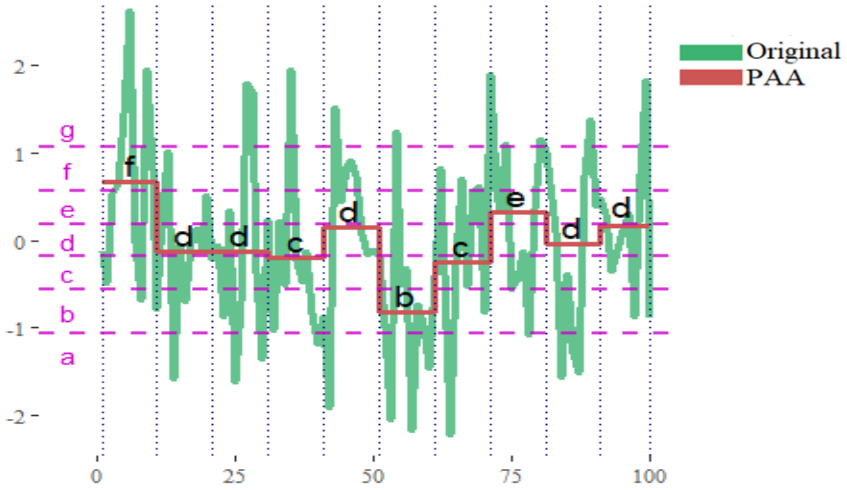

Figura 3.2. Representación SAX. Fuente: elaboración propia.

3.3. Medidas de similitud

Las medidas de distancia cuantifican la (di)similitud entre dos series de tiempo. Los autores definen la similitud como la medida que establece un valor absoluto de semejanza entre dos vectores. Esta medida juega un papel muy importante en el clustering, por lo que se han estudiado diferentes formas de medición entre series temporales. Principalmente, el tipo de similitud que miden se agrupa en [12]:

- **Similitud en el tiempo:** miden la distancia entre dos series de tiempo en el mismo instante, es decir, entre el punto i-ésimo de una serie y el punto i-ésimo de la de la otra. Una forma común de medir esta similitud es a través de la distancia euclidiana, perteneciente a las denominadas distancias L_p-norms.

- **Similitud en la forma:** se enfoca en agrupar series de tiempo que comparten patrones similares independientemente del momento en que ocurren. Para medir la similitud en la forma se utilizan medidas de distancia elásticas, que permiten comparar patrones en diferentes escalas y tiempos. Entre las formas más comunes de medir este tipo de similitud se encuentran la distorsión dinámica de tiempo (DTW) y las distancias de edición.

- **Similitud en el cambio:** busca identificar comportamientos similares en series de tiempo con una estructura de autocorrelación parecida. Para ello, se ajusta un modelo que describa el comportamiento de la serie de tiempo, como puede ser un modelo oculto de Markov o un modelo ARMA. Luego, se utiliza la similitud entre los parámetros de los modelos para medir la similitud entre series de tiempo [13].

3.3.1. Lp-norms

La distancia L_p-norm o de Minkowski es una medida de distancia que generaliza la distancia euclidiana, la distancia de Manhattan y la distancia máxima a través del parámetro p, que puede tomar cualquier valor real positivo. Este parámetro permite ajustar la importancia relativa de cada coordenada en la distancia final.

Definición 3. Sean dos series de tiempo de igual longitud $Q = (q_1, q_2, \dots, q_i, \dots, q_n)$ y $R = (r_1, r_2, \dots, r_j, \dots, r_n)$, la distancia L_p-norm se define como:

$$L_p(Q,R) = \left(\sum_{i=1}^{n} |q_i - r_i|^p\right)^{\frac{1}{p}} \quad (2)$$

- Si $p = \infty$, se obtiene la distancia máxima:

$$L_\infty(Q,R) = máx_{i=1}^{n} |q_i - r_i| \quad (3)$$

- Si $p = 1$, se obtiene la distancia de Manhattan:

$$L_1(Q,R) = \sum_{i=1}^{n} |q_i - r_i| \quad (4)$$

- Si $p = 2$, se obtiene la distancia euclidiana:

$$L_2(Q,R) = \sqrt{\sum_{i=1}^{n} (q_i - r_i)^2} \quad (5)$$

Para la medida de similitud entre series de tiempo, la distancia más usada es la euclidiana. Una de sus ventajas, además de su fácil implementación, es que es un enfoque sin parámetros, no requiere ajustes o configuraciones previas. Sin embargo, la distancia euclidiana es sensible al ruido. Asimismo, no puede lidiar con el desplazamiento local en el tiempo, lo que significa que segmentos similares que están fuera de fase pueden no ser identificados como similares por esta distancia [14].

3.3.2. Distorsión dinámica de tiempo (DTW)

La distorsión dinámica de tiempo (DTW – Dynamic Time Warping) es una técnica que calcula la ruta óptima de alineación entre dos series de tiempo. Para alinear dos secuencias se crea una matriz M de dimensión $n * m$ donde cada elemento (i, j) corresponde a la distancia local $d(q_i, r_j)$, normalmente se utiliza la distancia euclidiana para calcular este valor.

Definición 4. Sean dos series de tiempo $Q = (q_1, q_2, \dots, q_i, \dots, q_n)$ y $R = (r_1, r_2, \dots, r_j, \dots, r_m)$ de longitud n y m respectivamente y $M\,(n\,x\,m)$ la matriz de alineación entre Q y R donde cada elemento (i, j) corresponde a la distancia local $d(q_i, r_j)$. Se define como ruta de alineación $W = w_1, w_2, \dots, w_k, \dots, w_K$ (siendo el k-ésimo elemento de W definido como $w_k = (i, j)_k$ y cumpliendo $máx(n, m) \leq K \leq n + m - 1$) al conjunto de elementos contiguos de M que satisface las siguientes condiciones [15]:

- **Condición de contorno:** el camino debe comenzar en el extremo superior izquierdo de la matriz y acabar en el extremo inferior derecho de la misma.

$$w_1 = (1,1) \text{ y } w_K = (n, m) \tag{6}$$

- **Condición de continuidad:** el camino debe estar compuesto por celdas adyacentes, lo que incluye tanto celdas verticales como las horizontales y diagonales.

$$w_k = (i, j) \text{ y } w_{k-1} = (i', j') \text{ entonces } i - i' \leq 1 \text{ y } j - j' \leq 1 \tag{7}$$

- **Condición de monotonía:** los índices de la ruta deben estar ordenados en términos de tiempo.

$$w_k = (i, j) \text{ y } w_{k-1} = (i', j') \text{ entonces } i - i' \geq 0 \text{ y } j - j' \geq 0 \tag{8}$$

Hay múltiples rutas de alineación, la ruta óptima se calcula como [16]:

$$DTW(Q, R) = mín \left\{ \frac{1}{K} \sqrt{\sum_{k=1}^{K} w_k} \right. \tag{9}$$

La ruta óptima de alineación entre Q y R se puede calcular de forma eficiente usando programación dinámica para evaluar la siguiente ecuación de recurrencia [15]:

$$\gamma(i,j) = \begin{cases} d(q_i, r_j) + \text{mín} \begin{cases} \gamma(i-1,j) \\ \gamma(i,j-1) \\ \gamma(i-1,j-1) \end{cases} & , si\ i \neq 0\ y\ j \neq 0 \\ 0 & , si\ i = 0\ y\ j = 0 \\ \infty & , resto \end{cases} \quad (10$$

donde $1 \leq i \leq n$ y $1 \leq j \leq m$.

A continuación, se presentan las diferencias ente la distancia euclidiana y DTW, y, además, se presenta el camino de alineación correspondiente a la DTW de ambas series.

(a) **(b)**

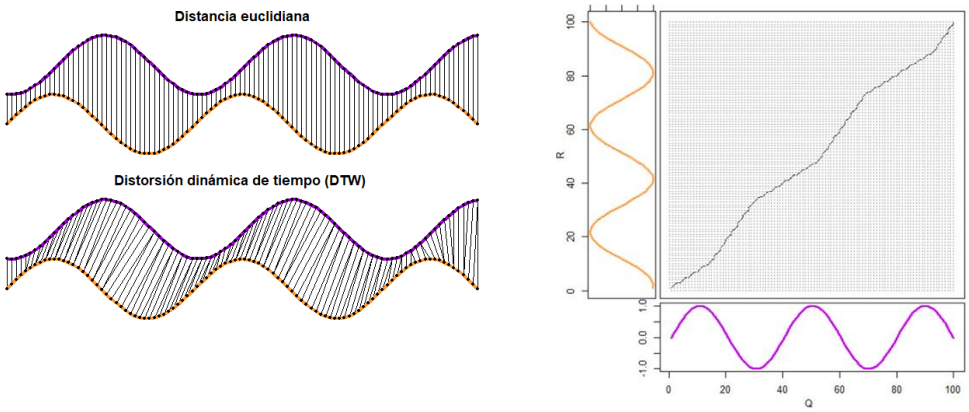

Figura 3.3. (a) Representación gráfica de las diferencias entre distancia euclidiana y DTW y sus respectivos emparejamientos. (b) Ruta óptima de alineación entre dos series Q y R. Fuente: elaboración propia.

3.3.3. Distancias de edición

Esta es una medida de similitud entre cadenas de caracteres, que consiste en cuantificar el número de operaciones elementales que son necesarias para convertir una cadena en otra. Ante la necesidad de cuantificar la similitud entre series de tiempo, surge un grupo de medidas adaptadas a series de tiempo que se basan en el concepto de la distancia de edición. Entre las más utilizadas se encuentran la subsecuencia común más larga y la distancia de edición de secuencia real.

3.3.3.1. Subsecuencia común más larga (LCSS)

Definición 5. Sean dos series de tiempo $Q = (q_1, q_2, \dots, q_i, \dots, q_n)$ y $R = (r_1, r_2, \dots, r_j, \dots, r_m)$, donde n y m no son necesariamente iguales. Se define como $LCSS\ (Q, R)$ a la secuencia más larga de elementos iguales de Q y R que se encuentran en el mismo orden relativo.

Para calcular dicha subsecuencia entre Q y R se aplica un algoritmo recursivo, cuya función de recurrencia es la siguiente [17]:

$$LCSSS(i,j) = \begin{cases} 0, & si\ i * j = 0 \\ 1 + LCSS(i-1, j-1), & si\ Q_i = R_j \\ máx \begin{cases} LCSS(i-1, j) \\ LCSS(i, j-1) \end{cases}, & resto \end{cases} \quad (11)$$

donde $1 \leq i \leq n$ y $1 \leq j \leq m$.

LCSS se deriva de la distancia de edición, originalmente diseñada para cadenas de texto. En la versión LCSS para series de tiempo, el concepto de "elementos iguales" se extiende para comparar valores numéricos en lugar de caracteres de texto. Para ello se establece un umbral de tolerancia ϵ que dicta el rango de valores que se consideran iguales, se debe cumplir:

$$q_i(1-\varepsilon) < r_j < q_i(1+\varepsilon), \varepsilon > 0 \quad (12)$$

Además, se puede establecer una ventana de tolerancia δ que controla las diferencias en la alineación temporal de los elementos de las series de tiempo. Un valor alto de δ permitirá una mayor flexibilidad en la alineación temporal.

La función de recurrencia EDR aplicada a series de tiempo resulta en:

$$LCSS(i,j) = \begin{cases} 0, & si\ i * j = 0 \\ 1 + LCSS(i-1, j-1), & si\ |Q_i - R_j| < \epsilon\ y\ |i-j| < \delta \\ máx \begin{cases} LCSS(i-1, j) \\ LCSS(i, j-1) \end{cases}, & resto \end{cases} \quad (13)$$

Finalmente, la medida de similitud LCSS entre Q y R se define como:

$$LCSS(Q,R) = \frac{n + m - 2l}{n + m} \quad (14)$$

siendo l la longitud de la subsecuencia de mayor longitud.

La mayor ventaja de esta distancia es su robustez ante el ruido, aunque esto puede afectar a su capacidad para capturar patrones sutiles en los datos. Además, su cálculo puede resultar computacionalmente muy costoso.

Ejemplo: LCSS entre dos series de tiempo $t1$ y $t2$ de longitud 12 y 14 respectivamente con $\delta = 2$ y $\varepsilon = 8$.

Se obtiene que la subsecuencia común más larga tiene longitud 9 y se calcula la medida de similitud:

$$LCSS(Q,R) = \frac{12 + 14 - 2 * 9}{12 + 14} \approx 0{,}31 \quad (15)$$

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
			-1	-2	0	2	4	6	2	0	7	5	4	2	2	0
0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	8	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0
2	10	0	0	0	1	2	2	2	2	2	2	2	2	2	2	1
3	12	0	0	0	1	2	3	3	3	3	3	3	3	3	3	2
4	14	0	0	0	1	2	3	4	4	4	4	4	4	4	4	3
5	10	0	0	0	1	2	3	4	5	5	5	5	5	5	5	4
6	8	0	0	0	1	2	3	4	5	6	6	6	6	6	6	5
7	15	0	0	0	1	2	3	4	5	6	7	7	7	7	7	7
8	13	0	0	0	1	2	3	4	5	6	7	8	8	8	8	8
9	20	0	0	0	1	2	3	4	5	6	7	8	8	8	8	8
10	18	0	0	0	1	2	3	4	5	6	7	8	8	8	8	8
11	23	0	0	0	1	2	3	4	5	6	7	8	8	8	8	8
12	8	0	0	0	1	2	3	4	5	6	7	8	9	9	9	9

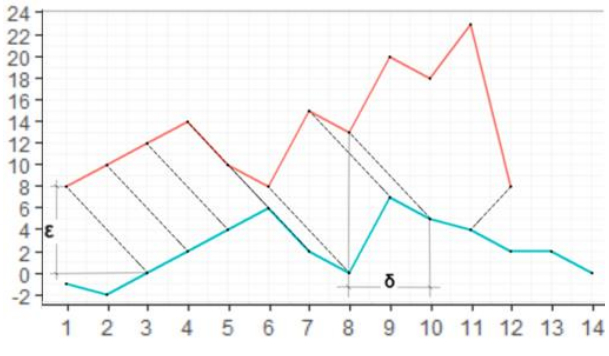

Figura 3.4. Ejemplo LCSS entre dos series de tiempo. Fuente: elaboración propia.

3.3.3.2. Distancia de edición de secuencia real (EDR)

Definición 6. Sean dos series de tiempo $Q = (q_1, q_2, \dots, q_i, \dots, q_n)$ y $R = (r_1, r_2, \dots, r_j, \dots, r_m)$, donde n y m no son necesariamente iguales. Se define como $EDR\ (Q, R)$ al número de operaciones de insertar, reemplazar o borrar que se necesitan para convertir Q en R.

Para calcular la distancia EDR entre Q y R se aplica un algoritmo recursivo, cuya función de recurrencia es la siguiente [18]:

$$EDR(i,j) = \begin{cases} i, & si\ j = 0 \\ j, & si\ i = 0 \\ mín \begin{cases} EDR\big(Resto(Q), Resto(R)\big) + subcoste \\ EDR(Resto(Q), R) + 1 \\ EDR\big(Q, Resto(R)\big) + 1 \end{cases}, & resto \end{cases} \quad (16)$$

siendo $Resto(Q)$ la subsecuencia de valores de Q que comienza desde el segundo elemento en adelante.

Para calcular el subcoste se establece la siguiente ecuación:

$$subcoste\ (q_i, r_i) \begin{cases} 0, & si\ q_i = r_i \\ 1, & resto \end{cases} \quad (17)$$

Para comprobar que $q_1 = r_1$, como en LCSS, al estar tratando con series de tiempo, se establece un umbral de tolerancia ϵ. Por tanto, $q_1 = r_1$ si $|q_1 - r_1| \leq \epsilon$, $|r_1 - q_1| \leq \epsilon$ y $\varepsilon > 0$.

La técnica EDR busca minimizar la puntuación, una puntuación cercana a 0 representa una coincidencia entre series de tiempo. EDR penaliza los huecos y los desajustes entre las series, asumiendo que el costo de reemplazar, insertar o borrar es 1.

3.4. Algoritmos

3.4.1 Partitivos

Los métodos de partición son usados para agrupar objetos en un número predefinido de grupos k. El número de grupos debe ser menor que el número de objetos y cada grupo tiene al menos un objeto asignado. Dentro de este tipo de algoritmos, los más conocidos son k-means y k-medoids, este último también se conocido como PAM (Partitioning Around Medoids). Ambos, a partir de k puntos asignados como centros, comparten el mismo objetivo, y es minimizar la distancia entre todos los objetos del grupo y el centro de dicho grupo. La diferencia entre ambos se encuentra en la forma en la que definen el centro, en k-means el centro se define como centroide y se calcula como la media de todos los puntos pertenecientes al grupo, mientras que k-medoids lo define como medoide y lo calcula como aquel punto que tiene la distancia media más cercana a los demás puntos de ese grupo.

3.4.1.1. K-means

El algoritmo k-means consiste en [19]:

Algoritmo 1: k-means

Entrada: *k* número de clústeres, *D* conjunto de datos.

Algoritmo:

(1) Seleccionar aleatoriamente *k* objetos del conjunto de datos *D* como centroides iniciales.

(2) **Repetir:**

(3) Calcular el centroide más cercano a cada objeto mediante su media y (re)asignar el objeto al clúster correspondiente.

(4) Recalcular el valor medio de cada clúster y actualizar el valor del centroide.

(5) **Hasta** satisfacer el criterio de convergencia.

En el caso de aplicar este algoritmo a series temporales, es importante definir una medida de similitud entre las series.

K-means es un algoritmo voraz, no garantiza devolver la solución óptima global, pude converger en una solución óptima local. Esta solución puede depender en gran medida de los centroides iniciales escogidos al azar. Además, k-means tiene como desventaja que es muy sensible a outliers y al ruido ya que el centro es calculado a través de la media. Por otro lado, este algoritmo tiene una ventaja, y es su bajo coste computacional.

3.4.1.2. K-medoids

En el algoritmo k-medoids (o PAM) en lugar de usar la media de los objetos en un clúster como punto de referencia, se eligen objetos específicos para representar los clústeres. A estos objetos se les denomina objetos representativos y se escoge uno por cada clúster. Los objetos no representativos se asignan al clúster cuyo objeto representativo es el más similar. El objetivo del método es minimizar la suma de las diferencias entre cada objeto y su objeto representativo correspondiente [12].

A continuación, se presentan los pasos fundamentales del algoritmo k-medoids:

Algoritmo 2: k-medoids (PAM):

Entrada: *k* número de clústeres, *D* conjunto de datos.

Algoritmo:

(1) Seleccionar aleatoriamente *k* objetos del conjunto de datos *D* como medoides iniciales.

(2) **Repetir:**

(3) Calcular el medoide más cercano a cada objeto restante de acuerdo con la distancia de medida seleccionada y (re)asignar el objeto al clúster correspondiente.

(4) Seleccionar aleatoriamente un objeto no representativo o_i.

(5) Calcular el costo S de intercambiar el objeto representativo m y o_i.

(6) Si $S < 0$, intercambiar m y o_i para formar el nuevo grupo de objetos representativos de k.

(7) **Hasta** satisfacer el criterio de convergencia.

El método k-medoids es más robusto al ruido y a los valores atípicos porque utiliza objetos reales en lugar de la media de los objetos del clúster como centro. Esto se traduce en que los valores atípicos no tienen un efecto tan fuerte en la elección del objeto representativo. Sin embargo, k-medoids es más complejo computacionalmente, para cantidades grandes de objetos y clústeres, resulta muy complejo. Por esta razón, se han estudiado en la literatura diversos métodos para mejorar la complejidad de k-medoids, entre ellos, el algoritmo CLARA.

3.4.1.3. CLARA

CLARA (de sus siglas en inglés Clustering LARge Applications) es un algoritmo de clustering que reduce la complejidad computacional mediante la creación de múltiples muestras de los objetos y aplicando el algoritmo PAM a cada muestra. Los medoides finales se obtienen a partir del mejor resultado de estos pasos [20]:

Algoritmo 3: CLARA:

Entrada: *k* número de clústeres, *D* conjunto de datos, n número de muestras, s tamaño de cada muestra.

Algoritmo:

(1) Repetir n veces:

(2) Tomar una muestra aleatoria de D de tamaño s, denominada S.

(3) Aplicar PAM a S.

(4) Clasificar D en los medoides obtenidos.

(5) Calcular la calidad del clustering en función de la distancia promedio.

(6) Devolver el mejor clustering.

El algoritmo CLARA es eficiente para trabajar con grandes conjuntos de datos. Sin embargo, el conjunto de medoides encontrados para las muestras puede no ser la mejor solución para el conjunto de datos completo. En general, el resultado final dependerá de las muestras tomadas y de la selección adecuada de los parámetros del algoritmo. En la literatura, se ha demostrado que se pueden obtener buenos resultados con CLARA utilizando un número de muestras relativamente pequeño $(n = 5)$ y un tamaño de muestra suficientemente grande $(s = 40 + 2k)$ para reducir la complejidad computacional y garantizar la representatividad de la muestra [21].

3.4.2. Jerárquicos

Los algoritmos jerárquicos son una familia de técnicas de clustering que construyen una jerarquía de clústeres anidados, donde cada nivel corresponde a un número diferente de clústeres. Dicha jerarquía se representa mediante un dendograma, en el cual, los clústeres más pequeños están contenidos dentro de los clústeres más grandes. Dependiendo de la dirección de agrupación de los objetos se distinguen dos tipos principales de algoritmos jerárquicos: aglomerativos y divisivos.

3.4.2.1. Aglomerativos

El método AHC (Agglomerative Hierarchical Clustering) inicialmente considera cada objeto como un clúster individual y, en cada paso, los dos clústeres más similares se fusionan en un nuevo clúster. Este proceso continúa iterativamente hasta que todos los objetos estén agrupados en un único clúster. El esquema de funcionamiento es el siguiente [22]:

Algoritmo 4: AHC

Entrada: *D* conjunto de datos de n objetos.

Algoritmo:

(1) **Considerar cada objeto del conjunto de datos *D* un propio clúster.**

(2) **Repetir:**

(3) Calcular la matriz de disimilaridad entre todos los pares de clústeres del conjunto de datos D.

(4) Encontrar el par de clústeres (d_i, d_j) más cercano en base a la medida de disimilaridad y fusionarlo en un único clúster (d_i).

(5) Eliminar el clúster (d_j) del conjunto de datos D.

(6) **Hasta** que el conjunto de clústeres D se reduzca a un único clúster.

La elección de la medida de distancia y la técnica de enlace para calcular la matriz de disimilaridad entre los clústeres puede afectar significativamente a los resultados del clustering.

Definición 7: Sean dos clústeres C_i y C_j con un número de instancias $|C_i|$ y $|C_j|$ respectivamente y sea $d(c_i, c_j)$ la distancia entre las instancias c_i y c_j. Los métodos de enlace más populares se definen como [23] :

Enlace simple:

$$D_{Simple}(C_i, C_j) = \min_{c_i \in C_i, c_j \in C_j} d(c_i, c_j) \tag{18}$$

Enlace completo:

$$D_{Completo}(C_i, C_j) = \max_{c_i \in C_i, c_j \in C_j} d(c_i, c_j) \tag{19}$$

Enlace promedio:

$$D_{Promedio}(C_i, C_j) = d\left(\frac{1}{|C_i|}\frac{1}{|C_j|}\right) \sum_{c_i \in C_i} \sum_{c_j \in C_j} d(c_i, c_j) \tag{20}$$

Método del centroide:

$$D_{Centroide}(C_i, C_j) = d\left(\left(\frac{1}{|C_i|}\sum_{c_i \in C_i} c_i\right), \left(\frac{1}{|C_j|}\sum_{c_j \in C_j} c_j\right)\right) = \mathrm{d}(v_i, v_j) \tag{21}$$

Método Ward:

$$D_{Ward}(C_i, C_j) = \frac{|C_i||C_j|}{|C_i| + |C_j|} d\left(\left(\frac{1}{|C_i|}\sum_{c_i \in C_i} c_i\right), \left(\frac{1}{|C_j|}\sum_{c_j \in C_j} c_j\right)\right) \tag{22}$$

El enlace simple, también conocido como vecino más cercano, toma la distancia entre dos clústeres como la menor disimilitud o distancia entre sus elementos. Este método es adecuado para formas no elípticas, además, es sensible a valores atípicos. Por su parte, el enlace completo, o vecino más lejano, toma la distancia entre dos clústeres como la mayor similitud o distancia entre sus objetos. Este enlace suele producir clústeres más compactos y es menos sensible a valores atípicos. Por otra parte, el enlace promedio, determina las distancias entre los pares de elementos de los distintos clústeres y toma el promedio de estas distancias como la distancia entre los dos clústeres. El método del centroide calcula la distancia entre los centroides de los clústeres. Este método es más robusto frente a los valores atípicos, además, tiene un mejor rendimiento que otros métodos cuando se manejan clústeres de diferentes tamaños [24].

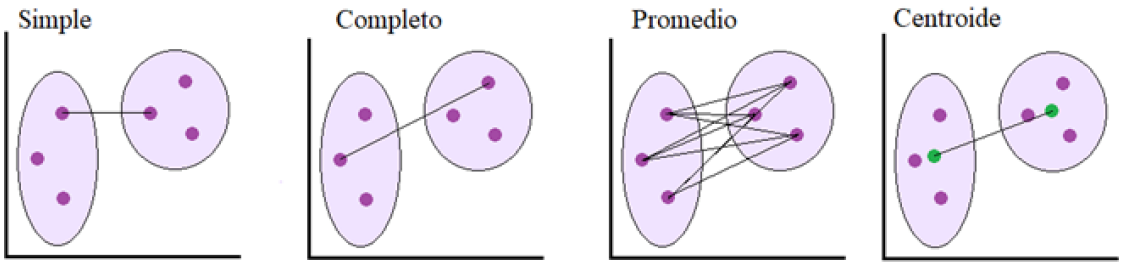

Figura 3.5. Representación de diferentes métodos de enlace para calcular la distancia entre clústeres. Fuente: elaboración propia.

El método Ward, sin embargo, es más complejo. Este método, propuesto por Ward en 1963 [25], tiene como objetivo minimizar la pérdida de información al fusionar clústeres. Para lograrlo, utiliza la noción de inercia, que se define como la suma de errores al cuadrado (ESS). Formalmente, se expresa como:

$$I(C_i) = \sum_{x \epsilon\, C_i} \left|\left|x - v_i\right|\right|^2 \tag{23}$$

donde x es un elemento del conjunto de datos, C_i es un clúster resultante de la partición y v_i su centro de gravitación.

La pérdida de información resultante de unir el clúster C_i y C_j se define como:

$$\begin{aligned} \delta(C_i, C_j) &= I(C_i \cup C_j) - I(C_i) - I(C_j) \\ &= \sum_{x \epsilon\, C_i \cup C_j} \left|\left|x - v_{i \cup j}\right|\right|^2 - \sum_{x \epsilon\, C_i} \left|\left|x - v_i\right|\right|^2 \\ &\quad - \sum_{x \epsilon\, C_j} \left|\left|x - v_j\right|\right|^2 = \frac{|\,C_i||\,C_j|}{|\,C_i| + |\,C_j|} d(v_i, v_j) \end{aligned} \tag{24}$$

Este valor corresponde a la variación de la inercia dentro del clúster.

3.4.2.1. Divisivos

Los algoritmos de clustering divisivos parten de la idea de que todos los datos pertenecen a un único clúster y luego se van dividiendo en clústeres más pequeños. Este algoritmo se realiza de manera recursiva hasta que cada clúster solo tiene un objeto. En cada iteración, los clústeres se separan de manera que los objetos dentro de cada grupo sean más parecidos entre sí que con los otros objetos de los otros clústeres [26].

En la práctica, se usan más los algoritmos aglomerativos que los divisivos. Y es que, en los algoritmos divisivos, cuando el conjunto de objetos n es muy grande, al existir $2^{n-1} - 1$ posibles formas de dividir un conjunto de tamaño n en dos subconjuntos, resulta prohibitivo computacionalmente examinar todas las posibilidades. Por tanto, se usan heurísticas para la partición, lo que puede conllevar resultados imprecisos [27].

3.4.3. Fuzzy

La base fundamental de los algoritmos *fuzzy* es la lógica difusa, que se basa en la idea de que los límites entre categorías no son absolutos, sino borrosos o difusos. Estos algoritmos, en lugar de asignar cada objeto a un determinado clúster, asignan grados de pertenencia de cada objeto a los diferentes clústeres de la partición.

Estos algoritmos, para un conjunto de datos $\{x_k \,|k = 1, \dots, n\}$, crean una matriz de partición $\mathrm{U} = [u_{ij}]_{cxn}$ donde $u_{ij} \in [0,1]$ representa el grado de pertenencia elemento j al clúster i. Este coeficiente debe cumplir las siguientes restricciones [28]:

$$0 < \sum_{j=1}^{n} u_{ij} < n \; \forall i \tag{25}$$

$$\sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij} = n \tag{26}$$

$$\sum_{i=1}^{c} u_{ij} = 1 \; \forall j \tag{27}$$

Uno de los algoritmos difusos más conocidos es fuzzy c-means (FCM)

3.4.3.1 Fuzzy c-means

Dunn en 1973 [29] realizó una extensión del algoritmo c-means para permitir una partición difusa, para ello, define la siguiente función objetivo:

$$J_D(U,V) = \sum_{i=1}^{c} \sum_{j=1}^{n} {u_{ij}}^2 \left|\left|x_j - v_i\right|\right|^2 \tag{28}$$

donde, $\{x_k \,|j = 1, \dots, n\}$ es el conjunto de datos, $\{v_i \,|i = 1, \dots, c\}$ el conjunto de centroides y u_{ij} representa el grado de pertenencia del elemento x_j al clúster i.

Después, en 1981, Bezdek [30] generalizó $J_D(U,V)$ a $J_{FCM}(U,V)$ con la siguiente función objetivo:

$$J_{FCM}(U,V) = \sum_{i=1}^{c}\sum_{j=1}^{n} {u_{ij}}^m \left|\left|x_j - v_i\right|\right|^2 \tag{29}$$

donde m representa el grado de difusión $(1 < m < \infty)$ [31].

Las funciones de los centroides y sus respectivos grados de pertenencia necesarias para minimizar la función objetivo son [28]:

$$v_i = \frac{\sum_{j=1}^{n}\left(u_{ij}\right)^m x_j}{\sum_{k=1}^{n}\left(u_{ij}\right)^m}\ , i = 1, \dots, c \tag{30}$$

$$u_{ij} = \left(\frac{\left|\left|x_j - v_i\right|\right|^2}{\sum_{k=1}^{c}\left|\left|x_j - v_k\right|\right|^2}^{\frac{1}{m-1}}\right)^{-1}\ , i = 1, \dots, c\ ; j = 1, \dots, n \tag{31}$$

El algoritmo sigue los siguientes pasos [32]:

Algoritmo 4: Fuzzy c-means

Entrada: *D* conjunto de datos $\{x_k\ | k = 1, \dots, n\}$

m grado de difusión $(1 < m < \infty)$

c número de clústeres $(2 \le c \le n)$

ε tolerancia para parar el proceso iterativo

Algoritmo:

(1) Inicializar el contador $l = 0$

(2) Inicializar la matriz $U^{(l)}$ aleatoriamente

(3) **Repetir**

(4) Calcular el centroide ${v_i}^{(l)}$ con la ecuación (30)

(5) Calcular la matriz de pertenencia $U^{(l+1)}$

Si $x_j \neq v_i$ entonces u_{ij} se calcula con la ecuación (31)

Sino si $i = j$ entonces $u_{ij} = 1$

Sino $u_{ij} = 0$

(6) Calcular $\Delta = \left\|U^{(l+1)} - U^{(l)}\right\|$ e incrementar el contador l

(7) **Hasta** $\Delta \le \varepsilon$

3.5. Validación

Los índices de validación de clústeres (CVI) se utilizan para evaluar la calidad de los resultados del agrupamiento. En la literatura, se distinguen dos tipos de CVI, externos e internos.

Los CVI externos comparan los resultados del clustering con una partición de referencia que contiene la información correcta sobre la agrupación de los datos. En la práctica, generalmente, la partición de referencia no está disponible, lo que limita el uso de CVI externos [33].

Por otro lado, los CVI internos no precisan de información externa. Se basan en la estructura interna de los clústeres y la distribución de los datos en ellos. Estos índices evalúan dos características fundamentales [34]:

- **Compactación:** evalúa la distancia o similitud intra-clúster, es decir, la cohesión entre los elementos de un mismo clúster.

- **Separación:** evalúa la distancia o similitud inter-clúster, es decir, cuantifica la distancia o similitud entre los clústeres.

A continuación, se definen algunos de los CVI internos más populares.

3.5.1 Silhouette

El índice de Silhouette, propuesto por P.J. Rousseeuw [35], calcula el índice de Silhouette para cada uno de los elementos del conjunto de datos. Para ello, calcula cuán cercano es cada elemento dentro de su propio clúster y, simultáneamente, la distancia más próxima entre dicho elemento a otro clúster.

Sea i un elemento perteneciente a un clúster A, $a(i)$ la distancia media entre i y los demás elementos de A y $d(i, C)$ la distancia media entre i y los demás elementos del clúster C y sea $b(i) = \min\{d(i, C)\}$. El índice de Silhouette $S(i)$ del elemento i se define como:

$$S(\mathrm{i}) = \frac{b(i) - a(i)}{máx\{a(i), b(i)\}} \tag{32}$$

Sean N elementos en el conjunto de datos, el valor global de Silhouette se define como:

$$S = \frac{1}{N}\sum_{i=1}^{N} S(i) \tag{33}$$

El rango de valores que puede tomar el índice se encuentra entre -1 y 1. Los valores cercanos a 1 indican un buen agrupamiento, donde los grupos están bien separados y los puntos están bien clasificados dentro de sus respectivos grupos. Por el contrario, un valor cercano a 0 corresponde a una clasificación dudosa, y un valor cercano a -1 se considera una mala agrupación.

3.5.2 Dunn

Este índice, propuesto por J.C. Dunn [36], se calcula como la razón de la mínima distancia intra-clúster y la máxima inter-clúster [37]:

Dado un agrupamiento en K grupos, donde C_i y C_j son respectivamente el i-ésimo y j-ésimo clúster de la partición.

La separación entre el clúster C_i y C_j se calcula como la mínima distancia entre sus respectivos centroides v_i y v_j:

$$\delta_{i,j} = \mathrm{d}(v_i, v_j) \tag{34}$$

El grado de compactación de un clúster C_i se calcula como la máxima distancia entre los elementos del clúster.

$$\Delta_i = \max_{x_i, x_j \in C_i} \mathrm{d}(x_i, x_j) \tag{35}$$

Finalmente, el índice de Dunn se obtiene de dividir la mínima separación entre los pares de clústeres por la máxima compactación de todos los clústeres:

$$DI = \frac{\min\limits_{1 \le i \ne k \le K} \delta_{i,k}}{\max\limits_{i \le k \le K} \Delta_k} \tag{36}$$

El índice de Dunn puede tener valores entre $[0, \infty)$. A medida que el valor del índice aumenta, mejor es la agrupación.

3.5.3 COP

El índice de COP, propuesto en [38], es un índice de tipo ratio que calcula la cohesión de los elementos como la distancia promedio entre los elementos de un clúster y su centroide y la separación como la distancia más lejana entre dos pares de elementos de distintos clústeres [39].

Siendo N el número total de elementos en el conjunto de datos, C los clústeres resultantes de la partición y C_k un clúster perteneciente a C. El índice COP se calcula como:

$$COP = \frac{1}{N} \sum_{C_k \in C} n_k \frac{intra_{COP}(C_k)}{inter_{COP}(C_k)} \tag{37}$$

donde

$$intra_{COP}(C_k) = \frac{1}{n_k} \sum_{x_i \in C_k} d(x_i, v_k) \tag{38}$$

$$inter_{COP}(C_k) = \min_{x_{i \notin} C_k} \max_{x_j \in C_k} \mathrm{d}(x_i, x_j) \tag{39}$$

siendo x_i y x_j elementos del conjunto de datos, n_k el número de elementos pertenecientes al clúster C_k y v_k el centroide del clúster C_k.

Un valor más bajo de este índice indica una mayor cohesión interna y una mejor separación externa, lo cual significa una mejor solución del agrupamiento.

3.5.4 Davies-Bouldin

El índice DB, propuesto por D.L. Davies y D.W. Bouldin en [40], este índice cuantifica la compactación como la distancia promedio de los objetos a sus respectivos centroides, y la separación como la distancia entre centroides [39].

El grado de compactación, calculado como la distancia promedio de todas las muestras en el clúster Ci a su centroide v_i, se expresa formalmente como:

$$s_i = \frac{1}{n_i} \sum_{x_i \in C_i} \mathrm{d}(x_i, v_i) \tag{40}$$

La separación entre el clúster C_i y C_j se calcula como la mínima distancia entre sus respectivos centroides v_i y v_j:

$$d_{i,j} = \mathrm{d}(v_i, v_j) \tag{41}$$

Finalmente, siendo K el número de clústeres, el índice DB se calcula como:

$$DB = \frac{1}{K} \sum_{k=1}^{K} \max_{k \neq i} \left\{ \frac{s_i + s_k}{d_{i,k}} \right\} \tag{42}$$

Los valores que dicho índice puede tomar se encuentran en el rango $[0, \infty)$ y, cuanto menor sea el índice DB, mejor será la agrupación.

3.5.6 Calinski-Harabasz

El índice CD, propuesto por T. Calinski y J.Harabasz [41], se basa en la relación entre la dispersión intra-clúster y la dispersión inter-clúster. El índice se calcula como:

$$CH = \frac{\frac{BGSS}{K-1}}{\frac{WGSS}{N-K}} \tag{43}$$

donde:

- K es el número de clústeres de la partición
- N es el número total de elementos en el conjunto de datos
- BGSS *(between-group sum of squares)* es la dispersión entre los clústeres. Basada en la distancia entre los centroides de cada clúster al centroide global [42].
- WGSS *(within-group sum of square distance)* es la dispersión de los elementos dentro de un mismo clúster. Basada en la distancia entre los puntos de un clúster y su centroide [42].

Siendo un clúster C_k perteneciente al conjunto de clústeres C, n_k el número de elementos pertenecientes al clúster C_k, v_k el centroide de C_k y v el centroide de todo el conjunto de datos y siendo x un elemento del conjunto de datos, BGGS y WGSS se calculan respectivamente como:

$$BGSS = \sum_{C_k \in C} n_k \, d^2(v_{k,} v) \tag{44}$$

$$WGSS = \sum_{C_k \in C} \sum_{x_i \in c_k} d^2(x_i, v_k) \tag{45}$$

El índice CH puede tomar valores entre $[0, \infty)$, cuanto mayor sea el índice mejor se considera el agrupamiento. Esto se traduce en valores altos de BGSS y bajos de WGSS.

Capítulo 4. Datos

Las emisiones y los efectos de los distintos agentes contaminantes varían en función de su tipo y concentración en el aire ambiental. En este sentido, la legislación de cada país establece límites legales, basándose en las recomendaciones y estándares internacionales establecidos por organizaciones como la OMS.

En España, el Ministerio para la Transición Ecológica y el Reto Demográfico, es el encargado de establecer la normativa y los estándares para la calidad del aire, así como de supervisar y coordinar la medición de la calidad del aire de la red de estaciones de medición de cada una de las Comunidades Autónomas. Estas últimas junto con el Ministerio, deben trabajar conjuntamente para salvaguardar la calidad del aire en todo el territorio español.

El Real Decreto 102/2011, de 28 de enero, de acuerdo con el anexo III de la Ley 34/2007 sobre la calidad del aire y la protección de la atmósfera, tiene como objeto definir y establecer objetivos de la calidad del aire con respecto a los siguientes agentes contaminantes: dióxido de azufre (SO2), dióxido de nitrógeno (NO2) y óxidos de nitrógeno (NOx), material particulado (PM10 Y PM2,5), monóxido de carbono (CO), plomo (Pb), benceno (C6H6), ozono, arsénico, cadmio, níquel y benzo(a)pireno (BaP) [44].

A la hora de evaluar la calidad del aire para la protección de la salud humana, se establecen unos valores límite en los niveles de SO2, NO2, Pb, C6H6, CO y partículas PM10 y PM2,5. Por otra parte, en cuanto a la evaluación de la calidad para la protección de la vegetación, se fijan niveles críticos de SO2 y NOx.

Este trabajo se enfoca en el estudio de los niveles de NOx. Este término hace referencia a un grupo de gases formado por NO y NO2. Tanto el NO como el NO2 son liberados directamente a la atmósfera tanto por fuentes naturales como antropogénicas, como son las emisiones de los vehículos automóviles, fábricas o centrales eléctricas.

A continuación, se presenta una ficha informativa con las principales características de los óxidos de nitrógeno. En ella se incluye el nivel crítico medio anual para la protección de la vegetación establecido en el Anexo VI de Real Decreto 102/2011 [44], sus principales fuentes de emisión y los riesgos que puede suponer tanto para la salud como para el medio ambiente.

Óxidos de nitrógeno (NOx)	
Nivel crítico	30 $\mu g/m^3$
Periodo	Año civil
Fuentes de emisión	- Transporte - Centrales eléctricas - Quema de biomasa
Efectos o riesgos	- Enfermedades cardiovasculares - Problemas respiratorios - Lluvia ácida - Cambio climático - Pérdida de biodiversidad

Tabla 4.1. Características básicas NOx. Fuente: elaboración propia.

Cada Comunidad Autónoma establece zonificaciones de calidad del aire adaptadas a aquellos contaminantes utilizados para evaluar la protección de la salud. La zonificación establece áreas con comportamientos similares en términos de dispersión de contaminantes atmosféricos. Esto permite definir zonas de calidad del aire equivalentes, lo cual es fundamental para poder implementar planes de actuación específicos en cada una de las zonas [45].

La Red de Calidad del Aire de Galicia establece que las áreas con una concentración de población de más de 50.000 habitantes o una densidad de habitantes por kilómetro cuadrado que justifique el control de la calidad del aire, son consideradas como aglomeraciones. En Galicia, las ciudades de Pontevedra, Vigo, A Coruña, Ferrol, Santiago, Lugo y Ourense son catalogadas como aglomeraciones. Por esto último, dichas urbes se consideran como zonas de calidad del aire independientes, en las que es obligatorio instalar estaciones de medición de los contaminantes previamente mencionados. Además, se han instalado otras estaciones que se encuentran repartidas a lo largo del territorio gallego [45].

Según el último informe publicado por la Consellería de Medio Ambiente, Territorio e Vivenda de 2021, se establecen 6 adaptadas a las características de diferentes contaminantes en el territorio gallego. La primera de ellas corresponde al SO2, la segunda al NO2 y CO, la tercera a las partículas en suspensión PM10 y PM2,5, la cuarta establece zonas para el O3, la quinta para el BaP y la sexta para el C6H6 y metales pesados. Sin embargo, el contaminante NOx no cuenta con una zonificación específica adecuada a sus niveles.

Aunque actualmente la legislación no establece un valor límite en los niveles de NOx para evaluar la calidad del aire en términos de protección de salud humana, sería apropiado tener una zonificación específica para este contaminante. Esto permitiría un control más efectivo y una mayor vigilancia de las zonas con niveles altos de contaminación, además de facilitar la toma de decisiones informadas en materia de protección de la salud pública.

Este análisis tiene como objetivo proponer una zonificación específica para el contaminante NOx. Para lograrlo, se aplica un análisis de técnicas clustering que agrupa las estaciones encargadas de medir los niveles de NOx en base a sus distintos comportamientos, patrones y magnitudes de sus niveles. Los resultados obtenidos permiten comprender la distribución y el impacto de las diferentes fuentes de emisión en los niveles del contaminante. Esto posibilita establecer zonas apropiadas para el NOx.

Se parte de una zonificación que distingue las 7 zonas correspondientes a cada ciudad con su respectiva área metropolitana, así como una octava zona, denominada Galicia Rural, donde se ubican el resto de las estaciones. Esta zonificación es la misma que se utiliza en la normativa vigente en 2021 para evaluar los contaminantes NO2 y CO en Galicia.

A partir de esta zonificación, se lleva a cabo un estudio con el fin de determinar la inclusión de alguna zona adicional o la redefinición de las ya existentes. El propósito es plantear una zonificación lo más precisa y representativa posible de los niveles de NOx en la Comunidad Autónoma.

Los datos utilizados han sido publicados por la Xunta de Galicia a través de la plataforma digital de la Unidad de Observación y Predicción Meteorológica de Galicia (MeteoGalicia). Dichos datos han sido previamente sometidos a rigurosos procesos de verificación y validación.

En 2022 la Red de Calidad del Aire de Galicia cuenta con un total de 39 estaciones que se encargan de medir los niveles de NOx. Del total de las estaciones, 13 son gestionadas por la Xunta de Galicia, 24 son gestionadas por la Red Industrial de Galicia y las 2 estaciones restantes, forman parte de una red de vigilancia de carácter europeo. Los registros de estas dos últimas estaciones no figuran publicados en la plataforma digital. Por lo tanto, solo se tienen en cuenta 37 estaciones de medición.

El NOx, como se ha comentado anteriormente, es un contaminante utilizado para evaluar la apropiada calidad del aire para la vegetación. Sin embargo, es importante señalar que tan solo 4 de las 39 estaciones de medición se consideran aptas para su evaluación. Estas estaciones se ubican y denominan Fraga Redonda, Laza, Noia y O Saviñao. De las mediciones de estas dos últimas estaciones, como pertenecen a la mencionada red de vigilancia de carácter europeo, no se tienen datos.

Las estaciones de medición de NOx se encuentran divididas en diferentes tipos de áreas en función de su ubicación geográfica en la comunidad gallega. Además, cada una de ellas está diseñada para medir las concentraciones en un entorno específico, que puede ser rural, urbano o suburbano, y se clasifican en tres tipos principales: estaciones de fondo, de tráfico e industriales.

Las estaciones de fondo miden la concentración de contaminantes en el aire en áreas que no están influenciadas por fuentes contaminantes locales, como el tráfico o las emisiones de industria. Por su parte, las estaciones de tráfico miden los niveles del contaminante en áreas con gran circulación vehicular. Por último, las estaciones industriales realizan mediciones en áreas con actividad industrial, como pueden ser fábricas, centrales térmicas u otras instalaciones que pueden ser grandes fuentes de emisión.

Los datos analizados corresponden a las mediciones de NOx recogidas durante todo el año calendario de 2022. Cada una de las estaciones de medición realiza una recogida de datos cada hora, lo que implica que se han recogido mediciones durante todos los días del año y en todas las horas del día. Hay un total de 8760 registros por estación, los cuales comienzan el 1 de enero de 2022 a la 00:00 y acaban el 31 de diciembre de 2022 a las 23:00, abarcando así todo el año. Es importante destacar que estas mediciones son recogidas con un intervalo de tiempo equidistante de una hora, lo que garantiza una consistencia y comparabilidad en la recopilación de datos en todas las estaciones de medición.

Capítulo 5. Herramientas

Se ha utilizado el entorno de desarrollo integrado *RStudio*. Específicamente, para realizar el análisis clúster, se hizo uso de dos paquetes, *factoextra* y *dtwclust*.

La función principal que ofrece el paquete *dtwclust* es *tsclust()*. Esta función se utiliza para realizar clustering de series de tiempo y tiene varios argumentos, entre ellos [46]:

-*Series:* series de tiempo que se utilizarán para el clustering.

-*Type*: permite especificar el tipo de clustering que se desea realizar. Puede ser partitivo, jerárquico, jerárquico con poda dinámica o difuso.

-*k*: número deseado de clústeres.

-*Prepoc:* función que permite preprocesar los datos de las series de tiempo antes de realizar el clustering.

-*Distance:* permite especificar la medida de distancia para el cálculo de similitud entre las series de tiempo.

El paquete *factoextra* ofrece la función *fviz_dend(),* que se utiliza para visualizar dendogramas resultantes del clustering jerárquico.

Además, el paquete *factoextra* cuenta con la función *fviz_nclust(),* que se utiliza, entre otras cosas, para determinar el número óptimo de clústeres en los algoritmos partitivos.

Capítulo 6. Análisis

Para llevar a cabo el análisis clustering de series temporales se sigue la siguiente metodología:

1. Identificación de la distribución espacial de las estaciones de monitoreo.
2. Tipificación de las estaciones de medición.
3. Análisis descriptivo de las series temporales.
4. Análisis clúster y propuesta de zonificación.
5. Conclusión.

Para comenzar el análisis, con el fin de comprender la distribución geográfica de las estaciones de medición, se presenta un mapa que permite visualizar de forma gráfica su ubicación en la región.

Figura 6.1. Localización estaciones de medición NOx en Galicia. Fuente: elaboración propia a partir del programa Google Earth Pro.

Además de localizar las estaciones en el mapa, para lograr una comprensión completa de la distribución de estos puntos de monitoreo, también es importante conocer sus características. Para ello, para cada provincia, se presenta una tabla en la que se especifica la zona, el tipo de estación y el tipo de área.

Zona	Estación	Tipo de estación	Tipo de área
A Coruña + Área Metropolitana	A Grela	Industrial	Urbana
	San Pedro	Industrial	Suburbana
	A Coruña - Torre de Hércules	Fondo	Suburbana
Santiago +Área Metropolitana	Santiago - San Caetano	Tráfico	Urbana
	Santiago - Campus	Fondo	Suburbana
Ferrol + Área Metropolitana	Ferrol – Reina Sofía	Tráfico	Urbana
	A Cabana	Industrial	Suburbana
Galicia rural	Cee	Industrial	Urbana
	Teixeiro	Industrial	Suburbana
	Centro Cívico	Industrial	Suburbana
	Parque da Cega	Industrial	Suburbana
	Magdalena	Industrial	Suburbana
	Louseiras	Industrial	Rural
	Marraxón	Industrial	Rural
	Maciñeira	Industrial	Rural
	Fraga Redonda	Industrial	Rural
	Buscás	Industrial	Rural
	Pastoriza	Industrial	Rural
	Rodís	Industrial	Rural
	Dumbría	Industrial	Rural
	Paiosaco	Industrial	Rural

Tabla 6.1. Estaciones ubicadas en la provincia de A Coruña.

Zona	Estación	Tipo de estación	Tipo de área
Pontevedra	Pontevedra - Campolongo	Tráfico	Urbana
Vigo + Área Metropolitana	Vigo - Coia	Tráfico	Urbana
	Vigo - Lope de Vega	Tráfico	Urbana
	Estación 1 – Este	Industrial	Urbana
	Estación 2 – Oeste	Industrial	Urbana
Galicia Rural	Campelo	Industrial	Rural
	Lalín	Fondo	Suburbana
	Ponteareas	Fondo	Suburbana

Tabla 6.2. Estaciones ubicadas en la provincia de Pontevedra.

Zona	Estación	Tipo de estación	Tipo de área
Lugo	Lugo - Fingoi	Tráfico	Urbana
Galicia rural	Xove	Industrial	Suburbana
	Est-Ou	Industrial	Suburbana
	Sur	Industrial	Rural
	Mourence	Industrial	Rural

Tabla 6.3. Estaciones ubicadas en la provincia de Lugo.

Zona	Estación	Tipo de estación	Tipo de área
Ourense	Ourense – Gómez Franqueira	Tráfico	Urbana
Galicia rural	Xinzo de Limia Laza	Tráfico Fondo	Suburbana Rural

Tabla 6.4. Estaciones ubicadas en la provincia de Ourense.

Se puede observar que la distribución de las estaciones no es uniforme por todo el territorio gallego. La provincia que cuenta con un mayor número de estaciones es A Coruña, con un total de 21, seguida de Pontevedra que cuenta con 8, Lugo con 5 y finalmente Ourense con 3 estaciones.

Esta distribución desigual se relaciona con la ubicación de los núcleos urbanos y zonas industriales. La provincia de A Coruña, que es la que presenta una mayor cantidad de estaciones, es la más poblada de Galicia, en ella se ubican importantes zonas industriales en las ciudades de A Coruña y Ferrol. Por su parte, Pontevedra, que es la segunda provincia con más estaciones de monitoreo de NOx, cuenta también con una gran densidad de población además de importantes núcleos industriales como Vigo y su área metropolitana, así como un puerto comercial importante. Por otro lado, Lugo y Ourense son provincias menos pobladas y con menor actividad industrial, lo que se refleja en una menor cantidad de estaciones de monitoreo en estas zonas. En definitiva, la presencia de zonas industriales y núcleos urbanos en las provincias de A Coruña y Pontevedra explica el porqué de un mayor número de estaciones de monitoreo.

No obstante, se pueden apreciar ciertas peculiaridades en la forma en la que se distribuyen los distintos tipos de estaciones en estas provincias. Por ejemplo, en la provincia de A Coruña, se encuentra una gran cantidad de estaciones industriales. En concreto, de las 21 estaciones existentes, 17 pertenecen a dicho tipo. Si bien es cierto que A Coruña es una provincia con gran actividad industrial, también se caracteriza por tener una gran densidad de población y tráfico rodado, por lo que la ausencia de una estación de tráfico en la ciudad es llamativa.

Además, es importante poner de manifiesto que muchas de ellas están próximas entre sí, como las estaciones de Marraxón, Maciñeira, Fraga Redonda y Magdalena o el caso de San Pedro, A Grela, Parque da Cega, Pastoriza y Centro Cívico. En Lugo también se puede observar que en la zona rural hay cuatro estaciones industriales, sin embargo, en la ciudad hay sólo una estación de tráfico y no hay ninguna de fondo.

Por otra parte, en Pontevedra, se puede apreciar un reparto más igualado del tipo de estaciones. De las 8 estaciones existentes, un tercio es industrial, otro tercio de tráfico y el resto de fondo.

A priori, esta situación sugiere la posibilidad de que existan ciertas carencias o redundancias en el reparto de las estaciones que miden los niveles de NOx en la Red de Calidad del Aire de Galicia.

Para comenzar, se hace un análisis descriptivo de los datos recogidos por cada uno de los puntos de monitoreo. Con ello se puede adquirir una noción global de las características fundamentales de cada estación. Esto permite identificar problemas que podrían estar presentes en los datos, como los valores ausentes, datos inconsistentes o duplicados.

En este caso, se realiza una verificación inicial de los datos y se encuentra que hay valores ausentes, pero no hay datos inconsistentes ni duplicados. Es importante abordar los valores ausentes antes de continuar con cualquier análisis, ya que pueden afectar significativamente a los resultados. Se excluyen del análisis aquellas estaciones con un porcentaje mayor del 5% de valores ausentes, estas estaciones son Est – Ou y Xinzo de Limia, ubicadas en Lugo y Ourense respectivamente. En el resto de las estaciones, se imputan los valores ausentes, para ello se usa el método de Kalman. Dicho método puede proporcionar una estimación precisa de los valores ausentes ya que es adecuado para datos con una estructura de correlación y dependencia como son las series temporales.

Originalmente, los datos registran una medición horaria de cada día del año 2022. Al estudiar los niveles de cada una de las estaciones, se tienen 35 series de tiempo de longitud 8760. Con el fin de facilitar la visualización, reducir el ruido de las series y reducir también el costo computacional que implicaría aplicar algoritmos de clustering en series de tal longitud, se consideraron dos comportamientos para el análisis, uno, el comportamiento semanal y otro, el comportamiento anual.

A continuación, se representan ambos comportamientos. Además, se realiza un análisis clúster para comprender cómo se agrupan las series en base a estos comportamientos y explorar los posibles factores que puedan influir en dichas agrupaciones.

En primer lugar, el comportamiento semanal consiste en el registro horario de los valores de NOx durante los siete días de la semana. Para su cálculo, se agrupan los datos por día de la semana y hora y se calcula la media de los registros correspondientes, obteniendo como resultado 35 series de tiempo de longitud 128. Este enfoque, permite identificar y comprender los patrones que se producen a lo largo de los días de la semana. A continuación, se muestra su representación:

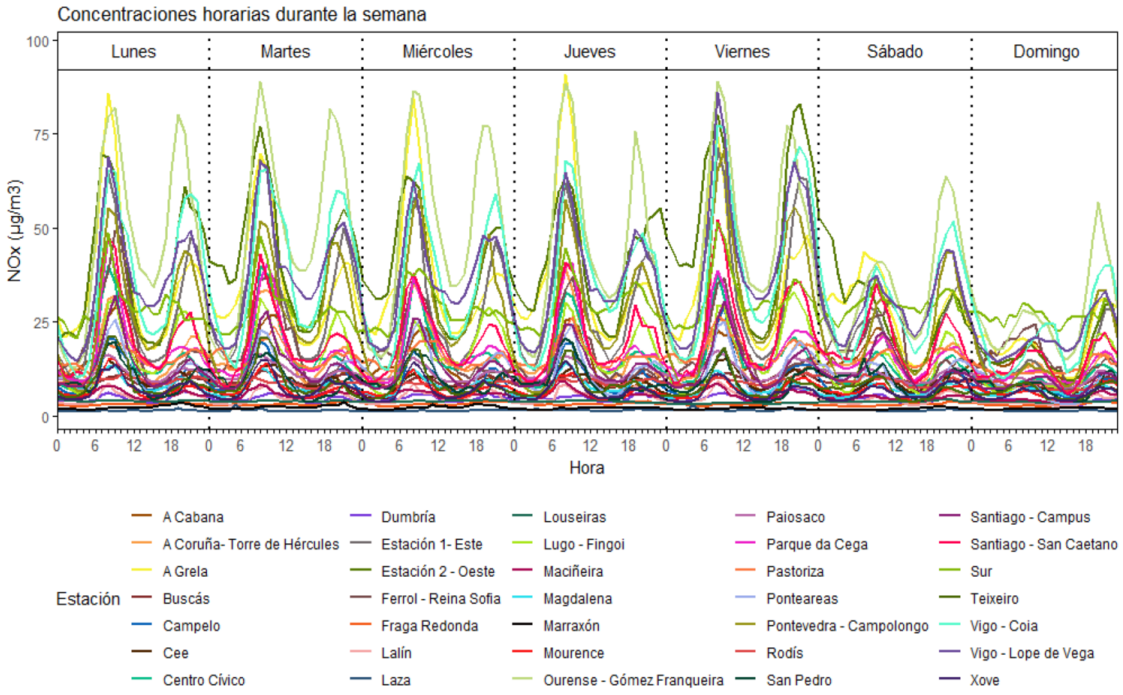

Figura 6.2. Representación gráfica comportamiento semanal en las estaciones.

Se observa que, por lo general, los días laborales presentan un patrón de concentración de NOx caracterizado por la presencia de dos picos. Estos máximos suelen ocurrir durante las horas de la mañana, entre las 7:00h y las 11:00h, y durante las horas de la tarde, entre las 18:00h y 22:00h aproximadamente, siendo el pico de concentración de la mañana ligeramente más alto. Por otro lado, durante los fines de semana, se observa una tendencia opuesta, en estos días se tiende a registrar niveles más altos de NOx durante las horas de la tarde.

Por otra parte, no hay ningún registro que llame la atención por ser anómalo, las mediciones de todas las estaciones se encuentran en los rangos esperables. Cabe destacar también que el comportamiento y magnitud de las mediciones varía según la estación.

Para realizar un estudio más detallado de los patrones existentes y para agrupar las estaciones según sus similitudes en el comportamiento de las concentraciones de NOx durante la semana, se realiza un análisis clúster.

En primer lugar, para agrupar las estaciones de manera efectiva, se normalizan los datos, eliminando cualquier sesgo causado por diferencias de escala entre las estaciones. Posteriormente, se utilizan técnicas de clustering jerárquico aglomerativo con el método Ward para agrupar las estaciones. Además, para evaluar la similitud entre las series de tiempo, se utiliza la distancia DTW, considerando así los desplazamientos y deformaciones de las series en el tiempo.

A continuación, se presentan los resultados:

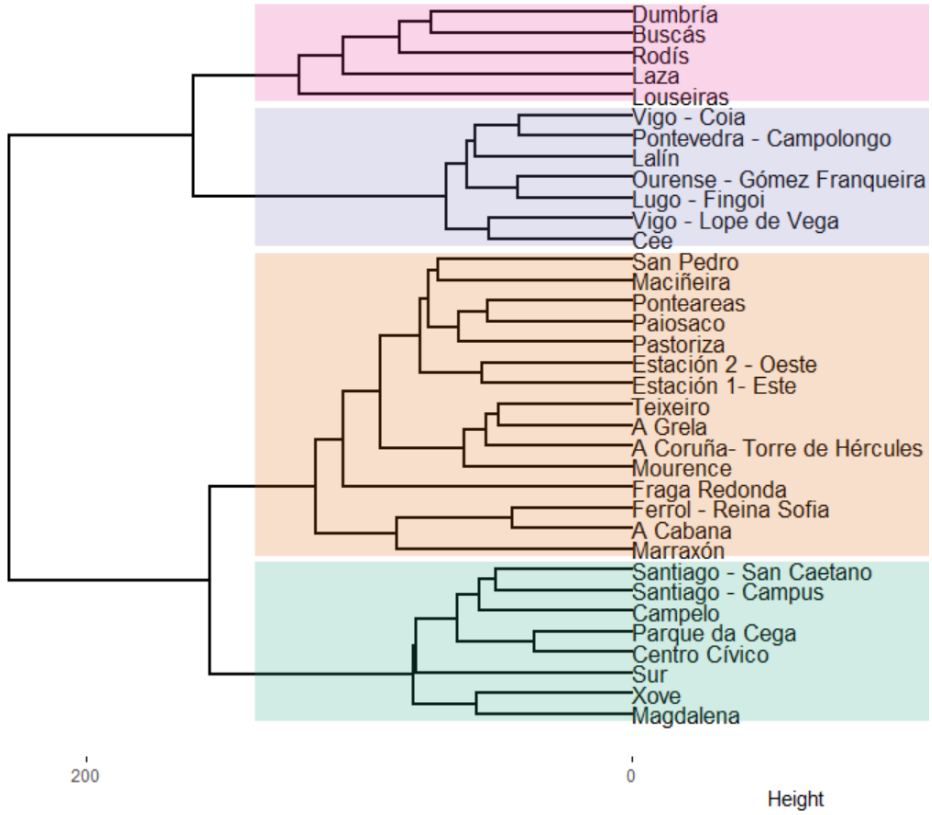

Figura 6.3. Dendograma clustering jerárquico de estaciones según comportamiento semanal utilizando método Ward con distancia DWT.

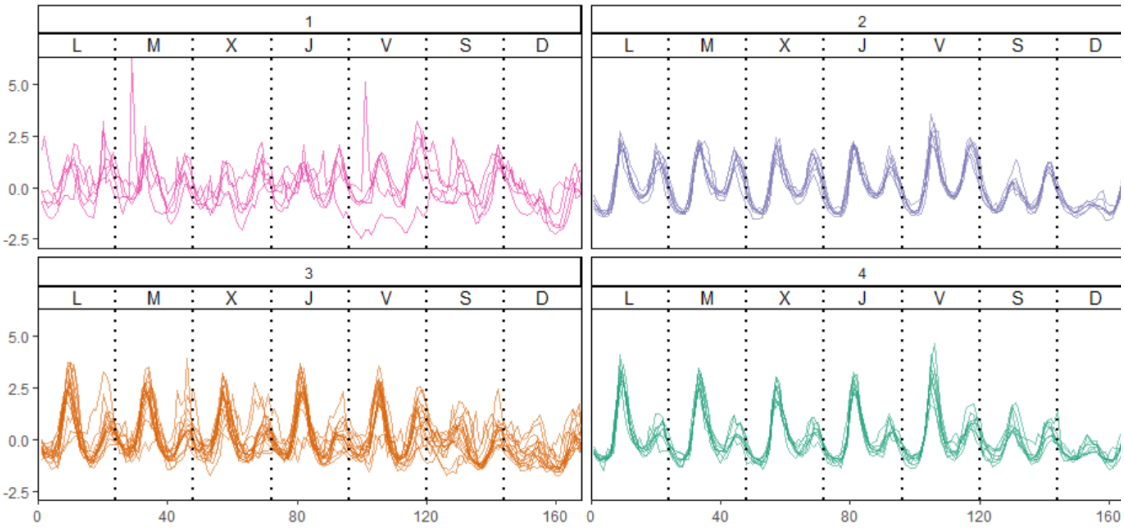

Figura 6.4. Representación gráfica de clústeres resultantes en Figura 6.3.

El primer clúster está compuesto por cuatro estaciones industriales ubicadas en zonas rurales. Al analizar la gráfica, se observa que las estaciones experimentan una variabilidad notable en sus niveles de NOx a lo largo de la semana, sin mostrar un patrón consistente. Por el contrario, los niveles parecen depender del día y de factores aleatorios, lo que resulta en un comportamiento impredecible. Esta variabilidad puede estar influenciada por diversas condiciones, como pueden ser cambios en la producción industrial o actividades específicas que ocurren en la zona.

En cuanto al segundo clúster, está formado principalmente por estaciones de tráfico en zonas urbanas, especialmente en las grandes ciudades de Galicia. Al examinar la gráfica correspondiente a este clúster, se pueden identificar patrones distintivos a lo largo de los días de la semana. Durante los días hábiles, se registra una disminución en las mediciones de NOx en las primeras horas de la madrugada, seguida de un aumento gradual por la mañana que alcanza un máximo de concentración. A media tarde, se produce una disminución parcial, aunque los niveles se mantienen más altos que durante la madrugada. Finalmente, hacia el final de la tarde, se registra otro pico de contaminación, levemente más pequeño que el matutino, seguido de una disminución nocturna. En definitiva, los días laborales se caracterizan por la presencia de dos picos, uno por la mañana y otro por la tarde, siendo ligeramente más alto el pico de la mañana. Los picos de concentración registrados en las horas de la mañana y de la tarde durante los días laborales están directamente relacionados con los horarios en los que la población se desplaza hacia su trabajo o regresa a sus hogares. Además, cabe destacar que el viernes es el día con las mediciones más elevadas de NOx. Durante el fin de semana, se observa una disminución de las mediciones, aunque se mantienen los dos momentos de mayor concentración, mañana y tarde, siendo el de la tarde el más pronunciado. En conclusión, este segundo clúster refleja en sus patrones la influencia del tráfico vehicular y las actividades urbanas en las ciudades.

El tercer clúster está formado mayoritariamente por estaciones industriales, aunque cuenta también con la presencia de estaciones de fondo y tráfico. En este grupo no predomina ningún tipo de zona en concreto, ya que las estaciones se encuentran en zonas urbanas, suburbanas y rurales. Al examinar la gráfica, se observa un patrón característico durante los días laborales. Por la mañana se produce un incremento significativo en todas las estaciones. No obstante, durante el segundo momento de mayor concentración por la tarde, algunas estaciones presentan niveles más altos que otras. Durante los fines de semana, los niveles de NOx son más difusos y no muestran un patrón claro y común entre las estaciones. Es importante destacar que la falta de un patrón claro en los fines de semana puede deberse a una menor actividad industrial y de tráfico, así como por los cambios en los patrones de comportamiento de la población durante esos días.

Por último, en el cuarto clúster predominan las estaciones industriales ubicadas en zonas suburbanas. En la gráfica se observa un comportamiento característico de lunes a viernes. Durante la madrugada se registran los niveles más bajos de contaminación de NOx. Después, por la mañana, se produce un gran pico de concentración, seguido de un descenso que alcanza niveles levemente más altos que los registrados en la madrugada. Posteriormente, se observa un segundo pico por la tarde, aunque con una diferencia notable en comparación con el pico matutino. Finalmente, los niveles vuelven a descender por la noche. Durante el fin de semana, se observa un comportamiento distinto, los niveles de NOx son más bajos, y los picos de concentración tanto por la mañana como por la tarde tienden a igualarse. Se observa bajos niveles de emisión los domingos durante todo el día.

Cabe destacar también que, en el segundo clúster, entre los dos momentos de máxima concentración, los niveles se mantienen más altos que los valores de la madrugada. Sin embargo, entre los dos picos del cuarto clúster, los niveles alcanzan valores similares a los de la madrugada. Esto refleja que en las zonas urbanas existe una persistencia de contaminantes atmosféricos a lo largo del día, mientras que en las zonas suburbanas hay una mayor variabilidad y disminución de los niveles durante ciertos periodos.

En definitiva, el comportamiento de las estaciones ubicadas en zonas rurales tiende a diferir más entre sí que en estaciones ubicadas en zonas urbanas o suburbanas. Esta diferencia puede ser atribuida a varias razones. En primer lugar, las estaciones en zonas rurales pueden estar influenciadas por fuentes de contaminación más específicas y localizadas, como actividades agrícolas, ganadería o procesos industriales específicos. Además, en estas zonas, existe una mayor diferencia en los horarios de desplazamiento. Otra de las razones podría ser las particulares características geográficas y topográficas de las zonas rurales.

Por otro lado, para obtener el comportamiento anual se calcula la media de los registros horarios de cada día, obteniendo el valor promedio de cada día. Este comportamiento permite detectar posibles eventos o estacionalidades presentes en los datos.

A continuación, se representa el comportamiento anual de las estaciones:

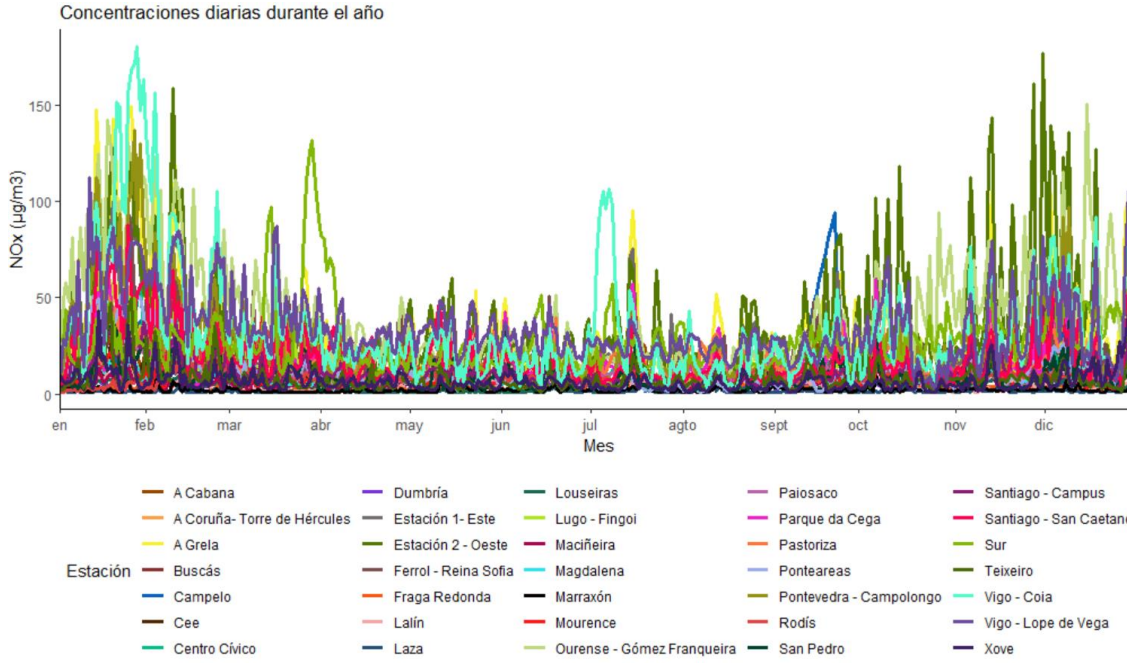

Figura 6.5. Representación gráfica comportamiento anual en las estaciones.

En el caso del comportamiento anual, los resultados pueden no ser tan visibles como en el caso del comportamiento semanal debido a la longitud de las series de tiempo. A pesar de ello, en la gráfica se puede observar una tendencia general de aumento en la concentración del contaminante durante los meses más fríos del año, especialmente de noviembre a febrero. Durante estos meses se presentan situaciones como un mayor tráfico rodado o un aumento en la demanda de calefacción que contribuyen a un incremento de las concentraciones [47]. A partir de marzo las concentraciones tienden a bajar hasta

estabilizarse durante la primavera y el verano. Sin embargo, se puede apreciar que a mediados de julio se presenta un pico de concentración en varias de los puntos de monitoreo. Este incremento podría estar relacionado con una fuerte ola de calor y múltiples incendios que tuvieron lugar en Galicia durante ese periodo.

Cabe destacar el comportamiento anómalo que se produce en la estación Sur, en Lugo, durante la segunda quincena de marzo y la primera de abril. Durante este período se produce una gran concentración del contaminante en la estación. Este hecho, está relacionado con las obras que ADIF realizó en un tramo cercano a la estación durante estas fechas.

Asimismo, destaca también el comportamiento anómalo que se produce durante la primera semana de julio en la estación Vigo – Coia. La alta concentración de NOx durante esta semana coincide con las fiestas populares de Coia, barrio densamente poblado de la ciudad de Vigo. La estación que recoge las mediciones está ubicada en la zona en la que se llevan a cabo los festejos. En esta semana de celebración se produce una gran intensificación de la actividad humana en la zona, además de una mayor contaminación debido al uso de fuentes de emisión adicionales como son los generadores o fuegos artificiales.

MeteoGalicia ha constatado que los dos eventos identificados son los causantes de las variaciones en las concentraciones de NOx en estas dos estaciones.

Para estudiar cómo se clasifican las estaciones según su comportamiento a lo largo del año, se hace un análisis jerárquico aglomerativo con el método Ward, y se usa la distancia euclidiana para calcular la similitud entre series temporales sobre los datos normalizados.

A continuación, se muestran los clústeres obtenidos y se representan en el mapa:

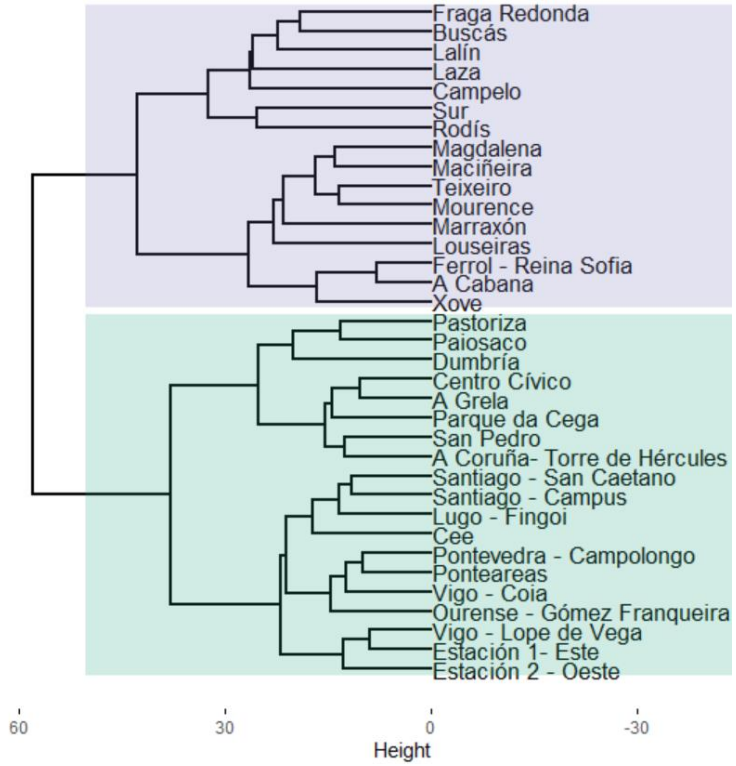

Figura 6.6. Dendograma clustering jerárquico de las estaciones según comportamiento anual utilizando método Ward con distancia euclidiana.

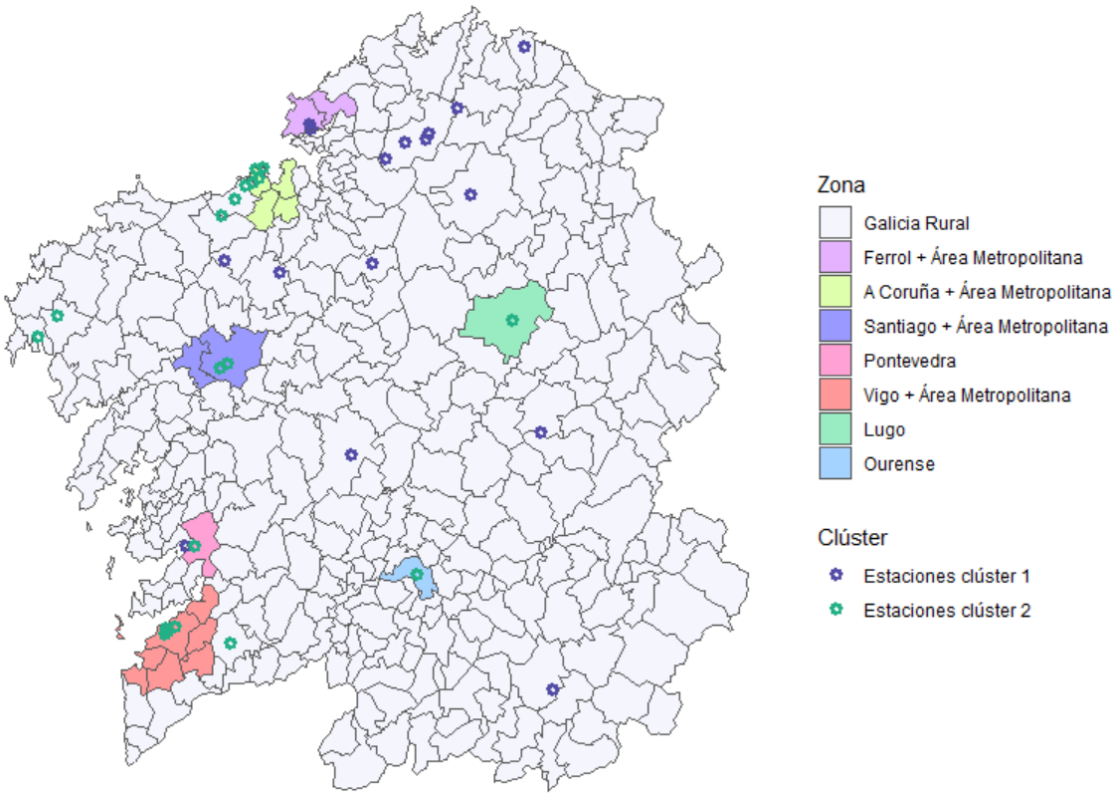

Figura 6.7. Representación en el mapa de los clústeres resultantes en Figura 6.6.

Tal y como podemos observar en el mapa, el primer clúster está formado por estaciones industriales ubicadas en zonas rurales repartidas por todo el territorio de la comunidad, todas ellas pertenecen a la zona Galicia Rural. Además, también se incluyen en este clúster las dos estaciones ubicadas en la zona de Ferrol y su área metropolitana.

Por otro lado, el segundo clúster está compuesto principalmente por estaciones ubicadas en las grandes ciudades y en sus alrededores. Vigo, Pontevedra, Ourense, Lugo, A Coruña y Santiago de Compostela son los núcleos urbanos con mayor densidad de población en Galicia, y comparten fuentes de emisión como pueden ser las calefacciones en los edificios residenciales y comerciales, o mayor tráfico rodado en los meses más fríos. Además, las áreas metropolitanas y los alrededores de las ciudades suelen estar interconectados y presentar similitudes en términos de patrones.

Se puede observar en el extremo izquierdo del mapa que también pertenecen a este segundo clúster la estación de Cee y Dumbría. La estación de Cee, a pesar de pertenecer a la zona Galicia Rural, está tipificada como urbana, por lo que presenta características propias de un entorno urbano. En cuanto a la estación de Dumbría, es de tipo industrial en una zona rural, tendría que realizarse un análisis más detallado para comprender las razones de esta agrupación.

En definitiva, el análisis clúster realizado muestra una clara distinción entre estaciones ubicadas en zonas rurales y estaciones ubicadas en núcleos urbanos. Es importante tener en cuenta que el clima también puede influir en los niveles de contaminación, pero en este caso, los resultados revelan que la influencia de los núcleos urbanos es más

significativa en la distribución de las estaciones. Esto sugiere que las actividades humanas tienen un impacto importante en el comportamiento de los niveles de contaminación.

Además, también se puede observar, que las estaciones cercanas a zonas de aglomeración, específicamente Vigo y A Coruña, a pesar de pertenecer a la zona Galicia Rural, muestran comportamientos similares a las estaciones ubicadas en las zonas asignadas a cada ciudad.

También es importante agrupar las estaciones por la magnitud de sus mediciones de NOx para comprender cuáles son las mayores fuentes de emisión del contaminante. Para ello, se aplican los algoritmos partitivos estudiados anteriormente (k-means y PAM). En este caso, debido a que el conjunto de datos es relativamente pequeño, no se utiliza el algoritmo CLARA. Se usan estos algoritmos junto con la distancia euclidiana en las series sin normalizar.

Para determinar el número óptimo de clústeres se utilizó el "método del "codo". Este método consiste en graficar la relación entre la suma de los cuadrados dentro de los clústeres (WSS) y el número de clústeres, y buscar el punto de inflexión en la curva. El punto de inflexión se conoce como el "codo" y representa el número óptimo de clústeres a utilizar. Los resultados obtenidos fueron los siguientes:

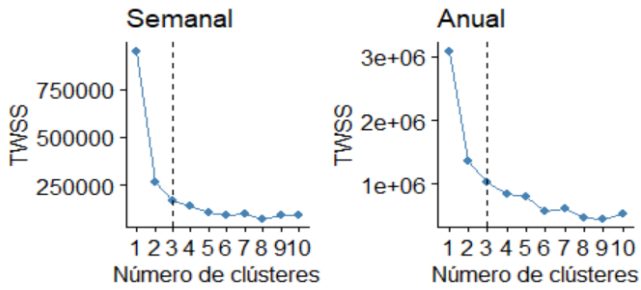

Figura 6.8. Representación del número de clústeres óptimos según "método del codo".

Observando las gráficas se ha llegado a la conclusión de que el número óptimo de clústeres es tres, tanto para el comportamiento semanal como para el anual.

Una vez obtenidos los resultados de los algoritmos aplicados a los dos conjuntos de datos, se procedió a evaluarlos mediante diferentes índices de validación de clústeres. En concreto, se evaluó el índice de Silhouette (máximo), el índice de Dunn (máximo), el índice COP (mínimo), el índice Davies-Bouldin (mínimo) y Davies-Bouldin (mínimo) modificado y el índice de Calinski-Harabasz (máximo). Los resultados obtenidos se muestran en la tabla:

Variación	**Algoritmo**	**Sil (máx.)**	**D (máx.)**	**COP (mín.)**	**DB (mín.)**	**CH (máx.)**
Semanal	k-means	0.46	0.11	0.15	0.69	33.50
	PAM	0.29	0.06	0.36	1.05	18.89
Anual	k-means	0.53	0.43	0.27	1.31	18.40
	PAM	0.33	0.17	0.25	1.29	19.31

Tabla 6.5. Evaluación clustering con algoritmo k-means y PAM en comportamiento semanal y anual.

A pesar de que los resultados son similares en ambos algoritmos, se observa que, en general, k-means tuvo un mejor rendimiento que el algoritmo PAM en el comportamiento semanal. En cuanto al comportamiento anual, ambos algoritmos presentan un rendimiento similar.

Los clústeres obtenidos tras aplicar el algoritmo k-means coinciden para el comportamiento semanal y anual y son los siguientes:

- **Clúster 1**: San Pedro, Paiosaco, Rodís, Buscás, Teixeiro, Dumbría, Cee, Fraga Redonda, Maciñeira, Magdalena, Marraxón, Louseiras, Mourence, Xove, Lalín, Campelo y Laza.
- **Clúster 2:** A Cabana, Centro Cívico, Ferrol – Reina Sofía, A Coruña – Torre de Hércules, Parque da Cega, Pastoriza, Santiago – Campus, Santiago – San Caetano, Ponteareas, Lugo – Fingoi.
- **Clúster 3:** A Grela, Estación 1- Este, Estación 2- Oeste, Vigo – Coia, Vigo – Lope de Vega, Pontevedra- Campolongo, Ourense – Gómez Franqueira, Sur.

Figura 6.9. Composición de los clústeres resultantes con algoritmo k-means para el comportamiento semanal y anual.

Los resultados con el algoritmo PAM presentan pequeñas diferencias según el comportamiento. En el comportamiento semanal, se observó que la estación Ponteareas que estaba en el segundo clúster, pasó a formar parte del primero. Mientras que en el comportamiento anual la estación Paiosaco y Sur, del primer y tercer clúster respectivamente, pasaron a formar parte del segundo.

A continuación, se presentan los resultados gráficos obtenidos para los dos tipos de comportamiento. En estas representaciones cada clúster está representado respetando la asignación de colores de Figura 6.9.

Resultados con el algoritmo k-means para el comportamiento semanal:

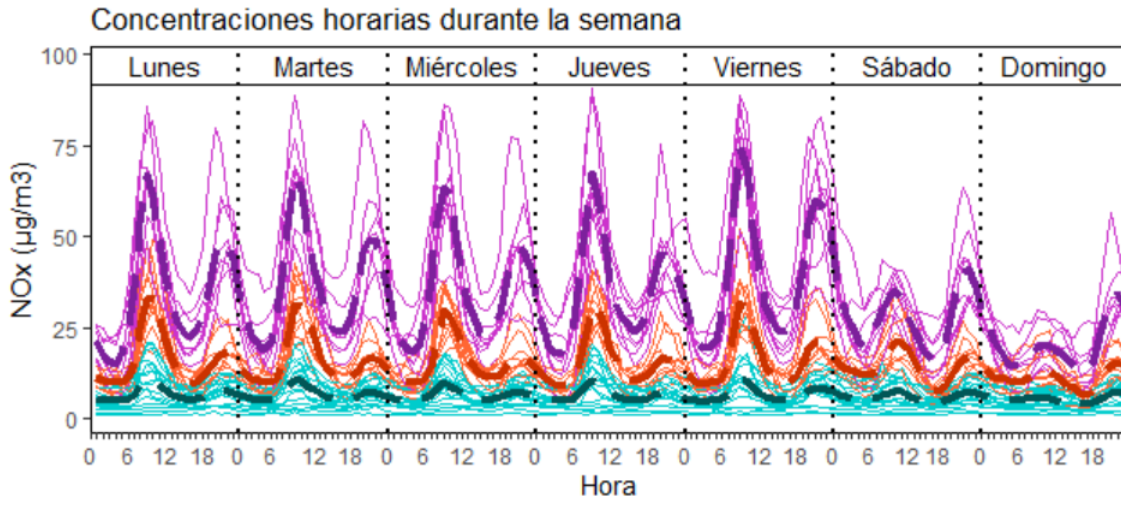

Figura 6.10. Representación gráfica del comportamiento semanal de los clústeres resultantes y sus respectivos centroides obtenidos con k-means.

Resultados con el algoritmo PAM para el comportamiento semanal:

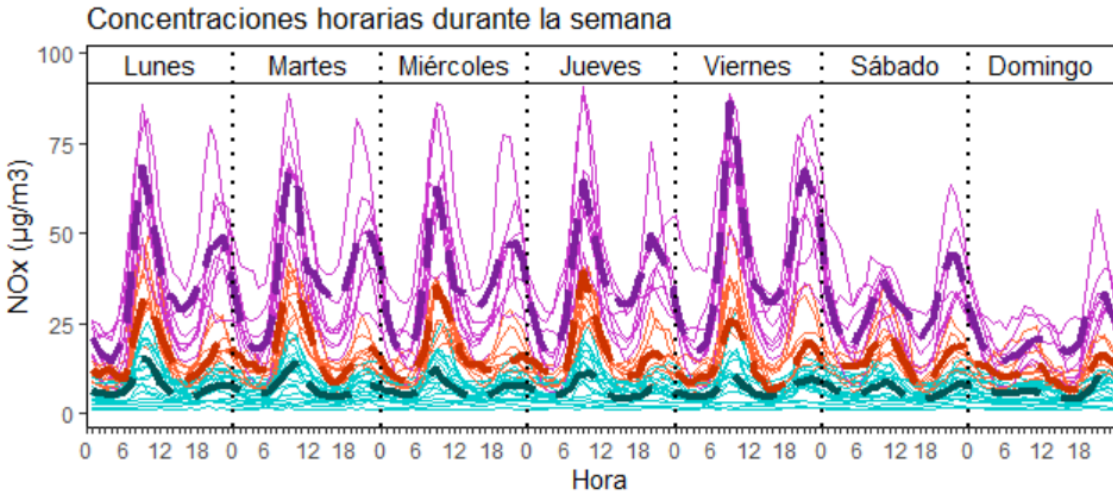

Figura 6.11. Representación gráfica del comportamiento semanal de los clústeres resultantes y sus respectivos medoides obtenidos con PAM.

El medoide del algoritmo PAM es la serie que minimiza la suma de las distancias euclidianas a todas las demás series del mismo clúster. Es, por tanto, la serie de la estación más representativa del clúster, ya que tiene las características más similares a las demás estaciones del mismo clúster en términos de distancia euclidiana. En este caso, el medoide del clúster 1 es la estación Mourence, el del clúster 2 es A Coruña – Torre de Hércules y el del clúster 3 es la estación de Vigo – Lope de Vega. En el caso de k-means, el centroide se define como la serie que minimiza las distancias entre todos los puntos del clúster y el centroide. Por lo tanto, no es posible identificar una estación representativa en k-means de la misma manera que se hace con el algoritmo PAM.

En la siguiente tabla podemos observar un análisis descriptivo de los correspondientes centroides y medoides. Los resultados revelan que, como también se puede apreciar visualmente en los gráficos, centroides y medoides son bastante similares entre sí, pero difieren significativamente entre clústeres, lo que indica que los resultados obtenidos son estables y robustos.

	Centroide 1	Medoide 1	Centroide 2	Medoide 2	Centroide 3	Medoide 3
Mínimo	4,290	4,019	6,867	6,502	12,938	14,525
1er cuartil	5,378	5,380	10,877	10,956	21,619	22,027
Mediana	6,071	6,410	13,196	13,616	28,852	32,784
Media	6,502	6,907	15,054	15,413	32,811	34,468
3er cuartil	7,316	7,788	16,679	17,740	41,233	42,674
Máximo	10,868	15,033	33,648	39,170	74,368	86,134

Tabla 6.6. Análisis descriptivo para centroides y medoides del comportamiento semanal.

En cuanto al comportamiento anual, los resultados con el algoritmo k-means y PAM son los siguientes:

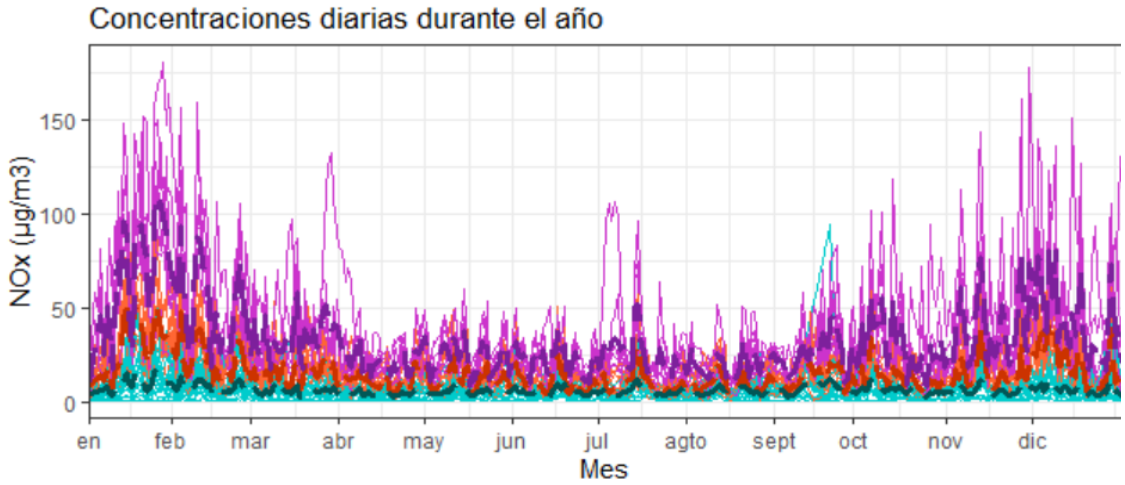

Figura 6.12. Representación gráfica del comportamiento anual de los clústeres resultantes y sus respectivos centroides obtenidos con k-means.

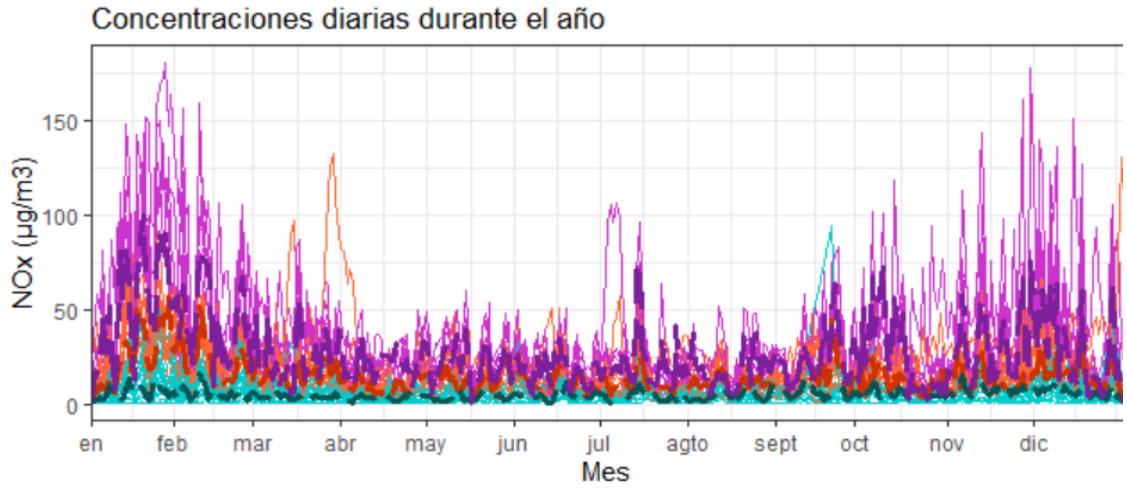

Figura 6.13. Representación gráfica del comportamiento anual de los clústeres resultantes y sus respectivos centroides obtenidos con PAM.

El análisis descriptivo correspondiente a cada centroide y medoide es el siguiente:

	Centroide 1	Medoide 1	Centroide 2	Medoide 2	Centroide 3	Medoide 3
Mínimo	2,963	1,000	3,323	1,150	8,196	2,175
1er cuartil	4,766	3,333	8,658	8,363	20,284	13,825
Mediana	5,976	4,833	12,323	12,292	26,752	21,571
Media	6,501	5,383	15,052	14,367	32,798	27,127
3er cuartil	7,553	6,917	18,716	17,229	38,573	34,658
Máximo	17,075	16,333	53,729	57,208	110,385	100,208

Tabla 6.7. Análisis descriptivo para centroides y medoides del comportamiento anual.

El análisis clustering realizado revela que los valores de los centroides son ligeramente mayores que los de los medoides en cada clúster. Esta diferencia sugiere que el centroide tiende a ser más sensible a los valores extremos. Es el caso específico de la estación Sur, los valores anómalos que presenta durante marzo y abril, pudieron haber afectado los resultados del algoritmo k-means, que la clasifica en el tercer clúster. En cuanto a los medoides, las estaciones representativas de casa clúster han cambiado, la estación representativa del primer clúster es Maciñeira, la del segundo Centro Cívico y la del tercero la Estación 1 – Este.

En resumen, los resultados obtenidos mediante los algoritmos k-means y PAM usando la distancia euclidiana sugieren que existe una clara agrupación de las estaciones de acuerdo con su magnitud, evidenciando la existencia de tres clústeres bien definidos. El primero de ellos presenta los niveles más bajos de NOx, mientras que el segundo clúster muestra niveles medios y el tercero presenta los niveles más altos de todos. Analizando el tipo de estaciones que compone cada clúster, se puede apreciar que el clúster 1, está compuesto, a excepción de la estación de San Pedro, por estaciones pertenecientes a la zona Galicia Rural. La mayoría de estas estaciones son industriales en zonas rurales o suburbanas, lo que podría sugerir una menor actividad en el tráfico y una mayor influencia de otras fuentes de emisión, como la actividad industrial o la agricultura. Por su parte, el segundo clúster está compuesto por estaciones industriales y de tráfico, tanto en zonas urbanas como suburbanas. Estas estaciones podrían estar asociadas a áreas con mayor densidad de población o más industrializadas, lo que implica una mayor presencia de vehículos y emisiones en el aire. Por último, el clúster 3 está compuesto, a excepción de la estación Sur, por estaciones ubicadas en zonas urbanas y metropolitanas, lo que sugiere una mayor actividad vehicular en dichas áreas.

Esto sugiere que la ubicación (ya sea en zonas metropolitanas, rurales o urbanas) y el tipo de estación influye directamente en la magnitud de las concentraciones de NOx en el aire, y que la actividad vehicular, industrial y agrícola son factores determinantes en la emisión de este contaminante.

Para poder visualizar geográficamente las estaciones que componen cada clúster, se ha realizado un mapa en donde se han ubicado las estaciones pertenecientes a cada grupo, de acuerdo con la clasificación obtenida mediante el algoritmo k-means. Se ha optado por utilizar los resultados obtenidos por este algoritmo, para simplificar la representación de las estaciones de cada clúster, dado que este algoritmo ha sido consistente en la agrupación de las estaciones en los dos comportamientos estudiados.

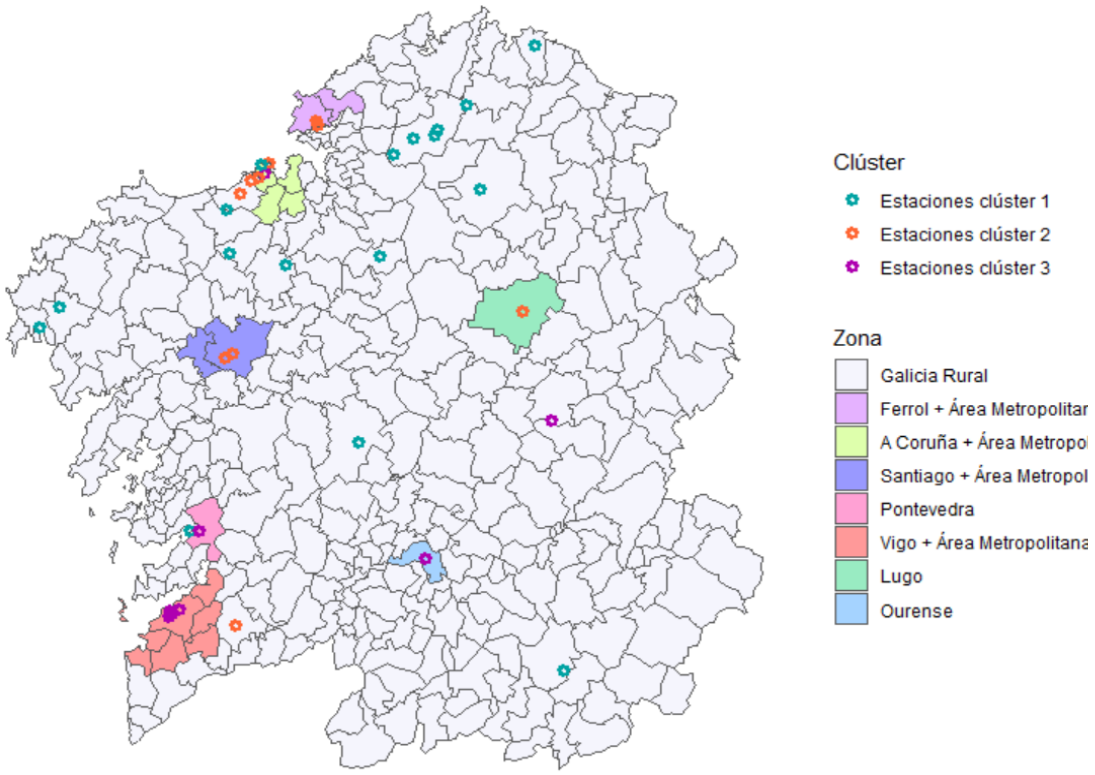

Figura 6.14. Representación en el mapa de los clústeres resultantes de Figura 6.9.

De acuerdo con los resultados obtenidos, en Lugo, la estación Sur, correspondiente a la zona Galicia Rural, pertenece al tercer clúster, que presenta los niveles más elevados de NOx. Estos valores sugieren que la estación Sur debería incluirse en una zona complementaria a las ya existentes con el fin de poder evaluar la calidad del aire de dicho territorio de una forma más específica y adecuada. Sin embargo, es importante destacar que los resultados podrían haberse visto afectados por el mencionado comportamiento anómalo de la estación Sur en los meses de marzo y abril debido a las obras en zonas cercanas. Para comprobar que realmente es adecuada la consideración de esta zona, se ha comprobado en el informe de la calidad del aire de Galicia en 2021 [45], que en el año 2021 la estación Sur también presentó valores altos.

Para otros contaminantes como el SO2, existe una zona específica denominada zona Oural, a la cual pertenece la estación Sur. Además, en 2021 se llevó a cabo una revisión de la zonificación para evaluar la calidad del aire y se incluyó también dicha zona en el estudio de las materias particuladas PM10 y PM25. Por lo que sería interesante incluir también esta zona en el estudio de NOx.

Lo mismo sucede con la zona de Arteixo, en la limitación con la zona de A Coruña. El área de Arteixo se considera para la evaluación de SO2, PM10 y PM2.5. También sería interesante incluirla en la zonificación del NOx.

También se puede apreciar viendo el mapa que todas las estaciones del segundo clúster se encuentran en zonas que incluyen ciudades y áreas metropolitanas o en sus inmediaciones. Este clúster tiene niveles intermedios de NOx en comparación con los otros dos y está conformado por estaciones industriales y de tráfico influenciadas por la presencia de vehículos y emisiones. Sería interesante ampliar dichas zonas para incluir también las estaciones de este clúster que actualmente se encuentran en la zona Galicia

Rural. Específicamente este hecho ocurre en las inmediaciones de la zona Vigo y su área metropolitana, donde podría realizarse una redefinición para conseguir tener una zonificación más precisa y completa.

A continuación, se propone una nueva zonificación que incluye la zona Oural y Arteixo, mencionadas anteriormente, y la redefinición de la zona de Vigo:

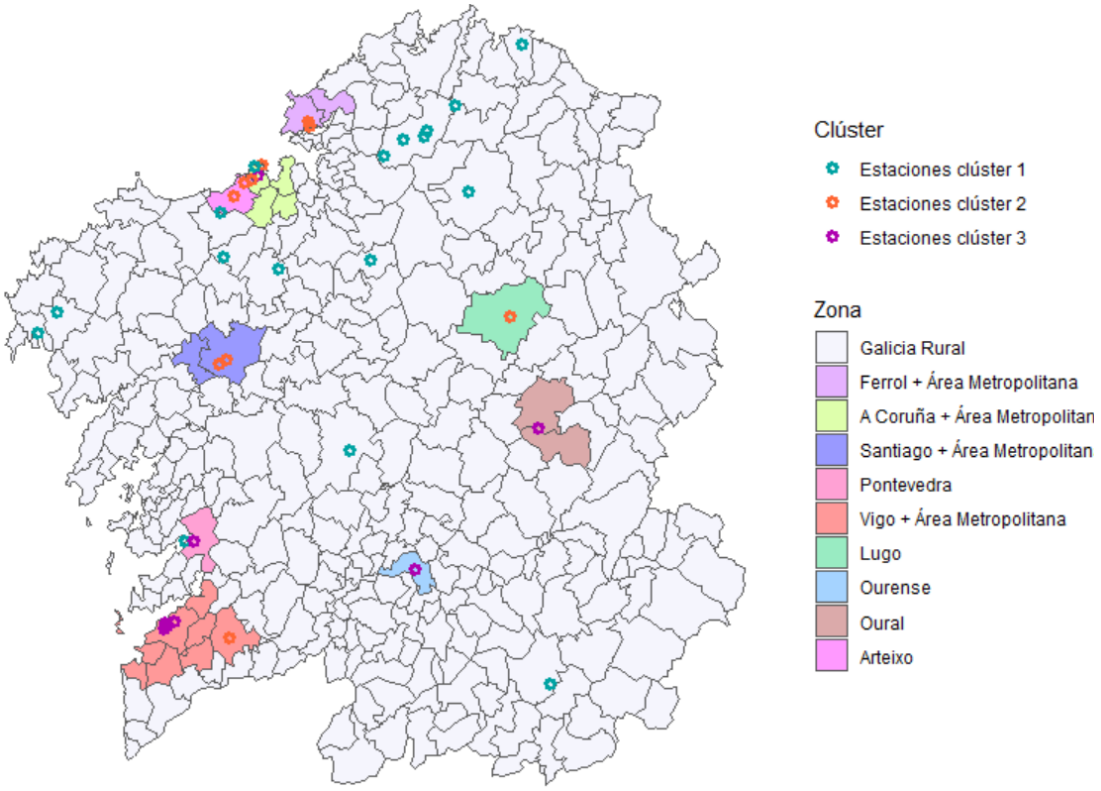

Figura 6.15. Representación en el mapa de los clústeres resultantes de Figura 6.9. con la zonificación propuesta.

De las 39 estaciones de monitoreo de NOx existentes en 2022, solo 4 cumplen los requisitos necesarios para la evaluación de la calidad del aire para proteger la vegetación. El hecho de que no se evalúe este contaminante para proteger también la salud humana, hace que todas estas estaciones, excepto las 4 anteriormente mencionadas, se puedan considerar redundantes, ya que no se presta suficiente atención a los niveles de NOx que recogen.

Una evaluación adecuada de los niveles de NOx en Galicia para proteger la salud humana y el entorno natural es crucial. La instalación de más estaciones de monitoreo, especialmente en las zonas con mayor aglomeración de población y sus alrededores, y la implantación de una zonificación específica para este contaminante son medidas clave para mejorar la calidad del aire y garantizar un ambiente saludable, tanto para las generaciones presentes como para las futuras.

Capítulo 7. Conclusiones y trabajo futuro

En este estudio, se ha llevado a cabo una recopilación de información existente en la literatura sobre técnicas de análisis clúster aplicadas a series de tiempo. Esta etapa tuvo como objetivo recoger los algoritmos, métricas de similitud y métodos de evaluación más empleados en este campo.

Para llevar a cabo el análisis del comportamiento anual y semanal de las emisiones de NOx en Galicia durante el año 2022, se utilizó la información previamente recopilada. Esto permitió seleccionar y aplicar los métodos más adecuados para el análisis.

Según los resultados obtenidos con el algoritmo jerárquico de clustering usando el método Ward, existe una clara distinción entre estaciones ubicadas en zonas rurales y estaciones ubicadas en núcleos urbanos. El comportamiento de las estaciones en zonas rurales tiende a diferir más entre sí, posiblemente debido a la influencia de fuentes de contaminación mas específicas y localizadas.

Los resultados de los algoritmos k-means y PAM usados para clasificar las series de tiempo según su magnitud, han evidenciado que las estaciones ubicadas en áreas metropolitanas y grandes ciudades presentan niveles más altos de NOx. Esto sugiere que el tráfico vehicular y las actividades humanas desempeñan un papel significativo en los niveles de contaminación de NOx en su entorno.

El riesgo que este contaminante supone para la salud y la evidencia de sus altos niveles en zonas urbanas, sugieren que, aunque la normativa por el momento no lo exija, sería conveniente realizar una evaluación de los niveles de NOx para garantizar un ambiente seguro que no suponga un riesgo para la salud humana. Por lo que, además de evaluar los registros de las estaciones que ya recogen los niveles del contaminante, sería apropiada la instalación y evaluación de más estaciones en el futuro.

Asimismo, se sugiere como una futura línea de investigación la inclusión de variables adicionales en el estudio, como las condiciones meteorológicas. Considerar factores meteorológicos como la velocidad y dirección del viento, la temperatura, la humedad o las precipitaciones, permitirá tener una visión más completa. Además, también se plantea la recopilación de datos de varios años para analizar la evolución de los niveles de NOx a lo largo del tiempo en Galicia.

Bibliografía

[1] M. Halkidi, *On Clustering Validation Techniques*, 2001.

[2] S. Aghabozorgi, A. Seyed Shirkhorshidi, y T. Ying Wah, *Time-series clustering – A decade review*, Inf Syst, vol. 53, pp. 16-38, oct. 2015, doi: 10.1016/J.IS.2015.04.007.

[3] E. Keogh y J. Lin, *Clustering of time-series subsequences is meaningless: implications for previous and future research*, Knowl Inf Syst, vol. 8, pp. 154-177, 2004, doi: 10.1007/s10115-004-0172-7.

[4] S. Zolhavarieh, S. Aghabozorgi, y Y. W. Teh, *A review of subsequence time series clustering*, ScientificWorldJournal, pp. 1-3, 2014, Accedido: 26 de febrero de 2023. [En línea]. Disponible en: https://doi.org/10.1155/2014/312521

[5] P. Esling y C. Agon, *Time-series data mining*, ACM Comput Surv, vol. 45, n.º 1, 2012, doi: 10.1145/2379776.2379788.

[6] A. Fakhrazari y H. Vakilzadian, *A survey on time series data mining*, en IEEE International Conference on Electro Information Technology, 2017. doi: 10.1109/EIT.2017.8053409.

[7] H. Ren, X. Liao, Z. Li, y A. AI-Ahmari, *Anomaly detection using piecewise aggregate approximation in the amplitude domain*, Applied Intelligence, vol. 48, n.º 5, 2018, doi: 10.1007/s10489-017-1017-x.

[8] E. Keogh, K. Chakrabarti, M. Pazzani, y S. Mehrotra, *Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases*, Knowl Inf Syst, vol. 3, n.º 3, 2001, doi: 10.1007/pl00011669.

[9] B. Lkhagva, Y. Suzuki, y K. Kawagoe, *Extended SAX: Extension of symbolic aggregate approximation for financial time series data representation*, DEWS2006 4A-i8, January, 2006.

[10] J. Lin, E. Keogh, S. Lonardi, y B. Chiu, *A symbolic representation of time series, with implications for streaming algorithms*, en Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, DMKD '03, 2003. doi: 10.1145/882082.882086.

[11] Y. Sun, J. Li, J. Liu, B. Sun, y C. Chow, *An improvement of symbolic aggregate approximation distance measure for time series*, Neurocomputing, vol. 138, 2014, doi: 10.1016/j.neucom.2014.01.045.

[12] J. Han, M. Kamber, y J. Pei, *10 – Cluster Analysis: Basic Concepts and Methods*, en Data Mining: Concepts and Techniques, Elsevier Inc., 2012, pp. 443-495.

[13] A. Bagnall y G. Janacek, *Clustering time series with clipped data*, Mach Learn, vol. 58, n.º 2-3, 205, doi: 10.1007/s10994-005-5825-6.

[14] X. Wang et al., *Experimental comparison of representation methods and distance measures for time series data*, Data Min Knowl Disc, vol. 26, pp. 5-9, 2013, doi: 10.1007/s10618-012-0250-5.

[15] E. J. Keogh y M. J. Pazzani, *Derivative Dynamic Time Warping*, Proceedings of the 2001 SIAM International Conference on Data Mining (SDM), pp. 1-11, 2001, doi: 10.1137/1.9781611972719.1.

[16] L. Liu, W. Li, y H. Jia, *Method of time series similarity measurement based on dynamic time warping*, Computers, Materials and Continua, vol. 57, n.º 1, pp. 97-106, 2018, doi: 10.32604/cmc.2018.03511.

[17] C. Cassisi, P. Montalto, M. Aliotta, A. Cannata, y A. Pulvirenti, *Similarity Measures and Dimensionality Reduction Techniques for Time Series Data Mining*, en Advances in Data Mining Knowledge Discovery and Applications, 2012. doi: 10.5772/49941.

[18] L. Chen, M. T. Özsu, y V. Oria, *Robust and fast similarity search for moving object trajectories*, en Proceedings of the ACM SIGMOD International Conference on Management of Data, 2005. doi: 10.1145/1066157.1066213.

[19] V. Niennattrakul y C. A. Ratanamahatana, *Clustering multimedia data using time series*, en Proceedings - 2006 International Conference on Hybrid Information Technology, ICHIT 2006, 2006. doi: 10.1109/ICHIT.2006.253514.

[20] J. E. Gentle, L. Kaufman, y P. J. Rousseuw, *Finding Groups in Data: An Introduction to Cluster Analysis.*, Biometrics, vol. 47, n.º 2, p. 788, jun. 1991, doi: 10.2307/2532178.

[21] R. T. Ng y J. Han, *CLARANS: A method for clustering objects for spatial data mining*, IEEE Trans Knowl Data Eng, vol. 14, n.º 5, 2002, doi: 10.1109/TKDE.2002.1033770.

[22] S. Sreedhar Kumar, M. Madheswaran, B. A. Vinutha, H. Manjunatha Singh, y K. V. Charan, *A brief survey of unsupervised agglomerative hierarchical clustering schemes*, Progress in Color, Colorants and Coatings, vol. 8, n.º 1, pp. 1-7, 2018, doi: 10.14419/ijet.v8i1.15803.

[23] A. Dogan y D. Birant, *K-centroid link: a novel hierarchical clustering linkage method*, Applied Intelligence, vol. 52, n.º 5, 2022, doi: 10.1007/s10489-021-02624-8.

[24] P. Govender y V. Sivakumar, *Application of k-means and hierarchical clustering techniques for analysis of air pollution: A review (1980–2019),* Atmos Pollut Res, vol. 11, n.º 1, pp. 40-56, ene. 2020, doi: 10.1016/J.APR.2019.09.009.

[25] J. H. Ward, *Hierarchical Grouping to Optimize an Objective Function*, J Am Stat Assoc, vol. 58, n.º 301, 1963, doi: 10.1080/01621459.1963.10500845.

[26] B. Moseley y J. R. Wang, *Approximation bounds for hierarchical clustering: Average linkage, bisecting K-means, and Local Search*, en Advances in Neural Information Processing Systems, 2017.

[27] J. Han, M. Kamber, y J. Pei, *Data Mining (Third Edition)*. 2012. doi: https://doi.org/10.1016/B978-0-12-381479-1.00016-2.

[28] W. Wang y Y. Zhang, *On fuzzy cluster validity indices*, Fuzzy Sets Syst, vol. 158, n.º 19, pp. 2095-2117, oct. 2007, doi: 10.1016/J.FSS.2007.03.004.

[29] J. C. Dunn, *A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters*, Journal of Cybernetics, vol. 3, n.º 3, 1973, doi: 10.1080/01969727308546046.

[30] J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function *Algorithms*, Pattern Recognition with Fuzzy Objective Function Algorithms, 1981, doi: 10.1007/978-1-4757-0450-1.

[31] Yang, M. S. (1993). A survey of fuzzy clustering. *Mathematical and Computer Modelling*, *18*(11). https://doi.org/10.1016/0895-7177(93)90202-A

[32] T. Warren Liao, *Clustering of time series data - A survey*, Pattern Recognit, vol. 38, n.º 11, 2005, doi: 10.1016/j.patcog.2005.01.025.

[33] S. Liang, D. Han, y Y. Yang, *Cluster validity index for irregular clustering results*, Applied Soft Computing Journal, vol. 95, 2020, doi: 10.1016/j.asoc.2020.106583.

[34] F. Ros, R. Riad, y S. Guillaume, *PDBI: A partitioning Davies-Bouldin index for clustering evaluation*, Neurocomputing, vol. 528, 2023, doi: 10.1016/j.neucom.2023.01.043.

[35] P. J. Rousseeuw, *Silhouettes: A graphical aid to the interpretation and validation of cluster analysis*, J Comput Appl Math, vol. 20, n.º C, 1987, doi: 10.1016/0377-0427(87)90125-7.

[36] J. C. Dunn, *Well-separated clusters and optimal fuzzy partitions*, Journal of Cybernetics, vol. 4, n.º 1, pp. 95-104, ene. 1974, doi: 10.1080/01969727408546059.

[37] C. E. Ben Ncir, A. Hamza, y W. Bouaguel, *Parallel and scalable Dunn Index for the validation of big data clusters*, Parallel Comput, vol. 102, 2021, doi: 10.1016/j.parco.2021.102751.

[38] I. Gurrutxaga et al., *SEP/COP: An efficient method to find the best partition in hierarchical clustering based on a new cluster validity index*, Pattern Recognit, vol. 43, n.º 10, 2010, doi: 10.1016/j.patcog.2010.04.021.

[39] A. José-García y W. Gómez-Flores, *A survey of cluster validity indices for automatic data clustering using differential evolution*, en GECCO 2021 - Proceedings of the 2021 Genetic and Evolutionary Computation Conference, 2021. doi: 10.1145/3449639.3459341.

[40] D. L. Davies y D. W. Bouldin, *A Cluster Separation Measure*, IEEE Trans Pattern Anal Mach Intell, vol. PAMI-1, n.º 2, 1979, doi: 10.1109/TPAMI.1979.4766909.

[41] T. Caliński y J. Harabasz, *A Dendrite Method Foe Cluster Analysis*, Communications in Statistics, vol. 3, n.º 1, 1974, doi: 10.1080/03610927408827101.

[42] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, y I. Perona, *An extensive comparative study of cluster validity indices*, Pattern Recognit, vol. 46, n.º 1, 2013, doi: 10.1016/j.patcog.2012.07.021.

[43] *Calidad del aire ambiente (exterior) y salud.* Accedido: 25 de mayo de 2023[En línea]. Disponible en: https://www.who.int/es/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health

[44] *Real Decreto 102/2011, de 28 de enero, relativo a la mejora de la calidad del aire.* Boletín Oficial del Estado, 25, de 29 de enero de 2011. Disponible en: https://www.boe.es/eli/es/rd/2011/01/28/102

[45] *Informe anual Calidade do aire de Galicia 2021*. Accedido: 17 de mayo de 2023. [En línea]. Disponible en: https://www.meteogalicia.gal/datosred/infoweb/caire/informes/ANUAL/ES/InformeAnual2021.pdf

[46] *CRAN - Package dtwclust*. https://cran.r-project.org/web/packages/dtwclust/index.html (accedido 18 de mayo de 2023).

[47] M. Ivanovski, K. Alatič, D. Urbancl, M. Simonič, D. Goričanec, y R. Vončina, *Assessment of Air Pollution in Different Areas (Urban, Suburban, and Rural) in Slovenia from 2017 to 2021*, Atmosphere (Basel), vol. 14, n.º 3, p. 578, mar. 2023, doi: 10.3390/ATMOS14030578/S1

Anexo:

Anexo A. Código utilizado para realizar el análisis

Se cargan las librerías necesarias

```
library(readxl)
library(imputeTS)
library(lubridate)
library(dplyr)
library(tidyr)
library(ggplot2)
library(factoextra)
library(gridExtra)
library(dtwclust)
library(mapSpain)
library(readr)
library(dtw)
```

Se cargan los datos

```
nox <- read_excel("ruta.xlsx",
                  col_types = c("text", "numeric", "numeric", "numeric", "numeric", "
numeric", "numeric", "numeric", "numeric", "numeric", numeric", "numeric", "numeric",
"numeric", "numeric", "numeric", "numeric", "numeric", "numeric", "numeric", "numeric
", "numeric", "numeric", "numeric", "numeric", "numeric", "numeric", "numeric", "nume
ric", "numeric", "numeric", "numeric", "numeric", "numeric", "numeric", "numeric", "n
umeric", "numeric"))
```

Análisis descriptivo

```
#Resumen descriptivo de los datos
summary(nox[,-1])

#Se eliminan las estaciones de Xinzo de Limia y Est - Ou
nox<-nox[,-c(12,37)]

#Se imputa con el metodo Kalman los valores perdidos
datos<-as.data.frame(nox[,-1])
for (i in 1:ncol(datos)) {
  datos[,i]<-as.numeric(as.matrix(na_kalman(datos[,i])))
}

#Se pasan las fechas a formato Date en la variable fecha_hora
nox$...1<-as.POSIXct(nox$...1, format="%d/%m/%Y %H:%M")
datos$fecha_hora <- ymd_hms(nox$...1)

#Se pasan los datos a formato largo
datos_long <- datos %>%
  pivot_longer(cols = -fecha_hora, names_to = "Estación", values_to = "valor") %>%
  mutate(dia_semana = factor(weekdays(fecha_hora),levels = c("lunes","martes","miérco
les","jueves","viernes","sábado","domingo"))) %>%
  mutate(dia_mes=day(fecha_hora))%>%
  mutate(hora = hour(fecha_hora)) %>%
  mutate(mes = month(fecha_hora))

#Se crea una paleta de colores
colores<-c("#f5a4a4","#784d4d","#872f2f","#de4343","#fa1616","#050101","#f55916","#fc
763d","#fa9e48","#4d2500","#994a00","#f7f12f","#969314","#a9e61c","#82b806","#4b6907"
,"#547801","#bdd97c","#04bd88","#55facb","#1a6952","#01422f","#2de0ed","#0963b8","#27
4f75","#9baceb","#6b4b9c","#7837db","#3c1f69","#8a1e76","#ed21c8","#b867a9","#a80752"
,"#ff0051","#736c6e")
```

```
#Grafico de los datos originales
ggplot(datos_long, aes(x = fecha_hora, y = valor, color = Estación)) +
  geom_line() +
  labs(x = "Hora", y = "NOx (µg/m3)") +
  theme_classic()+
  theme(axis.text.x = element_text(size = 10))+
  theme(axis.ticks = element_line(size = 0.5, colour = "black"))+
  theme(legend.position = "bottom")+
  scale_color_manual(values=colores)
```

Comportamiento semanal

```
#Se calcula el comportamiento semanal
semanal<- datos_long %>%
  group_by(Estación, dia_semana, hora) %>%
  summarise(media = mean(valor)) %>%
  group_by(Estación, hora)

#Se crea el grafico  del comportamiento semanal
dias_horas<-c("0","","","","","","6","","","","","","12","","","","","","18","","","","","","0","","","","","","6","","","","","","12","","","","","","18","","","","","","0","","","","","","6","","","","","","12","","","","","","18","","","","","","0","","","","","","6","","","","","","12","","","","","","18","","","","","","0","","","","","","6","","","","","","12","","","","","","18","","","","","","0","","","","","","6","","","","","","12","","","","","","18","","","","","","0","","","","","","6","","","","","","12","","","","","","18","","","","","","")
ggplot(semanal, aes(x = hora +  (as.numeric(as.factor(dia_semana)) - 1) * 24, y = media, color = Estación)) +
  geom_line(linewidth=1) +
  scale_x_continuous(breaks = seq(0,( 7 * 24-1), 1), labels = dias_horas, expand = c(0, 0)) +
  labs(x = "Hora", y = "NOx (µg/m3)") +
  theme_classic()+
  theme(axis.text.x = element_text(size = 10))+
  theme(axis.ticks = element_line(size = 0.5, colour = "black"))+
  geom_vline(xintercept = c(0,24, 48, 72, 96, 120, 144), color = "black",size=1, linetype = "dotted")+
  theme(legend.position = "bottom")+
  scale_color_manual(values=colores)+
  geom_hline(yintercept = max(semanal$media)+1 , color = "black")+
  annotate("text", x = (0 + 24)/2, y = max(semanal$media) * 1.07, label = "Lunes") +
  annotate("text", x = (24 + 48)/2, y = max(semanal$media) * 1.07, label = "Martes") +
  annotate("text", x = (48 + 72)/2, y = max(semanal$media) * 1.07, label = "Miércoles") +
  annotate("text", x = (72 + 96)/2, y = max(semanal$media) * 1.07, label = "Jueves") +
  annotate("text", x = (96 + 120)/2, y = max(semanal$media) * 1.07, label = "Viernes") +
  annotate("text", x = (120 + 144)/2, y = max(semanal$media) * 1.07, label = "Sábado") +
  annotate("text", x = (144 + 168)/2, y = max(semanal$media) * 1.07, label = "Domingo") +
  ggtitle("Concentraciones horarias durante la semana")+
  theme(panel.border = element_rect(color = "black", fill = NA))

 #Se pasan los datos a formato ancho
semanal_wide <- semanal %>%
  pivot_wider(names_from = c(dia_semana,hora), values_from = media)  %>%
  mutate_at(vars(-Estación), as.numeric)
semanal_wide<-as.data.frame(semanal_wide[,-1])
rownames(semanal_wide)<-sort(colnames(datos[,-36]))
```

Clustering jerárquico método Ward con distancia DTW para el comportamiento semanal

```
#Se realiza clustering jerárquico método Ward
semanal_ward<-tsclust(semanal_wide,type="hierarchical",k=4L,preproc = zscore,distance
= "dtw_basic",control=hierarchical_control(method="ward.D"),seed=745L)
fviz_dend(semanal_ward,k=4, lwd=0.78,cex = 0.9,rect = TRUE,k_colors = c("black","blac
k","black","black","black","black"),horiz=TRUE,rect_border = "Dark2", rect_fill = TRU
E,lower_rect = -100)
```

```
#Se presentan los resultados en un gráfico
semanal_nor<-zscore(semanal_wide)
semanalClust_wide <- data.frame(semanal_nor) %>%
  mutate(cluster = semanal_ward@cluster)

semanalClust_long <- semanalClust_wide %>%
  tibble:: rownames_to_column(var="Estación")%>%
  pivot_longer(cols=c(-Estación, -cluster), names_to = "dia_hora", values_to = "valor
")

semanalClust_long<-as.data.frame(semanalClust_long)
semanalClust_long$dia_hora <- as.numeric(factor(semanalClust_long$dia_hora, levels =
unique(semanalClust_long$dia_hora), ordered = TRUE))
semanalClust_long$cluster<-as.numeric(factor(semanalClust_long$cluster))



ggplot()+
  geom_line(data = semanalClust_long, aes(y =valor, x = dia_hora, group = Estación,co
lor=as.factor(cluster)),alpha=3/5)+
  geom_vline(xintercept = c(168), color = "black")+
  geom_vline(xintercept = c(24, 48, 72, 96, 120,144), color = "black",size=1, linetyp
e = "dotted")+
  theme_classic()+
  scale_color_manual(values= c("#d95f02","#e7289a","#1b9e77","#7570b3"))+
  labs(x = "",y="")+
  geom_hline(yintercept = max(semanalClust_long$valor) , color = "black")+
  annotate("text", x = (0 + 24)/2, y = max(semanalClust_long$valor) * 1.1, label = "L
") +
  annotate("text", x = (24 + 48)/2, y = max(semanalClust_long$valor) * 1.1, label = "
M") +
  annotate("text", x = (48 + 72)/2, y = max(semanalClust_long$valor) * 1.1, label = "
X") +
  annotate("text", x = (72 + 96)/2, y = max(semanalClust_long$valor) * 1.1, label = "
J") +
  annotate("text", x = (96 + 120)/2, y = max(semanalClust_long$valor) * 1.1, label =
"V") +
  annotate("text", x = (120 + 144)/2, y = max(semanalClust_long$valor) * 1.1, label =
"S") +
  annotate("text", x = (144 + 168)/2, y = max(semanalClust_long$valor) * 1.1, label =
"D") +
  scale_x_continuous(limits=c(0,168),expand = c(0, 0))+
  theme(panel.border = element_rect(color = "black", fill = NA))+
  facet_wrap(~factor(cluster,levels=c(2,4,1,3),labels=c("1","2","3","4")))+
  guides(color = FALSE)
```

Comportamiento anual

```
#Se calcula el comportamiento anual
anual <- datos_long %>%
  group_by(Estación, mes, dia_mes) %>%
  summarise(media = mean(valor)) %>%
  group_by(Estación)
```

```
#Se crea el grafico del comportamiento anual
anual$fechas<-rep(c(seq(1,31),seq(32,59),seq(60,90),seq(91,120),seq(121,151),seq(152,
181),seq(182,212),seq(213,243),seq(244,273),seq(274,304),seq(305,334),seq(335,365)),3
5)
ggplot(anual, aes(x = fechas, y = media, color = Estación)) +
  geom_line(linewidth=1.1) +
  scale_x_continuous(breaks = c(1,30,58,89,119,150,180,211,242,270,303,333), labels =
c("en","feb","mar","abr","may","jun","jul","agto","sept","oct","nov","dic"), expand =
c(0, 0)) +
  labs(x = "Mes", y = "NOx (µg/m3)") +
  theme_classic()+
  theme(axis.text.x = element_text(size = 10))+
  theme(legend.position = "bottom")+
  scale_color_manual(values=colores)+
  ggtitle("Concentraciones diarias durante el año")

#Se pasan los datos a formato ancho
anual_wide <- anual[,-5] %>%
  pivot_wider(names_from = c(mes,dia_mes), values_from = media)  %>%
  mutate_at(vars(-Estación), as.numeric)
anual_wide<-as.data.frame(anual_wide[,-1])
rownames(anual_wide)<-sort(colnames(datos[,-36]))
```

Mapa de Galicia

```
#Se cargan los datos para representar el mapa
galicia <- esp_get_munic_siane(region = "Galicia") %>%
  mutate(Provincia = esp_dict_translate(ine.prov.name, "es"))

#Se asigna la zonificación al mapa de Galicia
galicia$zonas<-rep(1,length(galicia$codauto))
#Zona Ferrol
galicia$zonas[which(galicia$name == "Ferrol")] <- 2
galicia$zonas[which(galicia$name == "Narón")] <- 2
#Zona A Coruna + Area Metropolitana
galicia$zonas[which(galicia$name == "A Coruña")] <- 3
galicia$zonas[which(galicia$name == "Oleiros")] <- 3
galicia$zonas[which(galicia$name == "Culleredo")] <- 3
galicia$zonas[which(galicia$name == "Cambre")] <- 3
#Zona Santiago + Area Metropolitana
galicia$zonas[which(galicia$name == "Ames")] <- 4
galicia$zonas[which(galicia$name == "Santiago de Compostela")] <- 4
#Zona Pontevedra
galicia$zonas[which(galicia$name == "Pontevedra")] <- 5
#Zona Vigo + Area Metropolitana
galicia$zonas[which(galicia$name == "Vigo")] <- 6
galicia$zonas[which(galicia$name == "Redondela")] <- 6
galicia$zonas[which(galicia$name == "Baiona")] <- 6
galicia$zonas[which(galicia$name == "Nigrán")] <- 6
galicia$zonas[which(galicia$name == "Gondomar")] <- 6
galicia$zonas[which(galicia$name == "O Porriño")] <- 6
galicia$zonas[which(galicia$name == "Mos")] <- 6
#Zona Lugo
galicia$zonas[which(galicia$name == "Lugo")] <- 7
#Zona Ourense
galicia$zonas[which(galicia$name == "Ourense")] <- 8

#Coordenadas de las estaciones a representar
coordenadas<- read_delim("ruta.csv", delim = ";", escape_double = FALSE, trim_ws = TR
UE)
```

Clustering jerárquico método Ward distancia con euclidiana para el comportamiento anual

```
#Se realiza clustering jerárquico método Ward
anual_ward<-tsclust(anual_wide,type="hierarchical",k=2L,preproc = zscore,distance = "
euclidean",control=hierarchical_control(method="ward.D"),seed=745L)
fviz_dend(anual_ward,k=2, lwd=0.78,cex = 0.9,rect = TRUE,k_colors = c("black","black"
,"black","black","black","black"),rect_border = "Dark2", rect_fill = TRUE,lower_rect
= -3,horiz=TRUE)

## Warning in get_col(col, k): Length of color vector was longer than the number
## of clusters - first k elements are used
```

```
#Se presentan los resultados en el mapa
ggplot(galicia) +
  geom_sf(aes(fill = as.factor(zonas)),alpha=2/5) +
  geom_point(data = coordenadas,
             aes(x = Longitud, y = Latitud ,color=as.factor(anual_ward@cluster)),size
=1.3,shape=1,stroke=2)+
  scale_fill_manual(values = c("lavender","darkorchid1", "greenyellow","blue","deeppi
nk","red","springgreen3","dodgerblue","brown"),labels=c("Galicia Rural","Ferrol + Áre
a Metropolitana","A Coruña + Área Metropolitana","Santiago + Área Metropolitana","Pon
tevedra","Vigo + Área Metropolitana","Lugo","Ourense","Oural"))+
  scale_color_manual(values = c("#534ca6","#23b086"),labels=c("Estaciones clúster 1",
"Estaciones clúster 2"))+
  theme_classic()+
  labs(color = "Cluster",
       fill = "Zona") +
  guides(color = guide_legend(title = "Clúster"),
         fill = guide_legend(title = "Zona"))
```

Método del codo para determinar el número óptimo de clústeres para algoritmo k-means

```
#Comportamiento semanal
g1<-fviz_nbclust(semanal_wide, kmeans, method="wss")+
  geom_vline(xintercept = 3, linetype = 2)+
  ggtitle("Semanal")+
  labs(y = "TWSS",x="Número de clústeres")

#Comportamiento anual
g2<-fviz_nbclust(anual_wide, kmeans, method="wss")+
  geom_vline(xintercept = 3, linetype = 2)+
  ggtitle("Anual")+
  labs(y = "TWSS",x="Número de clústeres")

#Se representan
grid.arrange(g1,g2,ncol=2)
```

K-means con distancia euclidiana para el comportamiento semanal

```
#Se realiza k-means con distancia euclideana
clusters_semanal<-tsclust(semanal_wide,type = "partitional", k = 3,distance="euclidea
n",centroid ="mean",seed=745L)

#Se presentan los resultados en un gráfico
semanalClust_wide <- semanal_wide %>%
```

```
  mutate(cluster = clusters_semanal@cluster)

semanalClust_long <- semanalClust_wide %>%
  tibble:: rownames_to_column(var="Estación")%>%
  pivot_longer(cols=c(-Estación, -cluster), names_to = "dia_hora", values_to = "valor
")

centers<-matrix(nrow=length(clusters_semanal@centroids),ncol=(ncol(semanalClust_wide)
-1))
for (i in 1:nrow(centers)){
  centers[i,]<-as.vector(clusters_semanal@centroids[[i]])
}
centers<-data.frame(centers)

centers$centroides<-c(1,2,3)
centers_long <- centers %>%
  pivot_longer(cols=-centroides, names_to = "dia_hora", values_to = "valor")
centers_long<-as.data.frame(centers_long)
centers_long$dia_hora <- as.numeric(factor(centers_long$dia_hora, levels = unique(cen
ters_long$dia_hora), ordered = TRUE))

semanalClust_long<-as.data.frame(semanalClust_long)
semanalClust_long$dia_hora <- as.numeric(factor(semanalClust_long$dia_hora, levels =
unique(semanalClust_long$dia_hora), ordered = TRUE))
semanalClust_long$cluster<-as.numeric(factor(semanalClust_long$cluster))

gg1<-ggplot() +
  geom_line(data = semanalClust_long, aes(y =valor, x = dia_hora, group = Estación, c
olor= as.factor(cluster)))+
  scale_color_manual(values = c("#00CCCC","#FF6633","#CC33CC","#015757","#CC3300","#7
d209c"),
                     labels = c("Cluster 1", "Cluster 2", "Cluster 3", "Centroide 1",
"Centroide 2", "Centroide 3"),
                     name = "Grupo") +
  geom_line(data=centers_long,aes(y=valor,x=dia_hora,group=centroides,color= ifelse(c
entroides==1,"a",ifelse(centroides==2,"b","c"))),size=1.5,linetype="longdash")+
  scale_x_continuous(breaks = seq(0,( 7 * 24-1), 1), labels = dias_horas, expand = c(
0, 0)) +
  geom_vline(xintercept = c(0,24, 48, 72, 96, 120, 144), color = "black",size=1, line
type = "dotted")+
  geom_hline(yintercept = max(semanalClust_long$valor)+1 , color = "black")+
  annotate("text", x = (0 + 24)/2, y = max(semanalClust_long$valor) * 1.07, label = "
Lunes") +
  annotate("text", x = (24 + 48)/2, y = max(semanalClust_long$valor) * 1.07, label =
"Martes") +
  annotate("text", x = (48 + 72)/2, y = max(semanalClust_long$valor) * 1.07, label =
"Miércoles") +
  annotate("text", x = (72 + 96)/2, y = max(semanalClust_long$valor) * 1.07, label =
"Jueves") +
  annotate("text", x = (96 + 120)/2, y = max(semanalClust_long$valor) * 1.07, label =
"Viernes") +
  annotate("text", x = (120 + 144)/2, y = max(semanalClust_long$valor) * 1.07, label
= "Sábado") +
  annotate("text", x = (144 + 168)/2, y = max(semanalClust_long$valor) * 1.07, label
= "Domingo") +
  labs(x = "Hora", y = "NOx (µg/m3)",color="")+
  theme_classic()+
  ggtitle("Concentraciones horarias durante la semana")+
  theme(panel.border = element_rect(color = "black", fill = NA))
```

K-means con distancia euclidiana para el comportamiento anual

```
#Se aplica k-means el el comportamiento anual
clusters_anual<-tsclust(anual_wide,type = "partitional", k = 3,distance="euclidean",c
entroid ="mean",seed=745L)
```

```
#Se presentan los resultados en un gráfico
anualClust_wide <- anual_wide %>%
  mutate(cluster = clusters_anual@cluster)

anualClust_long <- anualClust_wide %>%
  tibble:: rownames_to_column(var="Estación")%>%
  pivot_longer(cols=c(-Estación, -cluster), names_to = "dia", values_to = "valor")

centers<-matrix(nrow=length(clusters_anual@centroids),ncol=(ncol(anualClust_wide)-1))
for (i in 1:nrow(centers)){
  centers[i,]<-as.vector(clusters_anual@centroids[[i]])
}
centers<-data.frame(centers)
centers$centroides<-c(1,2,3)
centers_long <- centers %>%
  pivot_longer(cols=-centroides, names_to = "dia", values_to = "valor")
centers_long<-as.data.frame(centers_long)
centers_long$dia <- as.numeric(factor(centers_long$dia, levels = unique(centers_long$
dia), ordered = TRUE))

anualClust_long<-as.data.frame(anualClust_long)
anualClust_long$dia <- as.numeric(factor(anualClust_long$dia, levels = unique(anualCl
ust_long$dia), ordered = TRUE))
anualClust_long$cluster<-as.numeric(factor(anualClust_long$cluster))

gg2<-ggplot() +
  geom_line(data = anualClust_long, aes(y =valor, x = dia, group = Estación, color= a
s.factor(cluster)))+
  scale_color_manual(values = c("#FF6633","#00CCCC","#CC33CC","#CC3300","#015757","#7
d209c"),
                     labels = c("Cluster 1", "Cluster 2", "Cluster 3", "Centroide 1",
"Centroide 2", "Centroide 3"),
                     name = "Grupo") +
  geom_line(data=centers_long,aes(y=valor,x=dia,group=centroides,color= ifelse(centro
ides==1,"a",ifelse(centroides==2,"b","c"))),size=1.2,linetype="longdash")+
  scale_x_continuous(breaks = c(1,30,58,89,119,150,180,211,242,270,303,333), labels =
c("en","feb","mar","abr","may","jun","jul","agto","sept","oct","nov","dic"), expand =
c(0, 0)) +
  labs(x = "Mes", y = "NOx (µg/m3)",color="")+
  theme_bw()+
  ggtitle("Concentraciones diarias durante el año")
```

PAM con distancia euclidiana para el comportamiento semanal

```
#Se aplica PAM el el comportamiento semanal
clusters_semanalPAM<-tsclust(semanal_wide,type = "partitional", k = 3,distance="eucli
dean",centroid ="pam",seed=745)

#Se presentan los resultados en un gráfico
semanalClust_wide <- semanal_wide %>%
  mutate(cluster = clusters_semanalPAM@cluster)

semanalClust_long <- semanalClust_wide %>%
  tibble:: rownames_to_column(var="Estación")%>%
  pivot_longer(cols=c(-Estación, -cluster), names_to = "dia_hora", values_to = "valor
")

centers<-matrix(nrow=length(clusters_semanalPAM@centroids),ncol=(ncol(semanalClust_wi
de)-1))
for (i in 1:nrow(centers)){
  centers[i,]<-as.vector(clusters_semanalPAM@centroids[[i]])
}
centers<-data.frame(centers)
```

```
centers$centroides<-c(1,2,3)
centers_long <- centers %>%
  pivot_longer(cols=-centroides, names_to = "dia_hora", values_to = "valor")
centers_long<-as.data.frame(centers_long)
centers_long$dia_hora <- as.numeric(factor(centers_long$dia_hora, levels = unique(cen
ters_long$dia_hora), ordered = TRUE))

semanalClust_long<-as.data.frame(semanalClust_long)
semanalClust_long$dia_hora <- as.numeric(factor(semanalClust_long$dia_hora, levels =
unique(semanalClust_long$dia_hora), ordered = TRUE))
semanalClust_long$cluster<-as.numeric(factor(semanalClust_long$cluster))

gg1PAM<-ggplot() +
  geom_line(data = semanalClust_long, aes(y =valor, x = dia_hora, group = Estación, c
olor= as.factor(cluster)))+
  scale_color_manual(values = c("#00CCCC","#FF6633","#CC33CC","#015757","#CC3300","#7
d209c"),
                     labels = c("Cluster 1", "Cluster 2", "Cluster 3", "Centroide 1",
"Centroide 2", "Centroide 3"),
                     name = "Grupo") +
  geom_line(data=centers_long,aes(y=valor,x=dia_hora,group=centroides,color= ifelse(c
entroides==1,"a",ifelse(centroides==2,"b","c"))),size=1.5,linetype="longdash")+
  scale_x_continuous(breaks = seq(0,( 7 * 24-1), 1), labels = dias_horas, expand = c(
0, 0)) +
  geom_vline(xintercept = c(0,24, 48, 72, 96, 120, 144), color = "black",size=1, line
type = "dotted")+
  labs(x = "Hora", y = "NOx (µg/m3)",color="")+
  theme_classic()+
  geom_hline(yintercept = max(semanalClust_long$valor)+1 , color = "black")+
  annotate("text", x = (0 + 24)/2, y = max(semanalClust_long$valor) * 1.07, label = "
Lunes") +
  annotate("text", x = (24 + 48)/2, y = max(semanalClust_long$valor) * 1.07, label =
"Martes") +
  annotate("text", x = (48 + 72)/2, y = max(semanalClust_long$valor) * 1.07, label =
"Miércoles") +
  annotate("text", x = (72 + 96)/2, y = max(semanalClust_long$valor) * 1.07, label =
"Jueves") +
  annotate("text", x = (96 + 120)/2, y = max(semanalClust_long$valor) * 1.07, label =
"Viernes") +
  annotate("text", x = (120 + 144)/2, y = max(semanalClust_long$valor) * 1.07, label
= "Sábado") +
  annotate("text", x = (144 + 168)/2, y = max(semanalClust_long$valor) * 1.07, label
= "Domingo") +
  ggtitle("Concentraciones horarias durante la semana")+
  theme(panel.border = element_rect(color = "black", fill = NA))
```

PAM con distancia euclidiana para el comportamiento anual

```
#Se aplica PAM el el comportamiento anual
clusters_semanalPAM<-tsclust(anual_wide,type = "partitional", k = 3,distance="euclide
an",centroid ="pam",seed=745L)

#Se presentan los resultados en un gráfico
anualClust_wide <- anual_wide %>%
  mutate(cluster = clusters_semanalPAM@cluster)

anualClust_long <- anualClust_wide %>%
  tibble:: rownames_to_column(var="Estación")%>%
  pivot_longer(cols=c(-Estación, -cluster), names_to = "dia", values_to = "valor")

centers<-matrix(nrow=length(clusters_semanalPAM@centroids),ncol=(ncol(anualClust_wide
)-1))
for (i in 1:nrow(centers)){
```

```
  centers[i,]<-as.vector(clusters_semanalPAM@centroids[[i]])
}
centers<-data.frame(centers)
centers$centroides<-c(1,2,3)
centers_long <- centers %>%
  pivot_longer(cols=-centroides, names_to = "dia", values_to = "valor")
centers_long<-as.data.frame(centers_long)
centers_long$dia <- as.numeric(factor(centers_long$dia, levels = unique(centers_long$
dia), ordered = TRUE))

anualClust_long<-as.data.frame(anualClust_long)
anualClust_long$dia <- as.numeric(factor(anualClust_long$dia, levels = unique(anualCl
ust_long$dia), ordered = TRUE))
anualClust_long$cluster<-as.numeric(factor(anualClust_long$cluster))

gg2PAM<-ggplot() +
  geom_line(data = anualClust_long, aes(y =valor, x = dia, group = Estación, color= a
s.factor(cluster)))+
  scale_color_manual(values = c("#FF6633","#00CCCC","#CC33CC","#CC3300","#015757","#7
d209c"),
                     labels = c("Cluster 1", "Cluster 2", "Cluster 3", "Centroide 1",
"Centroide 2", "Centroide 3"),
                     name = "Grupo") +
  geom_line(data=centers_long,aes(y=valor,x=dia,group=centroides,color= ifelse(centro
ides==1,"a",ifelse(centroides==2,"b","c"))),size=1.2,linetype="longdash")+
  scale_x_continuous(breaks = c(1,30,58,89,119,150,180,211,242,270,303,333), labels =
c("en","feb","mar","abr","may","jun","jul","agto","sept","oct","nov","dic"), expand =
c(0, 0)) +
  labs(x = "Mes", y = "NOx (µg/m3)",color="")+
  theme_bw()+
  ggtitle("Concentraciones diarias durante el año")
```

Representación resultados k-means y PAM en un gráfico y en el mapa

```
grid.arrange(gg1,gg1PAM)

grid.arrange(gg2,gg2PAM)

#Se asigna cada estación a su respectivo clúster
coordenadas$diario<-factor(clusters_semanal@cluster)

ggplot(galicia) +
  geom_sf(aes(fill = as.factor(zonas)),alpha=2/5) +
  geom_point(data = coordenadas,
             aes(x = Longitud, y = Latitud ,color=diario),size=1.3,shape=1,stroke=2)+
  scale_fill_manual(values = c("lavender","darkorchid1", "greenyellow","blue","deeppi
nk","red","springgreen3","dodgerblue"),labels=c("Galicia Rural","Ferrol + Área Metrop
olitana","A Coruña + Área Metropolitana","Santiago + Área Metropolitana","Pontevedra"
,"Vigo + Área Metropolitana","Lugo","Ourense"))+
  scale_color_manual(values = c("#00a3a3","#FF6633","#AF00B0"),labels=c("Estaciones c
lúster 1","Estaciones clúster 2", "Estaciones clúster 3"))+
  theme_classic()+
  labs(fill = "Zona",color = "Clúster") +
  guides(fill = guide_legend(title = "Zona"),color = guide_legend(title = "Clúster"))

#Modificacion de zonificacion propuesta
galicia2<-galicia
galicia2$zonas[which(galicia2$name == "Pontеareas")] <- 6
galicia2$zonas[which(galicia2$name == "Arteixo")] <- 10
galicia2$zonas[which(galicia2$name == "O Incio")] <- 9
galicia2$zonas[which(galicia2$name == "Sarria")] <- 9

#Se representa en el mapa de la nueva zonificacion
ggplot(galicia2) +
  geom_sf(aes(fill = as.factor(zonas)),alpha=2/5) +
```

```
  geom_point(data = coordenadas,aes(x = Longitud, y = Latitud ,color=diario),size=1.3
,shape=1,stroke=2)+
  scale_color_manual(values = c("#00a3a3","#FF6633","#AF00B0"),labels=c("Estaciones c
lúster 1","Estaciones clúster 2", "Estaciones clúster 3"))+
  scale_fill_manual(values = c("lavender","darkorchid1", "greenyellow","blue","deeppi
nk","red","springgreen3","dodgerblue","brown","magenta"),labels=c("Galicia Rural","Fe
rrol + Área Metropolitana","A Coruña + Área Metropolitana","Santiago + Área Metropoli
tana","Pontevedra","Vigo + Área Metropolitana","Lugo","Ourense","Oural","Arteixo"))+
  theme_classic()+
  labs(color = "Cluster",
       fill = "Zona") +
  guides(color = guide_legend(title = "Clúster"),
         fill = guide_legend(title = "Zona"))
```

Anexo B. Código utilizado para realizar la figura 3.1.

```
#Se cargan las librerías necesarias
library(TSrepr)
library(ggplot2)
library(ggthemes)

#Se crea la serie original y la serie representada con el método PAA
serie_original <- rnorm(100)
tamano_fragmento <- 10
serie_paa <- repr_paa(serie_original, tamano_fragmento,func=meanC)

df <- data.frame(tiempo = 1:100,
                 serie_original = serie_original,
                 serie_paa = rep(serie_paa, each = tamano_fragmento))
df$fragmento <- seq(1,100,10)

#Representacion
ggplot(df, aes(x = tiempo, y = serie_original)) +
  geom_line(aes(col = "Original"),linewidth=1.5,alpha=4/5) +
  geom_step(aes(x = tiempo, y = serie_paa,col="PAA"),linewidth=1.3) +
  geom_vline(aes(xintercept = fragmento), linetype = "dotted", size = 0.5,color="navy
")+  geom_vline(aes(xintercept = 100), linetype = "dotted", size = 0.5,color="navy")+
  theme(panel.grid = element_blank())+
  labs(x = "", y = "", title = "",legend="") +
  scale_color_manual(values=c("Original"="mediumseagreen","PAA"="indianred3"),label=)
+  theme_tufte()
```

Anexo C. Código utilizado para realizar la figura 3.2.

```
#Se cargan las librerías necesarias
library(TSrepr)
library(ggplot2)
library(ggthemes)
library(jmotif)

#Se crea la serie original y la serie representada con el método PAA
serie_original <- rnorm(100)
tamano_fragmento <- 10
serie_paa <- repr_paa(serie_original, tamano_fragmento,func=meanC)

df <- data.frame(tiempo = 1:100,
                 serie_original = serie_original,
                 serie_paa = rep(serie_paa, each = tamano_fragmento))
df$fragmento <- seq(1,100,10)
```

```
#SAX
b<-series_to_string(serie_paa,7)
b<-strsplit(b, "")[[1]]
a<-alphabet_to_cuts(7)

ggplot(df, aes(x = tiempo, y = serie_original)) +
  geom_line(aes(col = "Original"),linewidth=1.5,alpha=4/5) +
  geom_step(aes(x = tiempo, y = serie_paa,col="PAA"),linewidth=1.3) +
  geom_vline(aes(xintercept = fragmento), linetype = "dotted", size = 0.5,color="navy
")+
  geom_vline(aes(xintercept = 100), linetype = "dotted", size = 0.5,color="navy")+
  geom_hline(aes(yintercept=a[2]), lty=2, lwd=1,col="magenta3",alpha=4/5)+
  geom_hline(aes(yintercept=a[3]), lty=2, lwd=1, col="magenta3",alpha=4/5)+
  geom_hline(aes(yintercept=a[4]), lty=2, lwd=1, col="magenta3",alpha=4/5)+
  geom_hline(aes(yintercept=a[5]), lty=2, lwd=1, col="magenta3",alpha=4/5)+
  geom_hline(aes(yintercept=a[6]), lty=2, lwd=1, col="magenta3",alpha=4/5)+
  geom_hline(aes(yintercept=a[7]), lty=2, lwd=1,col="magenta3",alpha=4/5)+
  theme(panel.grid = element_blank())+
  labs(x = "", y = "", title = "",legend="") +
  annotate("text", x =-5, y = -1.3, label = "a",color="magenta3")+
  annotate("text", x =-5, y = (a[2]+a[3])/2, label = "b",color="magenta3")+
  annotate("text", x =-5, y = (a[3]+a[4])/2, label = "c",color="magenta3")+
  annotate("text", x =-5, y = (a[4]+a[5])/2, label = "d",color="magenta3")+
  annotate("text", x =-5, y = (a[5]+a[6])/2, label = "e",color="magenta3")+
  annotate("text", x =-5, y = (a[6]+a[7])/2, label = "f",color="magenta3")+
  annotate("text", x =-5, y = 1.3, label = "g",color="magenta3")+
  annotate("text", x =( df$fragmento[1]+df$fragmento[2])/2, y = serie_paa[1]+0.2, lab
el = b[1],size=5)+
  annotate("text", x =( df$fragmento[2]+df$fragmento[3])/2, y = serie_paa[2]+0.2, lab
el = b[2],size=5)+
  annotate("text", x =( df$fragmento[3]+df$fragmento[4])/2, y = serie_paa[3]+0.2, lab
el = b[3],size=5)+
  annotate("text", x =( df$fragmento[4]+df$fragmento[5])/2, y = serie_paa[4]+0.2, lab
el = b[4],size=5)+
  annotate("text", x =( df$fragmento[5]+df$fragmento[6])/2, y = serie_paa[5]+0.2, lab
el = b[5],size=5)+
  annotate("text", x =( df$fragmento[6]+df$fragmento[7])/2, y = serie_paa[6]+0.2, lab
el = b[6],size=5)+
  annotate("text", x =( df$fragmento[7]+df$fragmento[8])/2, y = serie_paa[7]+0.2, lab
el = b[7],size=5)+
  annotate("text", x =( df$fragmento[8]+df$fragmento[9])/2, y = serie_paa[8]+0.2, lab
el = b[8],size=5)+
  annotate("text", x =( df$fragmento[9]+df$fragmento[10])/2, y = serie_paa[9]+0.2, la
bel = b[9],size=5)+
  annotate("text", x =( df$fragmento[10])+4, y = 0.20+0.2, label = b[10],size=5)+
  scale_color_manual(values=c("Original"="mediumseagreen","PAA"="indianred3"),label=)
+  theme_tufte()
```

Anexo D. Código utilizado para realizar la figura 3.3.

```
#Se cargan las librerías necesarias
library(dtw)
library(ggplot2)
library(gridExtra)
library(dplyr)
library(tidyr)

#Se crean las dos series de tiempo
longitud <- 100
serie1<- sin (seq(0, 5*pi, length.out = longitud))
serie2 <- cos(seq(3, (5*pi+3), length.out = longitud))+2

#Distancia euclidiana
eu_df <- data.frame(
  x1 = rep(seq(1, longitud), 2), y1 = c(serie1, serie2),y2 = c(serie2, serie1),
```

```
  serie1,serie2)

eu<-ggplot(eu_df, aes(x = x1, y = y1)) +
  geom_line(aes(y = serie1),color = "#FF9933",linewidth=1.5) +
  geom_line(aes(y = serie2), color = "#9900CC",linewidth=1.5) +
  geom_point(aes(x=x1,y=serie1),color="black",size=0.9)+
  geom_point(aes(x=x1,y=serie2),color="black",size=0.9)+
  geom_segment(aes(x = x1, y = y1, xend = x1, yend = y2))+
  theme(axis.line = element_blank(),
        axis.text = element_blank(),
        axis.title = element_blank(),
        panel.grid = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank(),axis.ticks = element_blank())+
  ggtitle("Distancia euclidiana") +
  theme(plot.title = element_text(face = "bold", hjust = 0.5))

#Distancia DTW
dtw_alineacion <- dtw(serie1, serie2, keep = TRUE, step.pattern = rabinerJuangStepPat
tern(6))

dtw_df <- data.frame(
  x1 = seq(1, longitud)[dtw_alineacion$index1],
  y1 = serie1[dtw_alineacion$index1],
  x2 = seq(1, longitud)[dtw_alineacion$index2],
  y2 = serie2[dtw_alineacion$index2]
)

ts_df <- data.frame(x = 1:longitud, y1 = serie1, y2 = serie2)
ts_df <- ts_df %>% gather(key = "series", value = "value", -x) %>%
  mutate( serie = ifelse(series == "y1", "y1", "y2"))
dt<-ggplot() +
  geom_segment(data = dtw_df, aes(x = x1, y = y1, xend = x2, yend = y2)) +
  geom_line(data = ts_df, aes(x = x, y = value, color = serie),linewidth=1.5,show.leg
end = FALSE) +
  geom_point(data = dtw_df, aes(x = x1, y = y1),color="black",size=0.9) +
  geom_point(data = dtw_df, aes(x = x2, y = y2),color="black",size=0.9) +
  scale_color_manual(values = c("y1" = "#FF9933", "y2" = "#9900CC"))+
  theme(axis.line = element_blank(),
        axis.text = element_blank(),
        axis.title = element_blank(),
        panel.grid = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank(),axis.ticks = element_blank())+
  ggtitle("Distorsión dinámica de tiempo (DTW) ") +
  theme(plot.title = element_text(face = "bold", hjust = 0.5))

#Representación
grid.arrange(eu,dt)

#Representación de la ruta óptima de alineación
dtwPlotThreeWay(dtw_alineacion,match.indices = 1000,type.ts = "l")
```