VNiVERSiDAD D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

Universidad de Salamanca
Facultad de Ciencias
Grado en Matemáticas

# Técnicas estadísticas multivariantes y su utilidad en el análisis de datos funcionales

Trabajo de fin de grado

Autora:
**Ana Mª Rodríguez Arcos**

Tutoras académicas
**Ana Belén Nieto Librero**
**Nerea González García**

Salamanca, 7 de septiembre de 2021

VNiVERSiDAD D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

# Técnicas estadísticas multivariantes y su utilidad en el análisis de datos funcionales

Trabajo de fin de grado


Autora:
**Ana Mª Rodríguez Arcos**

Tutoras:
**Ana Belén Nieto Librero**
**Nerea González García**


Firma de la autora:

Firmado por RODRIGUEZ ARCOS ANA MARIA - ***3356** el día 06/09/2021 con un certificado emitido por AC FNMT Usuarios

Fdo: Ana Mª Rodríguez Arcos

Firma de las tutoras:

NIETO LIBRERO ANA BELEN - 11966438K Firmado digitalmente por NIETO LIBRERO ANA BELEN - 11966438K Fecha: 2021.09.06 19:17:13 +02'00'

GONZALEZ GARCIA NEREA - 72828390V Firmado digitalmente por GONZALEZ GARCIA NEREA - 72828390V Fecha: 2021.09.06 19:17:35 +02'00'

Fdo: Ana Belén Nieto Librero y Nerea González García

Salamanca, 7 de septiembre de 2021

# Índice general

# Introducción

En la última década se está viviendo una metamorfosis ligada al avanzado desarrollo computacional. Algunos economistas, como Klaus Schwab [19], ya la categorizan como la Cuarta Revolución Industrial:

"Estamos al borde de una revolución tecnológica que modificará fundamentalmente la forma en que vivimos, trabajamos y nos relacionamos. En su escala, alcance y complejidad, la transformación será distinta a cualquier cosa que el género humano haya experimentado antes".

Sin duda es un momento en el que el poder se mide en datos, y saber interpretarlos y analizarlos es fundamental. Como consecuencia inmediata se puede señalar el auge de las técnicas estadísticas multivariantes y su utilidad. Términos como *Big Data* y *Data Mining*, se han convertido en habituales; *Big Data* hace referencia al manejo de grandes volúmenes de datos y el concepto de *Data Mining* se utiliza para denominar al proceso de búsqueda de dependencias, tendencias y patrones latentes en los datos analizados.

La disponibilidad de tal cantidad de información, lejos de ser ventajoso, sin conocimiento previo de cómo manipularla puede ser perjudicial para el estudio. En el proceso exploratorio de los datos se debe prestar especial atención a discriminar, por ejemplo, los datos redundantes y anómalos, tratar de reducir las dimensiones altas que impiden una representación clara o identificar la existencia de patrones que permitan simplicar el estudio y den respuesta a las hipótesis iniciales.

El análisis de datos multivariantes recoge todas las metodologías encargadas de estudiar un conjunto de variables observadas simultáneamente sobre un conjunto de individuos. Tanto Cuadras [3], como Peña [14] aportan a la literatura una revisión muy completa en la que se podrán indagar en aquellas técnicas no tratadas en el trabajo. En el texto se abordarán algunas de las principales técnicas estadísticas como el Análisis de Regresión Lineal, la Descomposión en Valores Singulares, el Análisis de Componentes Principales y el Análisis de Cluster.

La labor de estas técnicas estadísticas multivariantes en todas las tareas mencionadas

anteriormente es crucial, sin embargo, limita el análisis a un plano en el que solo se dispone de una muestra de datos discretos. Los dos grandes incovenientes del análisis de datos discretos se encuentran en garantizar la homogeneidad de las mediciones y conseguir almacenar el máximo número de observaciones de forma eficiente. Son precisamente los retos principales a los que se enfrenta el análisis de datos funcionales en esta rama del conocimiento. El análisis de datos funcionales contempla la recuperación de la naturaleza continua de los datos, convirtiendo una muestra discreta en una función mediante aproximaciones por bases de funciones. Fueron Ramsay [17], Ramsay y Dalzell [8] los que introdujeron el término de dato funcional; pero no fue hasta la publicación de [18] por Ramsay y Silverman, cuando creó revuelo entre los científicos. Por tanto, con la aproximación por bases de funciones se consigue en primer lugar homogeneizar los datos al estar representados mediante una base de manera universal y por otro lado al convertir los datos en funciones se logra conocer la información entre observaciones sin tener la necesidad de almacenarla. En este trabajo se verá cómo sobre estos datos se aplican las técnicas estadísticas del análisis multivariante (modificadas para adaptarse al nuevo marco de datos).

El trabajo tiene como objetivo principal revisar la bibliografía de las técnicas estadísticas multivariantes, para luego definir la relación y su utilidad en el análisis de datos funcionales. Como objetivo específico se desea demostrar la habilidad de dichas técnicas en el análisis de un conjunto de datos reales mediante el uso del software de programación Python.

El cuerpo del trabajo está organizado de la siguiente manera:

- El capítulo 1 está dedicado a conocer algunas de las principales técnicas estadísticas multivariantes clásicas para crear un contexto y asentar los principios de las metodologías antes de generalizar al caso funcional. Se exponen las técnicas de Regresión, centrándose en el modelo lineal; se continúa con la Descomposición en Valores Singulares (SVD, por sus siglas en inglés). Seguidamente se presenta el Análisis de Componentes Principales (PCA, por sus siglas en inglés) y por último se trata el Análisis de Cluster.

- En el capítulo 2 se plantean los fundamentos matemáticos del Análisis de Datos Funcionales y prepara al lector para entender la teoría subyacente de las adaptaciones de las técnicas al ámbito funcional recopilando algunos de los conceptos básicos necesarios. Muestra el proceso faseado de la representación de los datos en función de una base, elección de la base acorde al comportamiento de los datos y posterior ajuste y cálculo de coeficientes.

- En el capítulo 3 se definen los estadísticos descriptivos adaptados al marco funcional y después se definen cada una de las técnicas recogidas en el primer capítulo desde un espacio de trabajo de dimensión infinita. Se verá cómo el modelo de

Regresión se puede convertir en un modelo funcional desde varias perspectivas. Seguidamente se presenta el Análisis de Componentes Principales Funcional (FPCA, por sus siglas en inglés) y para el Análisis Funcional de Cluster se exponen distintas posibilidades de clasificación automática según el tratamiento de datos realizado.

- En el capítulo 4 se demuestra la utilidad de las técnicas expuestas en el análisis de un conjunto de datos reales. Se ha seleccionado una base de datos de la Agencia Estatal de Meteorología (AEMET) que almacena los registros de temperaturas mínimas, medias y máximas a lo largo de un año en veinte estaciones meteorológicas españolas. Se hará uso del FPCA para una mejor comprensión de los datos. Igualmente, se busca demostrar la capacidad clasificatoria del Análisis de Cluster aplicado a datos funcionales.
- Se finaliza con dos anexos complementarios. En el primero, se desgrana el código programado para el análisis del caso práctico utilizado para la conversión de datos a funcional, el FPCA y para el Análisis de Cluster Funcional. Se concluye con un último anexo con las figuras de los ejemplos explicativos del texto teórico

Para la implementación de los ejemplos y del caso práctico se ha recurrido al lenguaje de programación Python. Se ha elegido por ser un lenguaje muy potente en representación de datos y por disponer de una librería extensa y bien documentada para el análisis de datos funcionales. Además de ser un lenguaje eficiente y sencillo, existen diversos entornos de desarrollo que facilitan la programación; en este caso se ha trabajado sobre un cuaderno de Jupyter.

# Capítulo 1

# Técnicas clásicas multivariantes

Disponer de una muestra de observaciones de carácter multivariante para realizar un estudio es muy ventajoso por la cantidad de información que contienen y que puede evaluarse de manera conjunta. Sin duda, utilizando las herramientas adecuadas y un buen tratamiento de los datos, se pueden alcanzar conclusiones con gran valor. Es por ello que en este capítulo se presentan algunas de las técnicas más útiles en este proceso.

Si se desea examinar si existe alguna dependencia oculta en los datos o conocer las relaciones entre varibales para materializar una predicción del comportamiento de alguno en concreto conocidos el resto, es la Regresión Simple Multivariante la que aportará luz a estas inquietudes.

El hecho de contar con tantos datos puede originar información reduntante o que simplemente sea poco manipulable. En este aspecto, las técnicas de reducción de la dimensionalidad como la Descomposición en Valores Singulares o el Análisis de componentes principales pueden ser de gran ayuda. La Descomposición en Valores Singulares es además, un método especialmente valioso por su relevancia en el desarrollo de otras técnicas facilitando los cálculos.

También puede resultar de interés reconocer comportamientos comunes entre individuos o variables, para aislarlos o únicamente aprovechar dicha información. El papel del Análisis de cluster es indiscutible en dicho campo.

## 1.1. Análisis de Regresión Multivariante

En primer lugar se recogen varios conceptos claves en el desarrollo de la sección:

**Definición 1.1.1.** *Se denomina* ***variable dependiente y****, a la variable cuyo comportamiento se le pretende dar explicación. También recibe otros nombres como variable*

*endógena o respuesta.*

**Definición 1.1.2.** *Se conoce como* ***variable independiente*** $\boldsymbol{x}$*, a la variable que determina el comportamiento de la variable dependiente* $\boldsymbol{y}$ *cuyo valor no depende de otra variable. Recibe otros nombres como variable exógena, explicativa o predictora.*

**Definición 1.1.3.** *Dadas n observaciones independientes sobre dos variables* $\boldsymbol{x} = (x_1, ..., x_n)$ *e* $\boldsymbol{y} = (y_1, ..., y_n)$*, donde* $\boldsymbol{x}$ *se cree influyente sobre el comportamiento de* $\boldsymbol{y}$*, se conoce como modelo de* ***regresión lineal simple*** *a la función que relaciona a ambas:*

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad i = \{1, ..., n\} \tag{1.1}$$

*Siendo* $\boldsymbol{y} = (y_1, ..., y_n)$ *la variable dependiente,* $\boldsymbol{x} = (x_1, ..., x_n)$ *la variable independiente,* $\beta_0$ *y* $\beta_1$ *los coeficientes del modelo y* $\boldsymbol{\varepsilon} = (\varepsilon_1, ..., \varepsilon_n)$ *donde* $E(\boldsymbol{\varepsilon}) = 0$ *y* $Cov(\boldsymbol{\varepsilon}) = \sigma^2 \boldsymbol{I}$*, siendo* $Cov(\boldsymbol{\varepsilon})$ *la matriz de covarianzas de* $\boldsymbol{\varepsilon}$*, con* $\boldsymbol{\varepsilon}$ *el término de error con media nula y varianza* $\sigma^2$*, un componente aleatorio que representa toda la información residual para las variables independientes.*

El Análisis de Regresión Simple abarca tanto el estudio de relaciones entre las variables como la explicación y predicción de la variable dependiente, pero ¿qué ocurre si no basta con una variable independiente para describir la variable dependiente?

A diferencia del simple, el modelo del Análisis de Regresión Múltiple contempla varias variables explicativas o independientes, aunque mantiene su enfoque sobre una única variable respuesta o dependiente.

Por último se tratará la generalización al Análisis de Regresión Multivariante, con varias variables respuesta en el modelo. Para simplificar la explicación se ha escogido el caso de la Regresión Lineal.

### 1.1.1. Regresión Lineal Múltiple

Se ha tomado como referencia la literatura de Jonhson y Wichern [10], y de Ramsay y Sliverman [18].

**Definición 1.1.4.** *Dadas* $\{\boldsymbol{x_1}, ..., \boldsymbol{x_p}\}$ *variables independientes, con* $p > 1$*, sea* $\boldsymbol{y} = (y_1, ..., y_n)$ *la única variable dependiente y n observaciones independientes, se conoce*

*como modelo de* ***regresión lineal múltiple*** *a la función que las relaciona:*

$$\begin{pmatrix} y_1 \\ y_2 \\ \\ \vdots \\ \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ & & & \\ \vdots & \vdots & \cdots & \vdots \\ & & & \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \\ \vdots \\ \\ \varepsilon_n \end{pmatrix} \tag{1.2}$$

*Se presenta de forma abreviada como:* $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$*, siendo* $\boldsymbol{y} = (y_1, ..., y_n)$ *la variable dependiente,* $\boldsymbol{X}$ *la matriz que contiene las variables independientes* $\{\boldsymbol{x_1}, ..., \boldsymbol{x_p}\}$*,* $\boldsymbol{\beta} = (\beta_0, ..., \beta_p)$ *los coeficientes del modelo y donde* $\boldsymbol{\varepsilon} = (\varepsilon_1, ..., \varepsilon_n)$ *con* $E(\boldsymbol{\varepsilon}) = 0$*,* $Cov(\boldsymbol{\varepsilon}) = \sigma^2\boldsymbol{I}$ *es el error de la aproximación con media nula y varianza* $\sigma^2$*.*

El Análisis de Regresión Lineal persigue establecer la relación que involucra a las variables observadas y la variable respuesta. Se distinguen principalmente dos objetivos: estudiar cómo afectan los cambios de la variable predictora en la respuesta y construir un modelo de carácter predictivo para la variable respuesta o dependiente. En cualquier caso hay que determinar el valor de los parámetros $\boldsymbol{\beta} = (\beta_0, ..., \beta_p)$ que satisfagan la ecuación. El método más utilizado para llevar a cabo el proceso es la estimación por mínimos cuadrados.

**Definición 1.1.5.** *Se conoce como* ***valor estimado*** $\hat{\boldsymbol{y}} = (\hat{y}_1, ..., \hat{y}_n)$ *de* $\boldsymbol{y}$*, a la aproximación del valor de* $\boldsymbol{y}$ *dada por el modelo, construída a partir de los estimadores* $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, ..., \hat{\beta}_p)$ *de los coeficientes del modelo* $\boldsymbol{\beta} = (\beta_0, ..., \beta_p)$*:*

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} \tag{1.3}$$

**Definición 1.1.6.** *Se denomina* ***error*** *o residuo a la diferencia entre el valor observado de la variable dependiente* $\boldsymbol{y} = (y_1, ..., y_n)$ *y el valor estimado* $\hat{\boldsymbol{y}} = (\hat{y}_1, ..., \hat{y}_n)$ *de* $\boldsymbol{y}$ *por el modelo. Sean* $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, ..., \hat{\beta}_p)$ *los estimadores de los coeficientes del modelo* $\boldsymbol{\beta} = (\beta_0, ..., \beta_p)$*, el residuo queda definido para cada observación i:*

$$\hat{\varepsilon_i} = y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \cdots - \hat{\beta}_p x_{ip} = y_i - \hat{y}_i \tag{1.4}$$

*Donde* $i = \{1, ..., n\}$*.*

**Definición 1.1.7.** *Sea* $\hat{\boldsymbol{y}} = (\hat{y}_1, ..., \hat{y}_n)$ *un vector de n predicciones e* $\boldsymbol{y} = (y_1, ..., y_n)$ *el vector con los valores observados, entonces se define el estimador del* ***error cuadrático medio*** *(MSE, por sus siglas en inglés) como:*

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{1.5}$$

**Definición 1.1.8.** *El **método de mínimos cuadrados** es un método de análisis numérico encargado de estimar los coeficientes de la regresión $\boldsymbol{\beta} = (\beta_0, ..., \beta_p)$ con el fin de conseguir un modelo con un error cuadrático medio mínimo, donde se busca encontrar la solución que minimiza el siguiente problema de optimización:*

$$\min_{\hat{\boldsymbol{\beta}}}(\sum_{i=1}^{n}(y_i - \hat{y}_i)^2) = \min_{\hat{\boldsymbol{\beta}}}(\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}}) = \min_{\hat{\boldsymbol{\beta}}}(\boldsymbol{y'y} - 2\hat{\boldsymbol{\beta}}'\boldsymbol{X'y} + \hat{\boldsymbol{\beta}}'\boldsymbol{X'X}\hat{\boldsymbol{\beta}}) \tag{1.6}$$

*Siendo $\hat{\boldsymbol{\varepsilon}}$ el vector que contiene todos los residuos.*

Tiene como resultado la estimación $\hat{\boldsymbol{\beta}} = (\boldsymbol{X'X})^{-1}\boldsymbol{X'y}$. Así queda completamente determinado el modelo:

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{X}(\boldsymbol{X'X})^{-1}\boldsymbol{X'y} \tag{1.7}$$

$$\hat{\boldsymbol{\varepsilon}} = \boldsymbol{y} - \hat{\boldsymbol{y}} = (\boldsymbol{I} - \boldsymbol{X}(\boldsymbol{X'X})^{-1}\boldsymbol{X'})\boldsymbol{y} \tag{1.8}$$

Para comprobar cómo es el ajuste del modelo a los datos se pueden utilizar distintas métricas como el coeficiente de determinación para calcular la variabilidad de la variable dependiente explicada por las variables independientes.

**Nota 1.1.1.** *Todos los resultados obtenidos para el Análisis de Regresión Lineal Múltiple son aplicables para el análisis simple suponiendo que el número de variables independientes es uno ($p = 1$).*

### 1.1.2. Regresión Lineal Multivariante

El caso multivariante es la generalización del caso de regresión lineal múltiple a varias variables dependientes.

**Definición 1.1.9.** *Sean $s$ variables dependientes $\{\boldsymbol{y_1}, ..., \boldsymbol{y_s}\}$, donde $\boldsymbol{y_j} = (y_{1j}, ..., y_{nj})$ para $j = \{1, ..., s\}$, $p$ variables independientes $\{\boldsymbol{x_1}, ..., \boldsymbol{x_p}\}$ para $n$ observaciones independientes, se define el modelo de **regresión lineal multivariante** como:*

$$\begin{pmatrix} y_{11} & y_{12} & \cdots & y_{1s} \\ y_{21} & y_{22} & \cdots & y_{2s} \\ \\ \vdots & \vdots & \ddots & \vdots \\ \\ y_{n1} & y_{n2} & \cdots & y_{ns} \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \\ \vdots & \vdots & \ddots & \vdots \\ \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix} \begin{pmatrix} \beta_{01} & \cdots & \beta_{0s} \\ \beta_{11} & \cdots & \beta_{1s} \\ \vdots & \ddots & \vdots \\ \beta_{p1} & \cdots & \beta_{ps} \end{pmatrix} + \begin{pmatrix} \varepsilon_{11} & \cdots & \varepsilon_{1s} \\ \varepsilon_{21} & \cdots & \varepsilon_{2s} \\ \\ \vdots & \ddots & \vdots \\ \\ \varepsilon_{n1} & \cdots & \varepsilon_{ns} \end{pmatrix} \tag{1.9}$$

*La expresión matricial queda reducida a $\boldsymbol{Y} = \boldsymbol{XB} + \boldsymbol{E}$, siendo $\boldsymbol{Y} = \{\boldsymbol{y_1}, ..., \boldsymbol{y_s}\}$ el conjunto de variables dependientes, siendo $\boldsymbol{X} = \{\boldsymbol{x_1}, ..., \boldsymbol{x_p}\}$ el conjunto de variables*

*independientes,* $\boldsymbol{B} = \{\boldsymbol{\beta_0}, ..., \boldsymbol{\beta_s}\}$ *la matriz de coeficientes del modelo y* $\boldsymbol{E} = \{\boldsymbol{\varepsilon_1}, ..., \boldsymbol{\varepsilon_s}\}$ *la matriz de errores donde la media de cada columna es nula* $E(\boldsymbol{\varepsilon_i}) = 0$ *y la covarianza entre columnas es de* $\sigma_{ik}$*,es decir,* $Cov(\boldsymbol{\varepsilon_i}, \boldsymbol{\varepsilon_k}) = \sigma_{ik}\boldsymbol{I}$ *donde* $i, k = \{1, ..., s\}$.

## 1.2. Descomposición en Valores Singulares

La Descomposición en Valores Singulares (SVD) permite factorizar una matriz rectangular $\boldsymbol{X}_{n\times p}$ de $n$ observaciones y $p$ variables a partir del producto de tres matrices.

Sin duda el gran potencial está en sus aplicaciones y así lo demuestra Strang [21]. La descomposición de una matriz por un lado facilita la optimización de procesos computacionales y también ha aportado grandes avances en la compresión de información. Para su explicación se demuestra cómo aproximar una matriz a otra de rango menor inspirándose en los textos [4] y [2].

En primer lugar se define la descomposición en valores singulares de una matriz y los conceptos básicos de diagonalización entendiendo la matriz $\boldsymbol{X}$ como endomorfismo.

**Definición 1.2.1.** *Sea* $E$ *un espacio vectorial de dimensión finita y* $\boldsymbol{X} : E \longrightarrow E$*, un endomorfismo. Se dice que* $\lambda \in \mathbb{C}$ *es un* ***valor propio*** *para* $\boldsymbol{X}$ *si existe un vector* $\boldsymbol{e} \in E$ *tal que* $\boldsymbol{X}(\boldsymbol{e}) = \lambda\boldsymbol{e}$*. El vector* $\boldsymbol{e} \neq 0$*, se llama* ***vector propio*** *de* $\boldsymbol{X}$ *asociado al valor propio* $\lambda$*.*

**Definición 1.2.2.** *Toda matriz* $\boldsymbol{X}$ *de dimensión* $n \times p$ *y de rango* $r$ *puede factorizarse como:*

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V'} \tag{1.10}$$

*Siendo:*

*a)* $\boldsymbol{U}$ *una matriz ortonormal de dimensión* $n \times n$*, es decir,* $\boldsymbol{U}\boldsymbol{U'} = \boldsymbol{U'}\boldsymbol{U} = \boldsymbol{I}$*, donde* $\boldsymbol{U'}$ *denota la matriz transpuesta de* $\boldsymbol{U}$*. Las columnas de* $\boldsymbol{U}$*,* $(\boldsymbol{u_1}, ..., \boldsymbol{u_n})$*, donde* $\boldsymbol{u_j} = (u_{1j}, ..., u_{nj})$ *con* $j = \{1, ..., n\}$ *se conocen como vectores singulares por la izquierda de* $\boldsymbol{X}$ *y son los vectores propios de la matriz* $\boldsymbol{X}\boldsymbol{X'}$*.*

*b)* $\boldsymbol{V}$ *una matriz ortonormal de dimensión* $p \times p$*, es decir,* $\boldsymbol{V}\boldsymbol{V'} = \boldsymbol{V'}\boldsymbol{V} = \boldsymbol{I}$*. Las columnas de* $\boldsymbol{V}$*,* $(\boldsymbol{v_1}, ..., \boldsymbol{v_p})$*, donde* $v_j = (v_{1j}, ..., v_{pj})$ *con* $j = \{1, ..., p\}$ *se conocen como vectores singulares por la derecha de* $\boldsymbol{X}$ *y son los vectores propios de la matriz* $\boldsymbol{X}\boldsymbol{X'}$*.*

*c)* $\boldsymbol{\Sigma}$ *una matriz diagonal de dimensión* $n \times p$*, es decir,* $\boldsymbol{\Sigma_{ij}} = 0$*, para los índices* $i \neq j$*, con* $i = \{1, ..., n\}$ *y* $j = \{1, ..., p\}$ *que contiene las raíces cuadradas de los valores propios no nulos de las matrices* $\boldsymbol{X}\boldsymbol{X'}$ *y* $\boldsymbol{X'}\boldsymbol{X}$*. Los elementos de la diagonal se denominan valores singulares de* $\boldsymbol{X}$ *y se encuentran ordenados de mayor a menor*

$\alpha_1 \geq \cdots \geq \alpha_r \geq 0, \quad con \quad r \leq min\{n, p\}$. *El conjunto de valores singulares se denomina espectro singular de la matriz* $\boldsymbol{X}$.

### 1.2.1. Propiedades de la descomposición

Se demuestran las afirmaciones recogidas en la definición anterior:

1.- $U_{n\times n}$ es la matriz de autovectores de $\boldsymbol{XX'}$

2.- $V_{p\times p}$ es la matriz de autovectores de $\boldsymbol{X'X}$

3.- Las raíces cuadradas de los autovalores no nulos de las matrices $\boldsymbol{XX'}$ y $\boldsymbol{X'X}$ son los valores singulares de $\boldsymbol{X}$.

*Demostración.*

1.- Partiendo de la descomposición en valores singulares de la matriz $\boldsymbol{X}$, se puede expresar el producto $\boldsymbol{XX'}$ como:

$$\boldsymbol{XX' = U\Sigma V'V\Sigma U' = U\Sigma^2 U'} \tag{1.11}$$

Gracias a la ortogonalidad de $\boldsymbol{V}$ y utilizando ahora la ortogonalidad de $\boldsymbol{U}$ se multiplica por $\boldsymbol{U}$ por la derecha a la expresión anterior:

$$\boldsymbol{XX'U = U\Sigma^2 U'U = U\Sigma^2} \tag{1.12}$$

Si llamamos $\boldsymbol{u_j}$ a las columnas de $\boldsymbol{U}$, tenemos $\boldsymbol{XX'u_j} = \alpha_j^2 \boldsymbol{u_j}$ siendo $\alpha_j^2$ valor propio de $\boldsymbol{XX'}$ para cada $j = \{1, ..., n\}$.

De donde se deduce que, efectivamente, $\boldsymbol{U}$ es el conjunto de vectores propios de $\boldsymbol{XX'}$.

2.- Análogamente, para el producto $\boldsymbol{X'X}$, utilizando la ortogonalidad de $\boldsymbol{U}$, se obtiene:

$$\boldsymbol{X'X = V\Sigma U'U\Sigma V' = V\Sigma^2 V'} \tag{1.13}$$

Recurriendo ahora a la ortogonalidad de $\boldsymbol{V}$:

$$\boldsymbol{X'XV = V\Sigma^2 VV' = V\Sigma^2} \tag{1.14}$$

De donde se deduce, siguiendo el mismo razonamineto que en la primera propiedad, que $\boldsymbol{V}$ es el conjunto de vectores propios de $\boldsymbol{X'X}$.

3.- Como ya se ha demostrado:

$$\begin{aligned} \boldsymbol{XX' = U\Sigma^2 U'} \quad &\Rightarrow \quad \boldsymbol{XX'U = U\Sigma^2} \\ \boldsymbol{X'X = V\Sigma^2 V'} \quad &\Rightarrow \quad \boldsymbol{X'XV = V\Sigma^2} \end{aligned} \tag{1.15}$$

Procediendo como en la primera propiedad, si denotamos por $\boldsymbol{u_j}$ a las columnas de $\boldsymbol{U}$, tenemos $\boldsymbol{XX'u_j}$=$\alpha_j^2\boldsymbol{u_j}$, luego $\alpha_j^2$ es valor propio de $\boldsymbol{XX'}$ para cada $j = \{1, ..., n\}$. Por la propiedad dos, se deduce que $\alpha_j^2$ para cada $j = \{1, ..., n\}$ son valores propios de $\boldsymbol{X'X}$, luego $\boldsymbol{\Sigma^2}$ es la matriz de valores propios de las matrices $\boldsymbol{XX'}$ y $\boldsymbol{X'X}$.

Además se demuestra que los valores propios no nulos de la matriz $\boldsymbol{X'X}$ son positivos y puede tomarse la raíz cuadrada sin complicaciones debido a que si $\boldsymbol{v_j} \in \boldsymbol{V} = (\boldsymbol{v_1}, ..., \boldsymbol{v_p})$ para $j = \{1, .., p\}$ es vector propio de $\boldsymbol{X'X}$ entonces se verifica $\forall\alpha_j^2 \in \mathbb{C}$:

$$\boldsymbol{X'Xv_j} = \alpha_j^2\boldsymbol{v_j} \Rightarrow \boldsymbol{v_j'X'Xv_j} = \alpha_j^2\boldsymbol{v_j'v_j} \tag{1.16}$$

Por la definición de norma:

$$\|\boldsymbol{Xv_j}\|^2 = \alpha_j^2\|\boldsymbol{v_j}\|^2 \Rightarrow \alpha_j^2 \geq 0 \tag{1.17}$$

Análogamente para $\boldsymbol{XX'}$ y los vectores propios de $\boldsymbol{U}$.

□

A continuación se mostrará cómo la Descomposición en Valores Singulares proporciona la mejor aproximación de rango menor de la matriz original. Eckart y Young lo demostraron en 1936 [2]. La aproximación de rango bajo es un problema de optimización que busca minimizar la norma de la diferencia de una matriz dada (la original) y su aproximación de rango menor. En este caso se busca minimizar la norma de Fröbenius.

**Definición 1.2.3.** *Sea una matriz $\boldsymbol{A}$ de dimensión $n \times p$, se denomina norma de Fröbenius de la mariz $\boldsymbol{A}$:*

$$\|\boldsymbol{A}\|_F = (traza(\boldsymbol{A'A}))^{\frac{1}{2}} \tag{1.18}$$

*Su forma equivalente viene dada por:*

$$\|\boldsymbol{A}\|_F = \left(\sum_{i=1}^{n}\sum_{j=1}^{p}|a_{ij}|^2\right)^{\frac{1}{2}} \tag{1.19}$$

**Teorema 1.2.1.** *(Teorema de Eckart y Young) Sea $\boldsymbol{X}$ una matriz de dimensión $n \times p$ y de rango $r$, con $\boldsymbol{X} = \boldsymbol{U\Sigma V'}$ su descomposición en valores singuares y siendo $\{\alpha_1, ..., \alpha_r\}$ los valores singulares de $\boldsymbol{\Sigma}$. Entonces, la mejor aproximación de rango $k \leq r$ es la matriz de dimensión $n \times p$ cuya descomposición en valores singulares es $\boldsymbol{X_k} = \boldsymbol{U\Sigma_k V'}$, siendo $\boldsymbol{\Sigma_k}$ la matriz diagonal que contiene los valores singulares $\{\alpha_1, ..., \alpha_k\}$ en el sentido de:*

$$\|\boldsymbol{X} - \boldsymbol{X_k}\|_F = \min\{\|\boldsymbol{X} - \boldsymbol{A}\|_F, \quad \forall\boldsymbol{A}_{n\times p}, \quad rango(\boldsymbol{A}) = k\} \tag{1.20}$$

*Demostración.* La descomposición en valores singulares de la matriz $\boldsymbol{X}$ aporta una solución teórica casi inmediata:

$$\|\boldsymbol{X} - \boldsymbol{A}\|_F = \|\boldsymbol{U\Sigma V'} - \boldsymbol{A}\|_F = \|\boldsymbol{\Sigma} - \boldsymbol{U'AV}\|_F \tag{1.21}$$

De ahora en adelante se llamará $\boldsymbol{U'AV} = \boldsymbol{N}$, una matriz de rango $k$ y dimensión $n \times p$. Aplicando la definición de la norma de Fröbenius se obtiene la siguiente expresión:

$$\|\boldsymbol{\Sigma} - \boldsymbol{N}\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{p}|\Sigma_{ij} - n_{ij}|^2 = \sum_{i=1}^{r}|\sigma_i - n_{ii}|^2 + \sum_{i>r}|n_{ii}|^2 + \sum_{i\neq j}|n_{ij}|^2 \tag{1.22}$$

La expresión anterior es mínima cuando los dos últimos términos son nulos, es decir, que los elementos que no pertenezcan a la diagonal de matriz $\boldsymbol{N}$ sean nulos y también aquellos de la diagonal a partir del elemento $r$-ésimo. Finalmente, para minimizar el primer término de la expresión $\sum_{i=1}^{r}|\sigma_i - n_{ii}|^2$, como de todos los $n_{ii}$ exáctamente $k$ son distintos de cero, el mínimo de la expresión se alcanza cuando $i = k$, es decir, cuando $\{n_{11}, ..., n_{kk}\}$ son no nulos y el resto de elementos de la diagonal hasta alcanzar el elemento $n_{rr}$ lo son. □

## 1.3. Análisis de Componentes Principales

El Análisis de Componentes Principales es una técnica estadística que tiene como objetivo proyectar un conjunto de datos sobre un espacio de dimensión inferior a la dimensión original mediante nuevas variables sin correlación para resumir la información de la matriz original y facilitar así su interpretación. Estas nuevas variables son las que se conocen como componentes principales y se calculan utilizando la descomposición espectral de la matriz de varianzas-covarianzas o la descomposición en valores singulares de la matriz de datos original. Esta metodología fue desarrollada por Karl Pearson [15] en 1901 y posteriormente fue estudiada por Hotelling [6] en los años 30.

### 1.3.1. Cálculo de las componentes principales

Para la redacción de esta sección se ha consultado [16], [11] y [14].

**Definición 1.3.1.** *Sea $\boldsymbol{X}$ la matriz de datos de dimensión $n \times p$ y de rango $r$ que recoge la información de $n$ observaciones medidas de $p$ variables, se asume que es centrada, sin pérdida de generalidad. Se define una* ***componente principal*** $\boldsymbol{y_j}$ *con* $j = \{1, ..., r\}$ *como la combinación lineal de las variables originales* $\{\boldsymbol{x_1}, ..., \boldsymbol{x_p}\}$ *y en la que los coeficientes establecen la ponderación óptima que maximice la variabilidad recogida por las nuevas variables* $\{\boldsymbol{y_1}, ..., \boldsymbol{y_r}\}$*. Es decir:*

$$\boldsymbol{Y_{n\times r}} = \boldsymbol{X_{n\times p}A_{p\times r}} \tag{1.23}$$

*donde* $\boldsymbol{Y_{n\times r}} = \{\boldsymbol{y_1}, ..., \boldsymbol{y_r}\}$ *es la matriz de las componentes principales,* $\boldsymbol{X_{n\times p}} = \{\boldsymbol{x_1}, ..., \boldsymbol{x_p}\}$ *es la matriz de datos originales y* $\boldsymbol{A_{n\times r}} = \{\boldsymbol{a_1}, ..., \boldsymbol{a_r}\}$ *es la matriz de coeficientes de la combinación lineal.*

### Cálculo de la primera componente

**Definición 1.3.2.** *La* ***primera componente principal*** *es aquella combinación lineal de las p variables originales* $\{\boldsymbol{x_1}, ..., \boldsymbol{x_p}\}$, $\boldsymbol{y_1} = \boldsymbol{X}\boldsymbol{a_1}$, *que maximiza la varianza de* $\boldsymbol{y_1}$*:*

$$\boldsymbol{y_1} = a_{11}\boldsymbol{x_1} + \cdots + a_{p1}\boldsymbol{x_p}, \quad donde \quad \boldsymbol{a_1} = (a_{11}, ..., a_{p1}) \tag{1.24}$$

Para su obtención, se calcula la varianza de $\boldsymbol{y_1}$ en términos de la matriz de covarianzas de las observaciones de partida y posteriormente se procederá a su maximización.

$$Var(\boldsymbol{y_1}) = \frac{\sum_{i=1}^{n} y_{i1}^2}{n} = \frac{1}{n}\boldsymbol{y_1'}\boldsymbol{y_1} = \frac{1}{n}\boldsymbol{a_1'}\boldsymbol{X'}\boldsymbol{X}\boldsymbol{a_1} = \boldsymbol{a_1'}\boldsymbol{S}\boldsymbol{a_1} \tag{1.25}$$

siendo $Var(\boldsymbol{y_1})$ la varianza de $\boldsymbol{y_1}$ y $\boldsymbol{S} = \frac{1}{n}\boldsymbol{X'}\boldsymbol{X}$ la matriz de covarianzas de $\boldsymbol{X}$.

**Observación 1.3.1.** *Según la expresión obtenida se podría maximizar la varianza cuanto se quiera simplemente aumentando el módulo del vector* $\boldsymbol{a_1}$*, es por ello que se restringe su módulo a la unidad, es decir,* $\boldsymbol{a_1'}\boldsymbol{a_1} = 1$*, con* $\boldsymbol{a_1} = (a_{11}, ..., a_{p1})$.

Para resolver el problema de optimización que se plantea, se puede recurrir al método de los multiplicadores de Lagrange. En este caso la función que se quiere maximizar es la expresión 1.25 y estaría sujeta a la restricción del módulo comentada en la observación 1.3.1. Aplicando el método de los multiplicadores de Lagrange se consigue lo siguiente:

$$F_1 = \boldsymbol{a_1'}\boldsymbol{S}\boldsymbol{a_1} - \lambda(\boldsymbol{a_1'}\boldsymbol{a_1} - 1) \tag{1.26}$$

Una vez expresado el problema así, se procede a la maximización de forma habitual anulando el gradiente:

$$\frac{\partial F_1}{\partial \boldsymbol{a_1}} = 2\boldsymbol{S}\boldsymbol{a_1} - 2\lambda\boldsymbol{I}\boldsymbol{a_1} = 0 \tag{1.27}$$

Siendo $\boldsymbol{S}\boldsymbol{a_1} = \lambda\boldsymbol{a_1}$ la solución a la ecuación anterior. Además la solución al problema revela que $\boldsymbol{a_1}$ es un vector propio de la matriz $\boldsymbol{S}$ y $\lambda$ su correspondiente valor propio. Para concretar qué valor propio es la solución se recurre a 1.27 y obtenemos:

$$\boldsymbol{S}\boldsymbol{a_1} = \lambda\boldsymbol{I}\boldsymbol{a_1} \tag{1.28}$$

Sustituyendo dicha expresión en 1.25:

$$Var(\boldsymbol{y_1}) = \frac{1}{n}\boldsymbol{y_1'}\boldsymbol{y_1} = \frac{1}{n}\boldsymbol{a_1'}\boldsymbol{X'}\boldsymbol{X}\boldsymbol{a_1} = \boldsymbol{a_1'}\boldsymbol{S}\boldsymbol{a_1} = \boldsymbol{a_1'}\lambda\boldsymbol{I}\boldsymbol{a_1} = \lambda \tag{1.29}$$

Luego, si se busca maximizar la varianza de la primera componente hay que escoger el mayor valor propio de $\boldsymbol{S}$, $\lambda = \lambda_1$, -siendo $\{\lambda_1, ..., \lambda_r\}$ los valores propios no nulos de $\boldsymbol{S}$ ordenados en orden decreciente- y su correspondiente vector propio $\boldsymbol{a_1}$, que será el encargado de definir los coeficientes de la combinación lineal.

### Cálculo de la segunda componente

**Definición 1.3.3.** *Se denomina* ***segunda componente principal*** *a aquella combinación lineal de las p variables originales* $\{\boldsymbol{x_1}, ..., \boldsymbol{x_p}\}$, $\boldsymbol{y_2} = \boldsymbol{X a_2}$, *que maximiza la varianza de* $\boldsymbol{y_2}$ *y es ortogonal a la primera componente principal:*

$$\boldsymbol{y_2} = a_{12}\boldsymbol{x_1} + \cdots + a_{p2}\boldsymbol{x_p}, \quad donde \quad \boldsymbol{a_2} = (a_{12}, ..., a_{p2}) \tag{1.30}$$

Para su obtención, se calcula la varianza de $\boldsymbol{y_2}$ en términos de la matriz de covarianzas de las observaciones de partida, centradas, y posteriormente se procederá a su maximización.

$$Var(\boldsymbol{y_2}) = \frac{\sum_{i=1}^{n} y_{i2}^2}{n} = \frac{1}{n}\boldsymbol{y_2'y_2} = \frac{1}{n}\boldsymbol{a_2'X'Xa_2} = \boldsymbol{a_2'Sa_2} \tag{1.31}$$

siendo $Var(\boldsymbol{y_2})$ la varianza de $\boldsymbol{y_2}$ y $\boldsymbol{S} = \frac{1}{n}\boldsymbol{X'X}$ la matriz de covarianzas de $\boldsymbol{X}$.

**Observación 1.3.2.** *Al igual que la primera componente principal, estará sujeta a la restricción del módulo del vector de cargas* $\boldsymbol{a_2}$*:* $\boldsymbol{a_2'a_2} = 1$*. Además se deberá verificar que la covarianza de* $\boldsymbol{y_1}$ *e* $\boldsymbol{y_2}$ *sea nula, puesto que las componentes principales son incorreladas por definiciaón.*

*Demostración.* Como $\boldsymbol{y_1}$ e $\boldsymbol{y_2}$ deben ser incorrelacionadas, entonces se tiene:

$$0 = Cov(\boldsymbol{y_2, y_1}) = Cov(\boldsymbol{a_2'X, a_1'X}) = \boldsymbol{a_2'Sa_1} \tag{1.32}$$

Utilizando ahora la expresión 1.28 se obtiene:

$$\boldsymbol{a_2'Sa_1} = \boldsymbol{a_2'\lambda Ia_1} = 0 \Leftrightarrow \boldsymbol{a_2'a_1} = 0 \tag{1.33}$$

Con lo que se demuestra que $a_1$ y $a_2$ son ortogonales. □

El problema de maximización que plantea la segunda componente principal consta de dos restricciones, es por ello que se recurre nuevamente al método de los multiplicadores de Lagrange para la optimización. En este caso la función a maximizar es la varianza de $\boldsymbol{y_2}$:

$$Var(\boldsymbol{y_2}) = \boldsymbol{a_2'Sa_2} \tag{1.34}$$

Y las restricciones son:

$$\left.\begin{array}{c} \boldsymbol{a_2'a_2} = 1 \\ \boldsymbol{a_2'Sa_2} = 0 \end{array}\right\} \tag{1.35}$$

Aplicando los multiplicadores de Lagrange para construir la función:

$$F_2 = \boldsymbol{a_2'Sa_2} - 2\mu(\boldsymbol{a_2'Sa_2}) - \lambda(\boldsymbol{a_2'a_2} - 1) \tag{1.36}$$

Anulando el gradiente con respecto a $\boldsymbol{a_2}$ y multiplicando por $\boldsymbol{a_1'}$ se consigue:

$$\frac{\partial F_2}{\partial \boldsymbol{a_2}} = 2\boldsymbol{Sa_2} - 2\mu\boldsymbol{Sa_1} - 2\lambda\boldsymbol{Ia_2} = 0 \quad \Rightarrow \quad \boldsymbol{a_1'Sa_2} - \mu\boldsymbol{a_1'Sa_1} - \lambda\boldsymbol{a_1'a_2} = 0 \tag{1.37}$$

Se utiliza que $\lambda_1$ es valor propio de $\boldsymbol{S}$ asociado al vector propio $\boldsymbol{a_1}$ para reescribir la igualdad anterior.

$$\lambda_1\boldsymbol{a_1'a_2} - \mu\boldsymbol{a_1'}\lambda_1\boldsymbol{a_1} - \lambda\boldsymbol{a_1'a_2} = 0 \tag{1.38}$$

Además, teniendo en cuenta que $\boldsymbol{a_1}$ y $\boldsymbol{a_2}$ son ortogonales y que $\boldsymbol{a_1}$ es de módulo unitario, se puede afirmar:

$$\mu\boldsymbol{a_1'}\lambda_1\boldsymbol{a_1} = 0 \Rightarrow \mu = 0 \tag{1.39}$$

Volviendo a la expresión 1.37 aplicando el resultado anterior:

$$\frac{\partial F_2}{\partial \boldsymbol{a_2}} = 2\boldsymbol{Sa_2} - 2\lambda\boldsymbol{Ia_2} = 0 \Rightarrow (\boldsymbol{S} - \lambda\boldsymbol{I})\boldsymbol{a_2} = 0 \tag{1.40}$$

Esta ecuación solo tiene solución si se cumple que $|\boldsymbol{S} - \lambda\boldsymbol{I}| = 0$, de donde se puede deducir que $\lambda$ es valor propio de $\boldsymbol{S}$. Aplicando este resultado a la expresión 1.34, se tiene que $Var(\boldsymbol{y_2}) = \lambda$. Para que sea máximo, se elige el mayor valor propio ($\lambda = \lambda_2$), teniendo en cuenta que $\lambda_1$ ya se había tomado para la primera componente y que $\lambda_2$ es el siguiente mayor en orden decreciente. Y será el vector propio asociado, $\boldsymbol{a_2}$, el vector de coeficientes de la combinación lineal que define la segunda componente principal.

### Cálculo de la j-ésima componente

Si se generaliza el cálculo, se puede definir la $j$-ésima componente principal (donde $j = \{1, ..., r\}$) como combinación lineal de la variables originales como $\boldsymbol{y_j} = \boldsymbol{Xa_j}$, siendo $\boldsymbol{a_j}$ el vector propio de $\boldsymbol{S}$ asociado al $j$-ésimo valor propio. Además, se asegura

que $Var(\boldsymbol{y_j}) = \lambda_j$, es decir, se concluye que la matriz de covarianzas para la matriz $\boldsymbol{Y}$ (formada por $\{\boldsymbol{y_1}, ..., \boldsymbol{y_r}\}$) es:

$$\boldsymbol{\Lambda} = \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \lambda_r \end{pmatrix} \tag{1.41}$$

Utilizando la construcción de las componentes principales se obtiene otra expresión que será de gran utilidad en resultados siguientes:

$$\boldsymbol{\Lambda} = Var(\boldsymbol{Y}) = \boldsymbol{A'}Var(\boldsymbol{X})\boldsymbol{A} = \boldsymbol{A'SA} \tag{1.42}$$

siendo $\boldsymbol{A} = \{\boldsymbol{a_1}, ..., \boldsymbol{a_r}\}$ la matriz que contiene los vectores propios de la matriz $\boldsymbol{S}$, matriz de covarianzas de la matriz $\boldsymbol{X}$, y siendo $\boldsymbol{A'}$ la transpuesta de la matriz $\boldsymbol{A}$.

### 1.3.2. Variabilidad de las componentes principales

Sabiendo que la variabilidad de la componente principal $j$-ésima, donde $j = \{1, ..., r\}$, es equivalente al valor propio $j$-ésimo en orden descendente, se puede afirmar que la variabilidad total de la matriz será la traza de la matriz de covarianzas de las componentes principales.

$$\sum_{j=1}^{r} Var(\boldsymbol{y_j}) = \sum_{j=1}^{r} \lambda_j = tr(\boldsymbol{\Lambda}) \tag{1.43}$$

siendo $tr(\boldsymbol{\Lambda})$ la traza de la matriz de covarianzas de $\boldsymbol{Y}$. Sería de gran interés establecer la relación de este resultado con la matriz de covarianzas de $\boldsymbol{X}$, es decir, con respecto a $\boldsymbol{S}$. Utilizando la expresión 1.43 y que las matrices $\boldsymbol{A'}$ y $\boldsymbol{A}$ son ortogonales obtenemos:

$$tr(\boldsymbol{\Lambda}) = tr(\boldsymbol{A'SA}) = tr(\boldsymbol{SA'A}) = tr(\boldsymbol{S}) \tag{1.44}$$

De este resultado no sólo se concluye que ambas trazas coinciden, sino que eso implica que la cantidad de variabilidad explicada por las componentes principales y por las variables originales es exáctamente la misma:

$$tr(\boldsymbol{\Lambda}) = tr(\boldsymbol{S}) = \sum_{i=1}^{p} Var(\boldsymbol{x_i}) \tag{1.45}$$

Esta relación permitirá expresar la variabilidad de cada componente con respecto a la variabilidad total de la variables originales. El porcentaje de varianza explicado por las $k$ primeras componentes será:

$$P_1 + \cdots + P_k = 100 \cdot \frac{\lambda_1 + \cdots + \lambda_k}{\lambda_1 + \cdots + \lambda_r} = 100 \cdot \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{p} Var(\boldsymbol{x_i})} \tag{1.46}$$

Siendo $P_i$ el porcentaje de variabilidad explicado por la componente principal $i$-ésima, $i = \{1, ..., k\}$.

Es conveniente recordar que el objetivo perseguido con el Análisis de Componentes Principales es la reducción de la dimensionalidad de un conjunto de datos proyectándolos sobre un espacio de dimensión menor a la original. Conocer la varianza que alberga cada componente principal ayuda a saber la variabilidad que guarda la proyección en el espacio de dimensión menor o visto desde la perspectiva complementaria, ayuda a conocer si se pierde información y en qué medida. La siguiente sección se destina a proporcionar distintos criterios que ayudan a determinar qué componentes son suficientes para explicar un cierto porcentaje de la variabilidad total.

### Número de componentes principales

Se ha comprobado que si tenemos $p$ variables originales de una matriz $\boldsymbol{X}$ de rango $r$, $r$ son las componentes necesarias para reproducir la totalidad de la variabilidad de los datos. Sin embargo, en la práctica, dadas las propiedades de las componentes, que absorben varianza en orden decreciente de importancia, es habitual retener las $k$ primeras componentes para reproducir los datos, siendo $k << r$.

La elección del número de componentes depende del investigador y para tomar la decisión tiene distintos criterios a su disposición.

**Definición 1.3.4.** *Se denomina **criterio del porcentaje** a la toma de decisión que consiste en seleccionar las componentes principales de modo que la variabilidad explicada acumulada por ellas supere un porcentaje fijado por el analista, por ejemeplo 90 %.*

**Definición 1.3.5.** *Se conoce como **criterio del bastón roto** al criterio que asocia la variabilidad total de los datos con un bastón de longitud L. Si se divide dicho bastón en p partes desiguales, se podría interpretar cada tramo como la variabilidad explicada por cada componente principal ($l_1 > \cdots > l_r$), es decir, los valores propios ordenados. Entonces se supone que para el valor $l_j$, con $j = \{1, ..., r\}$, se tiene:*

$$E(l_j) = \frac{1}{r} \sum_{i=1}^{r-j} \frac{1}{j+i} \tag{1.47}$$

*Las k primeras componentes principales serán suficientes si el porcentaje de variabilidad que representan supera el valor acumulado esperado para esa partición.*

**Definición 1.3.6.** *Se denomina **gráfico de sedimentación** a la representación ordenada del número de componentes principales versus su valor propio correspondiente. Como criterio de decisión establece un símil entre la ladera de una montaña (primero*

*escarpada y luego cada vez más plana creando la zona de sedimentación) y propone escoger aquellas componentes princiaples previas a la zona de sedimentación.*

El lector puede encontrar más criterios en [3]. Se trata de hallar el equilibrio entre la variabilidad explicada de los datos y el número de componentes principales retenidas.

Además del enfoque de maximización de la variabilidad para el cálculo de las componentes principales, la aparición de SVD permitió el cálculo de las componentes principales como $\boldsymbol{Y} = \boldsymbol{XV}$. Siendo $\boldsymbol{X} = \boldsymbol{U\Sigma V'}$ la descomposición en valores singulares de la matriz $\boldsymbol{X}$ y $\boldsymbol{V}$ la matriz de cargas del ACP, obtenida en la descomposición de $\boldsymbol{X}$. De esta manera $\boldsymbol{Y} = \boldsymbol{XV}$ y también $\boldsymbol{XV} = \boldsymbol{U\Sigma}$, por lo que se consigue $\boldsymbol{Y} = \boldsymbol{U\Sigma}$. Es decir, surge una nueva interpretación de las componentes principales como una versión reescalada de las matrices de vectores propios.

## 1.4. Análisis de cluster

El Análisis de Cluster es un conjunto de técnicas estadísticas multivariantes no supervisadas que tienen como objetivo agrupar una serie de elementos buscando la máxima similitud entre los objetos del mismo conjunto y la mayor diferencia para aquellos objetos pertenecientes a grupos distintos.

A continuación se detallarán algunos de los conceptos necesarios para comprender el enfoque de Cuadras [3] para el análisis de cluster:

**Definición 1.4.1.** *Se considera $\mathcal{R}$ una **relación de equivalencia** sobre $\Omega$, un conjunto finito de $n$ elementos, si verifica las siguientes propiedades:*

1. *Reflexiva:* $\forall x \in \Omega, \quad x\mathcal{R}x$
2. *Simétrica:* $\forall x, y \in \Omega, \quad x\mathcal{R}y \Leftrightarrow y\mathcal{R}x$
3. *Transitiva:* $\forall x, y, z \in \Omega, \quad$ *si* $x\mathcal{R}y$ *y a su vez* $y\mathcal{R}z \Rightarrow x\mathcal{R}z$

La interpretación de la definición permite establecer la conexión con el análisis de cluster. Por ejemplo, la propiedad reflexiva significa que una estación meteorológica pertenece a la misma zona climatológica que ella misma; la simétrica indica que si una estación meteorológica $x$ pertenece a la misma zona que una estación $y$, entonces, la estación $y$ pertenece a la misma zona climatológica que la estación $x$; y la transitiva significa que si la estación $x$ pertenece a la misma zona que la estación $y$, y a su vez la estación $y$ pertenece a la misma zona que la estación $z$, entonces, la estación $x$ y la estación $z$ están en la misma zona.

De manera natural, al considerarse una relación de equivalencia, surge el concepto de clase de equivalencia.

**Definición 1.4.2.** *Se denomina* ***clase de equivalencia*** *de un elemento $x$, según una relación de equivalencia $\mathcal{R}$, definida sobre el conjunto $\Omega$, al subconjunto de elementos relacionados con $x$ mediante $\mathcal{R}$ y que pertenecen al conjunto $\Omega$, es decir, $\forall y \in \Omega \mid x\mathcal{R}y$.*

**Definición 1.4.3.** *Se denomina* ***partición*** *del conjunto $\Omega$ a la familia de las clases de equivalencia definidas sobre $\Omega$ acuerdo a la relación de equivalencia $\mathcal{R}$.*

**Nota 1.4.1.** *En cuanto a la nomenclatura, a las clases de equivalencia se les llamará clusters y a la partición clustering.*

Luego, dado un conjunto finito $\Omega$ de $n$ elementos, desarrollando la idea de Cuadras [3], se presenta el proceso de clasificación como definición de una relación de equivalencia $\mathcal{R}$ sobre $\Omega$.

La relación $\mathcal{R}$ permite dividir $\Omega$ en $m$ clases de equivalencia disjuntas $\{h_1, ..., h_m\}$:

$$\Omega = h_1 \cup \cdots \cup h_m \tag{1.48}$$

De tal manera que atendiendo a un determinado criterio, los componentes de los clusters guarden homogeneidad y los clusters entre sí sean tan heterogéneos como sea posible.

### 1.4.1. Criterios de semejanza

Para tener la capacidad de decidir si un elemento es homogéneo a otro, es necesario establecer un criterio que determine qué grado de semejanza relaciona a ambos elementos, dependiendo de si los elementos a clasificar son objetos o variables numéricas.

#### Distancias entre objetos

**Definición 1.4.4.** *Se denomina* ***distancia*** *a la aplicación $d : \Omega \times \Omega \longrightarrow \mathbb{R}^+$ definida sobre un conjunto no vacío $\Omega$, que verifica las siguientes propiedades:*

1. $d(x, y) = 0 \Leftrightarrow x = y, \quad \forall x, y \in \Omega$
2. $d(x, y) \geq 0, \quad \forall x, y \in \Omega$
3. $d(x, y) = d(y, x), \quad \forall x, y \in \Omega$
4. $d(x, z) \leq d(x, y) + d(y, z), \quad \forall x, y, z \in \Omega$

Respecto a la elección de utilizar un tipo de distancia u otra depende de factores subjetivos. A continuación se exponen las más comunes:

- **Distancia euclídea**: Dados dos objetos $\boldsymbol{x} = (x_1, x_2)$ e $\boldsymbol{y} = (y_1, y_2)$, la distancia que separa a ambos se puede expresar como:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \Omega \tag{1.49}$$

En el supuesto de que estén determinados por $p$ variables $\{\boldsymbol{x_1}, ..., \boldsymbol{x_p}\}$ , se puede formular como sigue:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \left( \sum_{k=1}^{p} (x_k - y_k)^2 \right)^{\frac{1}{2}} \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \Omega \tag{1.50}$$

Equivalentemente:

$$d(\boldsymbol{x}, \boldsymbol{y}) = ((\boldsymbol{x} - \boldsymbol{y})'(\boldsymbol{x} - \boldsymbol{y}))^{\frac{1}{2}}, \quad i, j = \{1, ..., p\} \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \Omega \tag{1.51}$$

- **Distancia Minkowski**: Esta distancia puede considerarse una generalización de la distancias anterior, ya que para $m = 1$, se obtiene la distancia valor absoluto y para $m = 2$ se obtiene la distancia euclídea.

$$d(\boldsymbol{x}, \boldsymbol{y}) = \left( \sum_{k=1}^{p} |x_k - y_k|^m \right)^{\frac{1}{m}}, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \Omega \tag{1.52}$$

- **Distancia Mahalanobis**: Representa la distancia entre los objetos $\boldsymbol{x}$ y $\boldsymbol{y}$ considerando la covarianza entre las variables utilizadas en la medición. Resulta de gran utilidad para el caso en el que las variables no se encuentran medidas en la misma escala y se quiere relativizar su importancia. Así, las variables con menos varianza tendrán más importancia que las de mayor varianza.

$$d(\boldsymbol{x}, \boldsymbol{y}) = ((\boldsymbol{x} - \boldsymbol{y})' \boldsymbol{S}' (\boldsymbol{x} - \boldsymbol{y}))^{\frac{1}{2}} \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \Omega \tag{1.53}$$

Siendo $\boldsymbol{S}$ la matriz de covarianzas de las $p$ variables $\{\boldsymbol{x_1}, ..., \boldsymbol{x_p}\}$.

### Distancias entre variables

Para establecer una medida de semejanza entre variables es recurrente el uso de coeficientes de correlación:

- **Coeficiente de correlación de Pearson:** Es una medida utilizada para calcular la dependencia lineal entre un par de variables. Dadas las variables $\boldsymbol{x_1}$ e $\boldsymbol{x_2}$, $S_{\boldsymbol{x_1 x_2}}$ su covarianza y $S_{\boldsymbol{x_1}}$, $S_{\boldsymbol{x_2}}$ las respectivas desviaciones estándar:

$$r_{\boldsymbol{x_1 x_2}} = \frac{S_{\boldsymbol{x_1 x_2}}}{S_{\boldsymbol{x_1}} S_{\boldsymbol{x_2}}} \tag{1.54}$$

Su valor oscila entre $[-1, 1]$ con una correlación de -1 en caso de que sean inversamente proporcionales perfectas y 1 para una relación proporcional perfecta. Si $r = 0$, no existe relación lineal, aunque no implica que pueda ser de otro tipo. Para $r \in (0, 1)$, existe una correlación positiva y para $r \in (-1, 0)$, la correlación es negativa.

Ante la limitación que presenta el coeficiente de correlación de Pearson (únicamente mide la dependencia lineal), se ofrecen otras alternativas:

- **Coeficiente de correlación de rangos de Spearman:** Se analizan $n$ observaciones ordenadas conforme a dos variables diferentes $\boldsymbol{x_1}$ e $\boldsymbol{x_2}$. Para ello, primero se calculan los rangos y se emparejan para ambas variables $(r_{x_{11}}, r_{x_{12}}), ..., (r_{x_{n1}}, r_{x_{n2}})$. Se definen además, las diferencias $d_i := (r_{x_{i1}} - r_{x_{i2}})$, es decir, la diferencia de los valores dados por las variables $\boldsymbol{x_1}$ e $\boldsymbol{x_2}$ para la observación $i$-ésimo, con $i = \{1, ..., n\}$. El coeficiente se puede expresar entonces como:

$$r_s = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2 - 1)} \tag{1.55}$$

  La interpretación del coeficiente de correlación de rangos de Spearman es equivalente a la de Pearson, sin embargo, es menos sensible a valores atípicos, por lo que es mejor que la correlación de Pearson para muestras con *outliers*.

- **Coeficiente de correlación de rangos de Kendall:** Se analizan $n$ observaciones ordenadas conforme a dos variables diferentes $\boldsymbol{x_1}$ e $\boldsymbol{x_2}$. Al igual que la medida anterior, se construyen las parejas $(r_{x_{11}}, r_{x_{12}}), ..., (r_{x_{n1}}, r_{x_{n2}})$. Se calculan el número de pares concordantes $(n_c)$ y pares discordantes $(n_d)$. Se considera par concordante si para cada pareja $(x_{i1}, x_{in})$ y $(x_{j1}, x_{jn})$, donde $i < j$ si $x_{i1} < x_{j1}$ y $x_{i2} < x_{j2}$, o $x_{i1} > x_{j1}$ y $x_{i2} > x_{j2}$ con $i, j = \{1, ..., n\}$. En caso contrario, se conoce como discordancia. El número de posibles parejas para $n$ observaciones es $\frac{n(n-1)}{2}$, luego el coeficiente resultante es:

$$\tau = \frac{n_c - n_d}{\frac{n(n-1)}{2}} \tag{1.56}$$

Una de las características más representativas del análisis cluster es que se desconocen a priori el número de clusters finales de los datos. Además se hace evidente que las posibilidades existentes para repartir los elementos entre los grupos de distintos tamaños son elevadas y en algunos casos llegan a ser inabarcables, es por ello que se definirán algoritmos de clasificación que resuelvan dicha tarea de forma eficaz.

Las técnicas de análisis de cluster forman dos grandes familias:

1. **Métodos jerárquicos:** Obtienen una sucesión de particiones, en la que cada cluster es resultado de la agrupación (Métodos jerárquicos aglomerativos) o división (Métodos jerárquicos divisivos) de otros clusters.

2. **Métodos no jerárquicos:** Establecen clusters aleatorios y mediante sucesivas iteraciones se reasignan las observaciones hasta que se estabilizan los grupos.

### 1.4.2. Métodos de cluster jerárquicos

**Definición 1.4.5.** *Un **clustering jerárquico** es una sucesión de clusterings en la que cada uno de ellos es resultado de una agrupación o separación de clusters. Puede ser aglomerativo o divisivo.*

**Definición 1.4.6.** *Se conoce como **clasificación jerárquica aglomerativa** a aquella que tiene como punto de partida los n elementos en n clusters, agrupa en primer lugar los que son más similares según alguna medida de similitud y culmina el proceso llegando a la agrupación de los n elementos en un solo cluster.*

**Definición 1.4.7.** *Se denomina **clasificación jerárquica divisiva** a aquella que representa el flujo inverso de la aglomerativa. Se inicia el método con los n elementos agrupados en un único cluster y se separan primero aquellos subconjuntos cuya distancia sea mayor hasta clasificar los n elementos en n clusters.*

Surge, por tanto, de manera natural el concepto de jerarquía indexada para abordar desde una perspectiva matemática la estructura definida anteriormente.

**Definición 1.4.8.** *Sea $\Omega$ un conjunto finito de n elementos, $\mathcal{P}(\Omega)$ el conjunto de partes de $\Omega$ y $H \subset \mathcal{P}(\Omega)$ un clustering de $\Omega$, se define el **índice $\alpha$** como la aplicación $\alpha : H \longrightarrow \mathbb{R}^+$ que cumple las siguientes propiedades:*

1. $\alpha(i) = 0 \quad \forall i = \{1, ..., n\} \in \Omega$ *siendo i los elementos del conjunto finito* $\Omega$.
2. $\alpha(h) \leq \alpha(h') \quad si \quad h \subset h', \quad h, h' \in H$

**Definición 1.4.9.** *Sea $\Omega$ un conjunto finito, $\mathcal{P}(\Omega)$ el conjunto de partes de $\Omega$ y $H \subset \mathcal{P}(\Omega)$ un clustering de $\Omega$. Se dice que $(H, \alpha)$ es una **jerarquía indexada** si cumple las siguientes propiedades:*

*1)* $Si\ h_1, h_2 \in H \Rightarrow h_1 \bigcap h_2 \in \{h_1, h_2, \phi\}$

*2)* $Si\ h \in H \Rightarrow h = \bigcup \{h' \mid h' \in H, h' \subset h\}$

*3)* $\Omega = \bigcup \{h \mid h \in H\}$

A continuación se presenta la teoría matemática que sustenta la representación gráfica de las jerarquías indexadas.

#### Geometría ultramétrica

Una de las formas más intuitivas de representar una clasificación es un esquema de llaves en el que se ramifican las distintas categorías. Una generalización de este tipo de esquemas es el árbol ultramétrico. Para explicar esta idea, Cuadras [3] se apoya en la geometría ultramétrica. En primer lugar se definen dos conceptos fundamentales para después demostrar varias propiedades de gran interés.

**Definición 1.4.10.** *Se denomina* ***distancia ultramétrica*** *a la aplicación* $u : \Omega \times \Omega \longrightarrow \mathbb{R}^+$ *definida sobre un conjunto no vacío* $\Omega$*, que verifica,* $\forall x, y, z \in \Omega$*, las siguientes propiedades:*

1. $u(x, y) = 0 \Leftrightarrow x = y$
2. $u(x, y) \geq 0$
3. $u(x, y) = u(y, x)$
4. $u(x, z) \leq \sup\{u(x, y), u(z, y)\}$

*Se denomina* ***espacio ultramétrico*** *al par* $(\Omega, u)$*.*

**Definición 1.4.11.** *Sea* $\Omega$ *un conjunto finito de* $n$ *elementos, un* ***árbol ultramétrico*** *o* ***dendograma*** *es un grafo conexo, sin ciclos, formado en uno de los extremos por un único nodo (raíz) y en el otro por* $n$ *nodos equidistantes de la raíz, siendo estos los elementos del conjunto* $\Omega$*.*

Un ejemplo de dendograma se puede encontrar en la figura 1.1 del teorema 1.4.1.

**Teorema 1.4.1.** *Todo espacio ultramétrico* $(\Omega, u)$ *puede representarse mediante un árbol ultramétrico.*

*Demostración.* La demostración se apoya en la figura 1.1. Sean $x, y, z \in \Omega$, la distancia que separa a los extremos $x$ de $y$ se puede medir como la distancia de cada uno de ellos al nodo que les liga. Se supone que dicho nodo es $\gamma$, luego, $u(x, y) = \gamma$. Se construye un triángulo $\{x, y, z\}$, siendo el lado $x - y$ el más pequeño. Por tanto $z$ se relaciona con $\{x, y\}$ en un nodo $\gamma'$ por encima de $\gamma$. Para ser un árbol ultramétrico debe verificarse que la distancia de todos los extremos a la raíz sea igual. Considerando a $\gamma'$ la distancia al nudo raíz, $u(x, z) = u(y, z) = u(x, y) + \beta, \quad \beta = \gamma' - \gamma$, luego queda demostrado que $\{x, y, z\}$ es un árbol ultramétrico. □

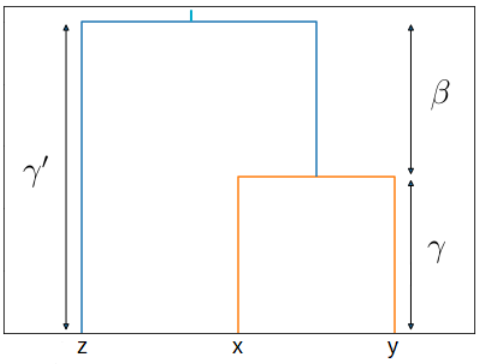



Figura 1.1: Gráfico explicativo teorema 1.4.1. Fuente: elaboración propia.

Otra propiedad importante está relacionada con cómo se comporta la distancia ultramétrica si se agrupan elementos que la cumplían previamente.

**Teorema 1.4.2.** *Sea $u$ la distancia ultramétrica definida sobre $\Omega = h_1 \cup \cdots \cup h_m$. Si se unen dos de los $m$ clusters, $h_i, h_j$ con $i, j \in \{1, ..., m\}$, se puede definir una nueva distancia ultramétrica $u'$ sobre los $m-1$ clusters resultantes.*

*Demostración.* Si $k \neq i, j$, como $u$ es ultramétrica se verifica la siguiente propiedad:

$$\begin{aligned} u(h_i, h_j) &\leq \sup\{u(h_i, h_k), u(h_j, h_k)\} \\ u(h_i, h_k) &= u(h_j, h_k) \end{aligned} \tag{1.57}$$

Por unirse $h_i, h_j$ con $i, j \in \{1, ..., m\}$. Se define la nueva distancia $u'$:

$$u'(h_k, h_i \cup h_j) := u(h_k, h_i) = u(h_k, h_j) \tag{1.58}$$

Siendo $h_i$ y $h_j$ los clusters más cercanos. Para el resto de elementos que no intervienen en la unión de clusters las distancias son equivalentes:

$$u'(h_a, h_b) := u(h_a, h_b) \quad a, b \neq i, j \tag{1.59}$$

Hay que demostrar que $u'$ es ultramétrica sabiendo que $u$ lo es. Por consiguiente se toma el triángulo $\{i, j, k\}$.

$$\begin{aligned} u'(h_a, h_b) = u(h_a, h_b) \leq \sup\{u(h_a, h_i), u(h_b, h_i)\} = \sup\{u'(h_a, h_i \cup h_j), u'(h_b, h_i \cup h_j)\} \\ u'(h_a, h_i \cup h_j) = u(h_a, h_i) \leq \sup\{u(h_a, h_b), u(h_b, h_i)\} = \sup\{u'(h_a, h_b), u'(h_b, h_i \cup h_j)\} \end{aligned} \tag{1.60}$$

En conclusión, queda demostrado que $u'$ es ultramétrica sobre los $m-1$ clusters del clustering. □

### Algoritmos de clasificación

Este apartado está dedicado a mostrar las distintas metodologías que se pueden seguir para llevar a cabo un proceso de cluster jerérquico apoyándose en el teorema 1.4.2:

**Teorema 1.4.3** (Algoritmo Fundamental de clasificación)**.** *Se puede definir un algoritmo capaz de construir una jerarquía indexada dado un espacio ultramétrico $(\Omega, u)$.*

*Demostración.* Para un clustering $\Omega = h_1 \cup \cdots \cup h_m$, se supone que los clusters $h_i$ y $h_j$ son los más próximos según alguna distancia $d$ y se decide unirlos en un mismo cluster:

$$\{h_i\} \cup \{h_j\} = \{h_i, h_j\} \tag{1.61}$$

Se define una nueva distancia $u'$ que por el teorema 1.4.2 se puede asegurar que es ultramétrica sobre los $m-1$ clusters restantes:

$$u'(h_k, \{h_i, h_j\}) = u(h_i, h_k) = u(h_j, h_k) \quad k \neq i, j \tag{1.62}$$

Se considera ahora la nueva partición $\Omega = h_1 \cup \cdots + \{h_i \cup h_j\} \cup \cdots \cup h_m$. Se repite el proceso hasta que $\Omega = \Omega$. En cada iteración, cada vez que se produce la unión de dos clusters $h_i$ y $h_j$ por ser los más próximos, a su vez se define el índice:

$$\alpha(h_i \cup h_j) = u(h_i, h_j) \tag{1.63}$$

El resultado obtenido al finalizar el proceso es una jerarquía indexada $(H, \alpha)$. □

**Observación 1.4.1.** *Dada una jerarquía indexada $(H, \alpha)$, se puede definir una distancia $u$ ultramétrica sobre el espacio $\Omega$. Por tanto, una jerarquía indexada y un espacio ultramétrico son estructuras equivalentes.*

Se supone ahora que se ha medido la distancia de $n$ elementos de $\Omega$ en base a variables observables haciendo uso de la distancia $d$. Dichos resultados se recogen en la matriz $\Delta_{n\times n}$ de distancias entre los $n$ elementos de $\Omega$:

$$\Delta = \begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{nn} \end{pmatrix} \tag{1.64}$$

De acuerdo al teorema 1.4.3 dado un espacio ultramétrico no hay inconveniente en construir una jerarquía indexada, sin embargo, la distancia $d$ no tiene por qué ser ultramétrica. Para asegurar que sea ultramétrica habrá que escoger debidamente una función para transformarla. Dependiendo de la elección de la función y por ende el método de unión, se distinguen varios métodos de clasificación. Si la distancia $d$ no es ultramétrica la transformación $f$ será de tipo:

$$d'(h_k, \{h_i, h_j\}) = f\{d(h_i, h_k), d(h_j, h_k)\} \tag{1.65}$$

**Definición 1.4.12.** *Dada la transformación de la ecuación 1.65, se denomina algoritmo de* ***mínima distancia*** *a aquel que establece:*

$$d'(h_k, \{h_i, h_j\}) = \min\{d(h_i, h_k), d(h_j, h_k)\} \tag{1.66}$$

*Con la elección adecuada para $f$ se consigue que el triángulo $\{i, j, k\}$ con $d(h_i, h_j) \leq d(h_i, h_k) \leq d(h_j, h_k)$, se transforme en un triángulo ultramétrico verificando $d'(h_i, h_j) \leq d'(h_i, h_k) = d'(h_j, h_k)$.*

La clasificación irá aglutinando los clusters más próximos como se ha descrito en el teorema 1.4.3 y la distancia ultramétrica definida para el resto de clusters en cada iteración se construirá a partir de $f$.

**Definición 1.4.13.** *Dada la transformación de la ecuación 1.65, se denomina algoritmo de* ***máxima distancia*** *a aquel que establece:*

$$d'(h_k, \{h_i, h_j\}) = \max\{d(h_i, h_k), d(h_j, h_k)\} \tag{1.67}$$

*Se consigue así, que el triángulo* $\{i, j, k\}$ *con* $d(h_i, h_j) \leq d(h_i, h_k) \leq d(h_j, h_k)$*, se convierta en un triángulo ultramétrico verificando* $d'(h_i, h_j) \leq d'(h_i, h_k) = d'(h_j, h_k)$*.*

**Observación 1.4.2.** *Si se utiliza uno de los algoritmos de los expuestos anteriormente, intrínsicamente se establece la noción de distancia entre dos clusters y se define con la elección del método.*

#### Número de clusters

Los métodos de cluster jerárquicos no requieren establecer a priori el número de clusters para realizar el agrupamiento, sin embargo, al final del proceso y gracias a la representación del dendograma, se puede escoger fácilmente el número de clusters, buscando el equilibrio entre el número de clusters y el número de elementos apoyándose en la distancia que los separa.

En la figura 1.2 se ilustra cómo eligiendo una distancia entre clusters u otra, se obtiene una configuración de clusters distinta. Si se elige una distancia superior a seis, los componentes se dividen en dos clusters de dos elementos cada uno y si se elige una distancia inferior a uno, se obtendrían cinco clusters de un elemento.

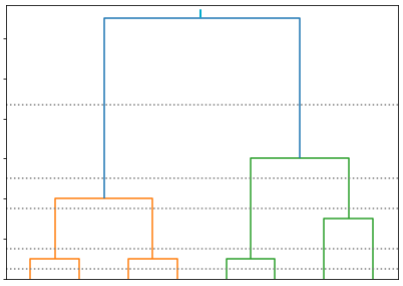


Figura 1.2: Número de clusters a partir de dendograma. Fuente: elaboración propia.

### 1.4.3. Métodos de clustering no jerárquicos

A diferencia de los métodos de cluster jerárquicos, los métodos no jerárquicos, como refleja su nombre, niegan cualquier tipo de estructura ordenada, es decir, no existe

ninguna dependencia ascendente o descendente entre los clusters. Es por ello que se presentarán nuevas técnicas de clasificación para las que no serán necesarios los fundamentos teóricos de la geometría ultramétrica expuestos para la clasificación jerárquica. Además, proporcionan una única partición de los datos. Sin embargo, los métodos no jerárquicos requieren conocer el número de clusters deseado antes de comenzar el proceso. Estos grupos no suelen estar definidos o ser conocidos, luego se realizarán varias iteraciones comprobando distintas configuraciones para después evaluar cuál se ajusta más al problema o facilite una mejor interpretación de los datos. Los métodos no jerárquicos también se denominan métodos partitivos aludiendo a su principal objetivo -construir una partición de $m$ clases-. La adjudicación de los elementos a los clusters se realiza atendiendo a un criterio específico.

**Definición 1.4.14.** *El **análisis de clustering no jerárquico** trata de construir $m$ clusters homogéneos excluyentes con máxima divergencia entre ellos. El número $m$ se fija antes del análisis y las $m$ clases forman una partición única.*

Sea $\boldsymbol{X_{n\times p}}$ una matriz de $n$ observaciones y $p$ variables $\boldsymbol{x_1}, ..., \boldsymbol{x_p}$ para clasificar en $m$ clusters. Si $\boldsymbol{W}$ es la matriz de covarianzas dentro de los clusters, $\boldsymbol{B}$ la matriz de covarianzas entre clusters y $\boldsymbol{T}$ la matriz de covarianzas del total, entonces:

$$\boldsymbol{T} = \boldsymbol{B} + \boldsymbol{W} \tag{1.68}$$

Siendo $\boldsymbol{T} = \boldsymbol{X'X}$, $\boldsymbol{B} = \bar{\boldsymbol{X}}'\boldsymbol{D}\bar{\boldsymbol{X}}$, siendo $\bar{\boldsymbol{X}}$ la matriz de medias para cada uno de los clusters en las $p$ variables y $\boldsymbol{D}$ una matriz diagonal de dimensión $m \times m$ y de donde se puede deducir el valor de $\boldsymbol{W}$ como $\boldsymbol{W} = \boldsymbol{T} - \boldsymbol{B}$.

El problema matemático de optimización subyacente en el análisis de cluster no jerárquico tiene varias alternativas para su resolución; ya sea minimizar la $tr(\boldsymbol{W})$, denotando $tr()$ la traza, minimizar el $Det(\boldsymbol{W})$, denotando $Det()$ el determinante, el $Det(\boldsymbol{W})/Det(\boldsymbol{T})$, o maximizar la $tr(\boldsymbol{W}^{-1}\boldsymbol{B})$. El objetivo final independientemente del procedimiento escogido es conseguir una buena clasificación, es decir, aquella que minimice la dispersión dentro de cada grupo. Este principio recibe el nombre de criterio de la varianza [3]. Aunque existen distintas técnicas de cluster no jerárquico, serán los métodos de reasignación a los que se les prestará mayor atención.

### Métodos de reasignación

Los métodos de cluster de reasignación son métodos iterativos de agrupación que tienen como objetivo encontrar una partición de un conjunto de $n$ elementos en $m$ grupos. Reciben el nombre de métodos de reasignación, para destacar la posibilidad de que un elemento puede ser asignado varias veces a distintos clusters durante el proceso de clasificación. Los métodos de reasignación realizan sucesivas iteraciones en las que se cambia la configuración de los distintos grupos con el fin de encontrar la distribución

óptima para el criterio escogido. El papel del centroide es fundamental en este paso y será definido más adelante. El proceso finaliza cuando la métrica elegida converge y no se realizan más reasignaciones.

**Definición 1.4.15.** *Se denomina **centroide** de un cluster al punto que se sitúa equidistante con respecto a todos los elementos de dicho cluster.*

En este tipo de métodos se pueden diferenciar tres etapas:

1. Inicialización: Hay distintas variantes, algunos métodos comienzan con una partición aleatoria y otros establecen de manera aleatoria los centroides.
2. Asignación: Se asigna cada observación al grupo con el centroide más cercano a partir de las distancias entre ambos.
3. Actualización: Se recalcula el valor del centroide del cluster. Puede ser después de cada asignación o después de todas las asignaciones del ciclo completo.

El método K-Medias busca encontrar una partición en grupos tomando como centroide de cada grupo la media de los elementos del cluster. Los algoritmos propuestos por McQueen [13], o por Forgy [5] son algunos de los más utilizados. Ambos métodos tienen como objetivo minimizar la suma de errores cuadráticos (SSE). La función se define como:

$$SSE = \sum_{j=1}^{m} \sum_{i=1}^{n} \|x_i^j - c_j\|^2 \tag{1.69}$$

Donde $x_i^j$ el elemento i-ésimo del cluster j-ésimo y $c_j$ el centroide del cluster j-ésimo, con $i = \{1, ..., n\}$ y $j = \{1, ..., m\}$; siendo $\|x_i^j - c_j\|^2$ la distancia euclídea entre cada uno de los elementos del cluster y su centroide.

El algoritmo de McQueen [13], detallado en el diagrama de la figura B.1, se caracteriza por realizar el recálculo de los centroides después de cada reasignación de cada uno de los elementos. El algoritmo de Forgy [5], explicado en el diagrama de la figura B.2, se caracteriza por realizar el recálculo de los centroides después de todas las reasignaciones del ciclo. También se conocen como métodos de los centroides por la importacia de este concepto en el flujo de decisión.

# Capítulo 2

# Fundamentos matemáticos del análisis funcional

El concepto de datos funcionales aparece para describir procesos que de manera habitual se comportan como una curva que cambia de manera suave y continua en el tiempo. El Análisis de Datos Funcionales (FDA) amplía los horizontes del análisis tradicional permitiendo el estudio estadístico para observaciones de un proceso funcional. Utilizar este tipo de datos requiere adaptar las herramientas existentes y crear algunas nuevas para rentabilizar todas las opciones que ofrece este cambio de perspectiva.

Como ya se mencionó en la introducción del trabajo, aunque las observaciones pertenezcan a un proceso de esencia funcional, en la práctica solo se dispone de una muestra discreta. Para dotar de ese nuevo conocimiento al estudio, lo primero que se debe preparar es la recuperación de la naturaleza funcional de los datos. En esta sección se desarrollan tanto las razones como las herramientas para llevar a cabo este proceso.

## 2.1. Conceptos básicos

**Definición 2.1.1.** *Un **dato funcional** es un conjunto de mediciones a lo largo de un continuo, que consideradas como conjunto deben tratarse como una sola entidad, curva o imagen. Es decir, son datos en los que las observaciones coinciden con los valores de una funcón real $\boldsymbol{x}(t)$, $t \in T$, con $T$ un intervalo real continuo que representa el tiempo.*

Un conjunto de datos funcionales es una colección de curvas $\{\boldsymbol{x_1}(t), ..., \boldsymbol{x_n}(t)\}$, con $t \in T$. Las observaciones de cada una no tienen por qué ser homogéneas y cada una de ellas puede haber sido observada para diferentes valores de tiempo $t$, como por ejemplo, $\boldsymbol{x_i}(t) = (x_i(t_{i1}), ..., x_i(t_{im_i})) := (x_{i1}, ..., x_{im_i})$ para $\boldsymbol{t} = \{t_{i1}, ..., t_{im_i}\}$ donde $i \in \{1, ..., n\}$

y siendo $m_i$ el número de observaciones de cada curva -que además de poder diferir en número también pueden diferir en distribución-.

Por ejemplo, en una base de datos recogidos en distintas estaciones meteorológicas puede ocurrir que para la estación meteorológica del aeropuerto de Madrid se guarden diariamente las mediciones y, sin embargo, en la estación de Valdepeñas, solo se almacenen las mediciones producidas los domingos.

Si se supone que las $m$ observaciones son idénticas para cada curva $\{\boldsymbol{x_1}(t), ..., \boldsymbol{x_n}(t)\}$, la información disponible en esta situación puede almacenarse en la matriz $\boldsymbol{X}_{n\times m}$:

$$\boldsymbol{X}_{n\times m} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix} \tag{2.1}$$

El **Análisis de Datos Funcionales** se ocupa del estudio de los datos funcionales. Introduce varias nociones novedosas frente al análisis estadístico clásico:

1. Conceptualmente los datos funcionales están definidos de manera continua aunque en la práctica las observaciones sean discretas o se almacenen como tal. Uno de los primeros pasos que se debe adoptar al trabajar con datos funcionales es la transformación del conjunto de observaciones discretas a forma funcional.

2. El conjunto de datos completo se extrae de la función, en vez de cada observación por separado.

3. En muchos casos se utiliza como variable independiente el tiempo por ser una variable continua.

**Definición 2.1.2.** *Se denomina espacio de funciones de cuadrado integrable al espacio definido por:*

$$L^2 = \left\{ f : T \longrightarrow T : \int_T \boldsymbol{f}^2(t)dt < \infty \right\} \tag{2.2}$$

*Siendo $T$ el espacio continuo en el que se desarrollan los procesos dependientes del tiempo.*

**Nota 2.1.1.** *De ahora en adelante la integral definida $\int_T \boldsymbol{x}(t)dt$ se abreviará como $\int \boldsymbol{x}(t)dt$, siempre y cuando estén claros tantos los límites de integración como la variable sobre la que tiene lugar la integración.*

**Definición 2.1.3.** *Sea $L^2$ el espacio de funciones de cuadrado integrable, la aplicación*

*del* ***producto escalar*** *euclidiano en* $L^2$ *viene definida como:*

$$\begin{array}{rccl} \langle \cdot, \cdot \rangle \colon & L^2 \times L^2 & \longrightarrow & \mathbb{R} \\ & (\boldsymbol{x}, \boldsymbol{y}) & \longmapsto & \langle \boldsymbol{x}, \boldsymbol{y} \rangle = \int \boldsymbol{x}(t)\boldsymbol{y}(t)dt \end{array} \tag{2.3}$$

**Observación 2.1.1.** *El producto escalar euclidiano verifica las siguientes propiedades:*

- *Simetría:*

$$\langle \boldsymbol{e_1}, \boldsymbol{e_2} \rangle = \int \boldsymbol{e_1}(t)\boldsymbol{e_2}(t)dt = \int \boldsymbol{e_2}(t)\boldsymbol{e_1}(t)dt = \langle \boldsymbol{e_2}, \boldsymbol{e_1} \rangle \quad \forall \boldsymbol{e_1}, \boldsymbol{e_2} \in L^2 \tag{2.4}$$

- *Positividad:*

$$\begin{gathered} \langle \boldsymbol{e_1}, \boldsymbol{e_1} \rangle = \int \boldsymbol{e_1}(t)\boldsymbol{e_1}(t)dt \geq 0, \quad \forall \boldsymbol{e_1} \in L^2 \\ con \quad \langle \boldsymbol{e_1}, \boldsymbol{e_1} \rangle = \int \boldsymbol{e_1}(t)\boldsymbol{e_1}(t)dt = 0 \Leftrightarrow \boldsymbol{e_1} = 0 \end{gathered} \tag{2.5}$$

- *Bilinealidad:* $\forall a, b \in \mathbb{R}$

$$\begin{gathered} \langle a\boldsymbol{e_1} + b\boldsymbol{e_2}, \boldsymbol{e_3} \rangle = \int [a\boldsymbol{e_1}(t) + b\boldsymbol{e_2}(t)]\boldsymbol{e_3}(t)dt = \int [a\boldsymbol{e_1}(t)\boldsymbol{e_3}(t) + b\boldsymbol{e_2}(t)\boldsymbol{e_3}(t)]dt = \\ \int a\boldsymbol{e_1}(t)\boldsymbol{e_3}(t)dt + \int b\boldsymbol{e_2}(t)\boldsymbol{e_3}(t)dt = a\langle \boldsymbol{e_1}, \boldsymbol{e_3} \rangle + b\langle \boldsymbol{e_2}, \boldsymbol{e_3} \rangle \\ \forall \boldsymbol{e_1}, \boldsymbol{e_2}, \boldsymbol{e_3} \in L^2 \end{gathered} \tag{2.6}$$

**Observación 2.1.2.** *El espacio de funciones de cuadrado integrable* $L^2$ *es un espacio de* ***Hilbert****.*

**Definición 2.1.4.** *La* ***norma*** *de un elemento* $\boldsymbol{e_1}(t)$ *en el espacio* $L^2$ *se define como:*

$$\|\boldsymbol{e_1}(t)\| = \sqrt{\langle \boldsymbol{e_1}, \boldsymbol{e_1} \rangle} = \sqrt{\int \boldsymbol{e_1}^2(t)dt} \tag{2.7}$$

*Y sus propiedades son:*

1. $\|\boldsymbol{e_1}(t)\| \geq 0$ *y* $\|\boldsymbol{e_1}(t)\| = 0 \Leftrightarrow \boldsymbol{e_1}(t) = 0$
2. $\|a\boldsymbol{e_1}(t)\| = |a|\|\boldsymbol{e_1}(t)\| \ \forall a \in \mathbb{R}$
3. $\|\boldsymbol{e_1}(t) + \boldsymbol{e_2}(t)\| \leq \|\boldsymbol{e_1}(t)\| + \|\boldsymbol{e_2}(t)\|$
4. $|\langle \boldsymbol{e_1}, \boldsymbol{e_2} \rangle| \leq \|\boldsymbol{e_1}(t)\| \cdot \|\boldsymbol{e_2}(t)\| = \sqrt{\langle \boldsymbol{e_1}, \boldsymbol{e_1} \rangle \cdot \langle \boldsymbol{e_2}, \boldsymbol{e_2} \rangle}$ *(desigualdad de Cauchy-Schwartz)*

Al igual que para el espacio euclídeo también se puede deducir la definición de ángulo entre funciones a partir de la desigualdad de Cauchy-Schwartz.

**Definición 2.1.5.** *De la desigualdad de Cauchy-Schwartz se puede deducir la* ***desigualdad del coseno****:*

$$-1 \leq \frac{\langle \boldsymbol{e_1}, \boldsymbol{e_2} \rangle}{\|\boldsymbol{e_1}(t)\| \cdot \|\boldsymbol{e_2}(t)\|} \leq 1 \quad \forall \boldsymbol{e_1}, \boldsymbol{e_2} \in L^2 \tag{2.8}$$

Y de esta a su vez se puede extraer el concepto de **ángulo** entre dos funciones calculando el arccoseno de la expresión anterior:

$$\theta = arcos \frac{\langle \boldsymbol{e_1}, \boldsymbol{e_2} \rangle}{\|\boldsymbol{e_1}(t)\| \|\boldsymbol{e_2}(t)\|} = \frac{\int \boldsymbol{e_1}(t)\boldsymbol{e_2}(t)dt}{\sqrt{\int \boldsymbol{e_1}^2(t)dt \int \boldsymbol{e_2}^2(t)dt}} \quad \forall \boldsymbol{e_1}, \boldsymbol{e_2} \in L^2 \tag{2.9}$$

**Definición 2.1.6.** *La noción de* ***ortogonalidad*** *entre funciones viene dada por el producto escalar. La función* $\boldsymbol{e_1}(t)$ *es ortogonal a* $\boldsymbol{e_2}(t)$ *si:*

$$\langle \boldsymbol{e_1}, \boldsymbol{e_2} \rangle = \int \boldsymbol{e_1}(t)\boldsymbol{e_2}(t)dt = 0 \quad \forall \boldsymbol{e_1}, \boldsymbol{e_2} \in L^2 \tag{2.10}$$

## 2.2. Representación en términos de funciones de una base

Uno de los objetivos perseguidos es encontrar un sistema de representación universal para homogeneizar los datos, ya que puede ocurrir que las funciones no estén evaluadas para las mismas franjas de tiempo. También se busca tener acceso a la información entre observaciones sin que eso conlleve aumentar el tamaño de la muestra volviéndola inmanejable. Indudablemente, en la solución interviene la aproximación mediante funciones de una base, aunque siempre intentando utilizar solo algunos de sus componentes. Si se dispone de $m$ observaciones en los tiempos $\boldsymbol{t} = \{t_1, ..., t_{m_i}\}$ para la realización $\boldsymbol{x_i}(t)$, el flujo de pasos consiste en elegir la base, para posteriormente construir la expresión de $k$ términos de la base basado en las observaciones. Esto es, las curvas pertenecen a un espacio de dimensión finita generado por la base de funciones $\{\boldsymbol{\phi_1}(t), ..., \boldsymbol{\phi_k}(t)\}$ de manera que pueden expresarse como:

$$\boldsymbol{x_i}(t) = \sum_{h=1}^{k} b_{ih} \boldsymbol{\phi_h}(t) \tag{2.11}$$

Siendo $k \in \mathbb{N}$, $b_{ih} \in \mathbb{R}$ los coeficientes de la combinación lineal, $\boldsymbol{\phi_h}$ las funciones base y con $h = \{1, .., k\}$ e $i = \{1, .., n\}$.

Si se supone que de las observaciones discretas de la curva $\boldsymbol{x_i}(t)$ en los tiempos $\boldsymbol{t} = \{t_1, ..., t_{m_i}\}$ se obtienen los valores $(x_{i1}, ..., x_{im_i})$ para cada $i = \{1, .., n\}$, el objetivo siguiente será establecer la relación existente entre cada una de las observaciones con la función $\boldsymbol{x_i}(t)$. Una vez conocida, se podrá calcular el valor para cualquier valor de $t \in T$. Normalmente se considera que cada una de las curvas pertenece a un subespacio de funciones generadas por una base. De la misma manera que un vector de un espacio vectorial puede expresarse de forma única como combinación lineal de los vectores de una base, se define el concepto de base de funciones para expresar las curvas a partir de las mismas.

Por ejemplo, en la figura 2.1 se puede ver la comparación de cómo se representa una muestra de datos de temperaturas recogidas para diferentes estaciones meteorológicas españolas antes y después de aproximarla mediante bases de funciones.

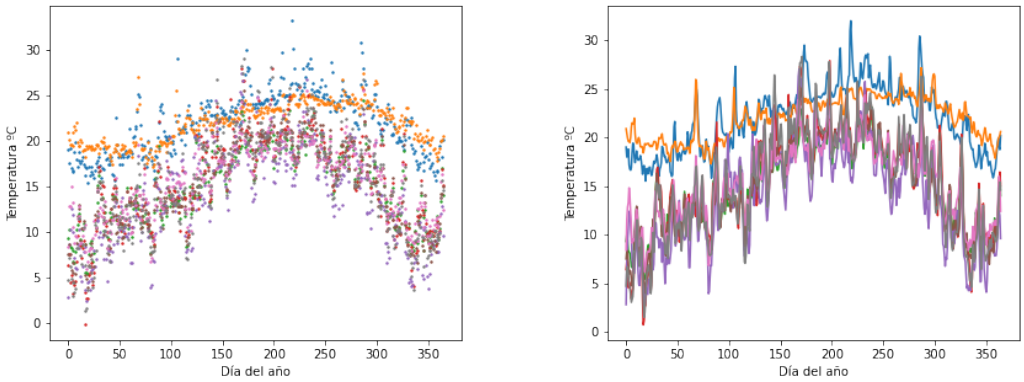



Figura 2.1: A la izquierda se puede observar una representación de una muestra discreta de datos, mientras que a la derecha se representa la misma muestra de datos funcionales. Fuente: elaboración propia

### 2.2.1. Bases de funciones

Un sistema de bases de funciones es un sistema de funciones conocidas, independientes, capaces de aproximar cualquier función mediante la combinación lineal de un número $k$ suficiente de todas ellas.

**Definición 2.2.1.** *Sea $H$ un espacio métrico de funciones, una **base** de $H$ es un sistema numerable de funciones $\{\boldsymbol{\phi_h}\}_{h=1}^{\infty} \in H$ verificando que cualquier elemento de $H$ puede expresarse como combinación lineal de sus componentes.*

Por tanto, como se ha mencionado anteriormente, en el transcurso de devolver la naturaleza funcional a las observaciones discretas, se identifican claramente dos fases:

1. Elección de la base.
2. Cálculo de los coeficientes de la combinación lineal en términos de la base escogida.

Para la primera, se espera que la base se ajuste al comportamiento original de los datos y se entiende como elección acertada si logra una buena aproximación con un número mínimo ($k \ll \infty$) de funciones base. A continuación se muestran algunas de las bases de funciones más comunes.

**Colección de monomiales:** El sistema de monomiales se utiliza para construir series potenciales. Una serie potencial es básicamente un polinomio de grado infinito que representa alguna función. La serie potencial viene definida como:

$$\sum_{h=0}^{\infty} a_h x^h \tag{2.12}$$

Donde $a_h \in \mathbb{R}$ son los coeficientes de la combinación lineal del conjunto de monomiales $\left\{1, x, ..., x^h, ...\right\}$.

**Sistema de bases de Fourier:** Una de las mejores aproximaciones para datos periódicos la proporcionan las series de Fourier. Sus términos quedan determinados por:

$$\left.\begin{array}{rcl} \boldsymbol{\phi_0}(t) & = & 1 \\ \boldsymbol{\phi_{2r-1}}(t) & = & sen(r\omega t) \\ \boldsymbol{\phi_{2r}}(t) & = & cos(r\omega t) \end{array}\right\} \quad t \in T \tag{2.13}$$

Siendo $\omega$ el parámetro que determina el período $\dfrac{2\pi}{\omega}$ de la base periódica. Es la base más apropiada para funciones con un comportamiento periódico definidos sobre un intervalo $T$ y con cambios ligeros en su curvatura.

**Sistema de bases de Splines:** Las funciones de splines es la elección más apropiada para aproximar datos que no tienen un comportamiento periódico. Se detallan primero las funciones splines, para después presentar el sistema de bases de splines.

Los splines son polinomios definidos sobre subintervalos del dominio de observación y están determinados por un conjunto de puntos de unión (nodos) y por el orden[1]. Es decir, una función spline definida a partir de $L+1$ nodos; $\boldsymbol{t} = \{t_0, t_1, ..., t_L\}$ con $t \in T$ y fijado $m \geq 0$, se dice que es de orden $m$ si verifica:

- En cada uno de los $L$ subintervalos determinados por los nodos $[t_{i-1}, t_i)$, la función es un segmento poligonal con grado menor o igual a $m$, con $i = \{1, ..., L\}$.

[1]Número de constantes necesarias para definir un polinomio. Se puede calcular como el grado del polinomio más uno

- La función spline es una curva diferenciable y tiene una derivada de orden $(m-1)$ continua en el intervalo definido por $[t_0, t_L]$.

Para cada uno de los subintervalos los splines presentan un grado fijo. Los nodos funcionan como fronteras y para splines de orden mayor a 1, se calculan con la condición de que el valor en dichos puntos sea el mismo tanto en la aproximación derecha como la izquierda. Para splines de orden mayor a 2 la unión será suave por la naturaleza de los polinomios. Para elegir una base de funciones para aproximar los datos, el número $k$ de funciones base necesarias viene determinado por la suma del orden y el número de nodos internos. Uno de los sistemas de splines más utilizados es el de base de B-splines.

En el sistema de bases de B-splines, cada función base $\boldsymbol{\phi_h}(t)$, $h = \{1, ..., k, ...\}$; está determinada por una función spline definida por un orden $m$ y un conjunto de nodos $L$. La base de B-splines cobra interés en datos no periódicos con tramos diferenciables localizados. Además, permite calcular rápidamente los coeficientes de la combinación lineal y tan solo requiere un número $k$ de funciones relativamente pequeño para conseguir una buena aproximación.

La elección del tipo de base es determinante para el análisis, una vez elegida es momento de abordar la segunda fase: construir la expresión de aproximación de la curva. Para ello, habrá que calcular los coeficientes de la combinación. A continuación, en la figura 2.2 se muestra la diferencia de utilizar una base con siete funciones y una base con veinte.

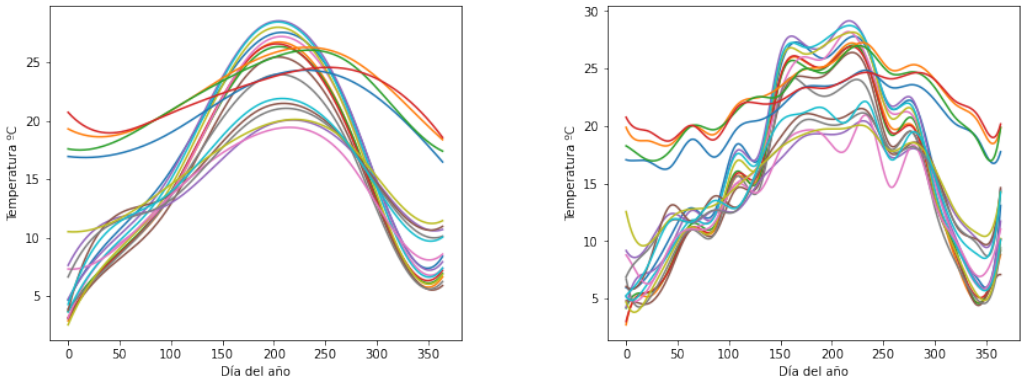



Figura 2.2: Representación de los datos de temperaturas medias mediante bases de B-Splines. A la izquierda truncada a siete funciones y a la derecha truncada a veinte.

### 2.2.2. Métodos de ajuste y cálculo de coeficientes

El cálculo de los coeficientes dependerá de la exactitud de las $n$ observaciones $\{\boldsymbol{x_1}(t), ..., \boldsymbol{x_n}(t)\}$:

- Si se suponen sin error:
$$x_{im_i} = \boldsymbol{x_i}(t_{im_i}) \tag{2.14}$$
siendo $x_{im_i}$ la observación de la curva $\boldsymbol{x_i}(t)$ en el instante de tiempo $t_{im_i} \in T$, donde $i = \{1, ..., n\}$. El método numérico para estimarlos será una interpolación.

- Si se suponen con error:
$$x_{im_i} = \boldsymbol{x_i}(t_{im_i}) + \varepsilon_{im_i} \tag{2.15}$$
siendo $x_{im_i}$ la observación de la curva $\boldsymbol{x_i}(t)$ en el instante de tiempo $t_{im_i} \in T$ y $\varepsilon_{im_i}$ el error asociado a dicha medición, para $i = \{1, ..., n\}$. En este será una técnica de suavizado, como por ejemplo el método de mínimos cuadrados.

Se tienen observaciones discretas con error de la forma $x_{ij} = \boldsymbol{x_i}(t_{ij}) + \varepsilon_{ij}$ con $i = \{1, ..., n\}$, $j = \{1, ..., m_i\}$. Sea $\{\boldsymbol{\phi_h}\}_{h=1}^{k}$ el conjunto finito de funciones base utilizado para aproximar el valor de $\boldsymbol{x_i}(t)$, se obtiene:

$$\boldsymbol{x_i}(t) = \sum_{h=1}^{k} b_{ih}\boldsymbol{\phi_h}(t) = \boldsymbol{b_i'}\boldsymbol{\Phi} \tag{2.16}$$

con $\boldsymbol{b_i} = (b_{i1}, ..., b_{ik})'$ los coeficientes pendientes de la aproximación de la curva $\boldsymbol{x_i}(t)$ $\forall i = \{1, ..., n\}$. Una de las opciones es recurrir al método de mínimos cuadrados. Se define la matriz $\boldsymbol{\Phi} = (\boldsymbol{\phi_1}(t_{ij}), ..., \boldsymbol{\phi_k}(t_{ij}))$ con $j = \{1, ..., m_i\}$ representa el valor de las funciones base para la curva $\boldsymbol{x_i}(t)$ y tiene dimensión $m_i \times k$.

$$\boldsymbol{\Phi} = \begin{pmatrix} \boldsymbol{\phi_1}(t_{i1}) & \boldsymbol{\phi_2}(t_{i1}) & \cdots & \boldsymbol{\phi_k}(t_{i1}) \\ \boldsymbol{\phi_1}(t_{i2}) & \boldsymbol{\phi_2}(t_{i2}) & & \vdots \\ \vdots & & \ddots & \vdots \\ \boldsymbol{\phi_1}(t_{im_i}) & & & \boldsymbol{\phi_k}(t_{im_i}) \end{pmatrix} \tag{2.17}$$

Minimizar el error por mínimos cuadrados implica minimizar la siguiente expresión o su forma vectorial para cada una de las curvas $\{\boldsymbol{x_1}(t), ..., \boldsymbol{x_n}(t)\}$:

$$\sum_{j=1}^{m_i} \left( x_{ij} - \sum_{h=1}^{k} b_{ih}\boldsymbol{\phi_h}(t_{ij}) \right)^2 \tag{2.18}$$

Si se presupone que los errores $\varepsilon_{ij}$ para cada curva $\boldsymbol{x_i}(t)$, para $i = \{1, ..., n\}$ son independientes con una distribución normal de media nula y varianza $\sigma^2$, $N(0, \sigma^2)$, el estimador que da solución al problema es:

$$\hat{\boldsymbol{b}}_i = (\boldsymbol{\Phi'\Phi})^{-1}\boldsymbol{\Phi' x}_i \tag{2.19}$$

Ahora, es inmediato calcular los valores estimados que alcanza la curva $\boldsymbol{x_i}(t)$:

$$\hat{\boldsymbol{x}}_{\boldsymbol{i}} = \boldsymbol{\Phi}\hat{\boldsymbol{b}}_i = \boldsymbol{\Phi}(\boldsymbol{\Phi'\Phi})^{-1}\boldsymbol{\Phi' x_i} \tag{2.20}$$

# Capítulo 3

# Técnicas multivariantes en el contexto del análisis de datos funcionales

Las técnicas funcionales han supuesto desde sus inicios una auténtica revolución al ser capaces de contextualizar la muestra de datos discretos en un marco variante en el tiempo. Admiten trabajar con la naturaleza funcional de los datos aportando ese conocimiento al análisis.

En el capítulo anterior se ha detallado cómo recuperar el carácter continuo de los procesos y en la redacción de este, se mostrará cómo abordar el estudio utilizando técnicas del capítulo 1, aunque adaptadas para ejecutar sus procesos en un espacio de dimensión infinita.

Para ello, en un principio, se definen los estadísticos descriptivos en el contexto del análisis de datos funcionales, puesto que al aplicarse sobre un espacio $L^2$ también sufren transformaciones con respecto a la estadística clásica y posteriormente se plantean técnicas de análisis multivariantes adaptadas a este contexto.

## 3.1. Estadísticos descriptivos en el contexto de datos funcionales

Medidas como las de dispersión o variabilidad utilizadas sobre variables aleatorias se pueden adaptar sobre muestras de funciones aleatorias, suponiendo que el espacio de trabajo es $L^2$.

La sección se estructura en tres apartados en los que se recogen las distintas aproximaciones que ofrece un problema funcional según cuál sea el objeto de estudio.

### 3.1.1. Estadísticos sobre una función

Sea $t \in T$, se denotará la función $\mathbf{1}(t) := \mathbf{1} \quad \forall t \in T$.

**Definición 3.1.1.** *El **valor medio** de la función $\boldsymbol{x}(t)$ se define como:*

$$\bar{x} = \frac{1}{T}\langle \boldsymbol{x}, \mathbf{1}\rangle = \frac{1}{\int \mathbf{1}(t)^2 dt} \cdot \int_0^T \boldsymbol{x}(t)\mathbf{1}(t)dt \tag{3.1}$$

El valor medio de la función $\boldsymbol{x}(t)$ es un estadístico que describe su tendencia central.

**Definición 3.1.2.** *La función **valor medio** de $\boldsymbol{x}(t)$, $\boldsymbol{f}_{\bar{\boldsymbol{x}}}(t)$, está definida $\forall t \in T$ por el valor medio $\bar{x}$ de la función $\boldsymbol{x}(t)$. Se puede denotar como $\bar{x} \cdot \mathbf{1}(t)$.*

**Observación 3.1.1.** *Es importante no confundir los conceptos de media de una función y la función valor medio.*

**Definición 3.1.3.** *La **varianza** de $\boldsymbol{x}(t)$ se define como:*

$$S^2_{\boldsymbol{x}(t)} = \frac{1}{T}\langle \boldsymbol{x} - \bar{x}\cdot\mathbf{1}, \boldsymbol{x} - \bar{x}\cdot\mathbf{1}\rangle = \frac{1}{\int \mathbf{1}(t)^2 dt} \cdot \int \left(\boldsymbol{x}(t) - \bar{x}\cdot\mathbf{1}(t)\right)^2 dt \tag{3.2}$$

Representa la variación media al cuadrado de todos los valores de la función con respecto al valor medio.

**Definición 3.1.4.** *La **covarianza** de dos funciones $\boldsymbol{x}(t)$ e $\boldsymbol{y}(t)$ se define como:*

$$S_{\boldsymbol{x}(t),\boldsymbol{y}(t)} = \frac{1}{T}\langle \boldsymbol{x}-\bar{x}\cdot\mathbf{1}, \boldsymbol{y}-\bar{y}\cdot\mathbf{1}\rangle = \frac{1}{\int \mathbf{1}(t)^2 dt}\cdot\int \left(\boldsymbol{x}(t) - \bar{x}\cdot\mathbf{1}(t)\right)\left(\boldsymbol{y}(t) - \bar{y}\cdot\mathbf{1}(t)\right) dt \tag{3.3}$$

**Definición 3.1.5.** *La **correlación** entre dos funciones $\boldsymbol{x}(t)$ e $\boldsymbol{y}(t)$ se define como:*

$$\begin{aligned} r_{\boldsymbol{x}(t),\boldsymbol{y}(t)} &= \frac{S_{\boldsymbol{x}(t),\boldsymbol{y}(t)}}{S_{\boldsymbol{x}(t)}S_{\boldsymbol{y}(t)}} = \frac{\frac{1}{T}\langle \boldsymbol{x} - \bar{x}\cdot\mathbf{1}, \boldsymbol{y} - \bar{y}\cdot\mathbf{1}\rangle}{\frac{1}{T}\langle \boldsymbol{x} - \bar{x}\cdot\mathbf{1}, \boldsymbol{x} - \bar{x}\cdot\mathbf{1}\rangle\frac{1}{T}\langle \boldsymbol{y} - \bar{y}\cdot\mathbf{1}, \boldsymbol{y} - \bar{y}\cdot\mathbf{1}\rangle} \\ &= \frac{\frac{1}{\int \mathbf{1}(t)^2 dt}\cdot\int \left(\boldsymbol{x}(t) - \bar{x}\cdot\mathbf{1}(t)\right)\left(\boldsymbol{y}(t) - \bar{y}\cdot\mathbf{1}(t)\right) dt}{\frac{1}{\int \mathbf{1}(t)^2 dt}\cdot\int \left(\boldsymbol{x}(t) - \bar{x}\cdot\mathbf{1}(t)\right)^2 dt\,\frac{1}{\int \mathbf{1}(t)^2 dt}\cdot\int \left(\boldsymbol{y}(t) - \bar{y}\cdot\mathbf{1}(t)\right)^2 dt} \end{aligned} \tag{3.4}$$

### 3.1.2. Estadísticos sobre una función aleatoria

**Definición 3.1.6.** *Dada una muestra de n funciones* $\boldsymbol{x_1}(t), \boldsymbol{x_2}(t), ..., \boldsymbol{x_n}(t)$ *de una función aleatoria* $\boldsymbol{x}(t)$*, con* $t \in T$*. La función* ***media muestral*** *de* $\boldsymbol{x}(t)$ *viene definida como:*

$$\bar{\boldsymbol{x}}(t) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x_i}(t) \tag{3.5}$$

**Definición 3.1.7.** *Dada una muestra de n funciones* $\boldsymbol{x_1}(t), \boldsymbol{x_2}(t), ..., \boldsymbol{x_n}(t)$ *de una función aleatoria* $\boldsymbol{x}(t)$*, con* $t \in T$*. La función* ***varianza muestral*** *de* $\boldsymbol{x}(t)$ *viene definida como:*

$$Var_{\boldsymbol{x(t)}}(t) = \frac{1}{n-1}\sum_{i=1}^{n} \left(\boldsymbol{x_i}(t) - \bar{\boldsymbol{x}}(t)\right)^2 \tag{3.6}$$

*Y la función* ***desviación estándar muestral*** *de* $\boldsymbol{x}(t)$ *como:*

$$Stdev_{\boldsymbol{x}(t)}(t) = \sqrt{Var_{\boldsymbol{x}(t)}(t)} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n} \left(\boldsymbol{x_i}(t) - \bar{\boldsymbol{x}}(t)\right)^2} \tag{3.7}$$

**Definición 3.1.8.** *Dada una muestra de funciones* $\boldsymbol{x_1}(t), \boldsymbol{x_2}(t), ..., \boldsymbol{x_n}(t)$ *de una función aleatoria* $\boldsymbol{x}(t)$*, con* $t \in T$*. La función* ***covarianza muestral*** *de* $\boldsymbol{x}(t)$ *entre dos tiempos* $t_1$ *y* $t_2$ *se define como:*

$$Cov_{\boldsymbol{x}(t)}(t_1, t_2) = \frac{1}{n-1}\sum_{i=1}^{n} \left(\boldsymbol{x_i}(t_1) - \bar{\boldsymbol{x}}(t_1)\right)\left(\boldsymbol{x_i}(t_2) - \bar{\boldsymbol{x}}(t_2)\right), \quad t_1, t_2 \in T \tag{3.8}$$

**Definición 3.1.9.** *Dada una muestra de funciones* $\boldsymbol{x_1}(t), \boldsymbol{x_2}(t), ..., \boldsymbol{x_n}(t)$ *de una función aleatoria* $\boldsymbol{x}(t)$*, con* $t \in T$*. La función* ***correlación muestral*** *de* $\boldsymbol{x}(t)$ *entre dos tiempos* $t_1$ *y* $t_2$ *viene definida como:*

$$Corr_{\boldsymbol{x}(t)}(t_1, t_2) = \frac{Cov_{\boldsymbol{x}(t)}(t_1, t_2)}{\sqrt{Var_{\boldsymbol{x}(t)}(t_1) \cdot Var_{\boldsymbol{x}(t)}(t_2)}}, \quad t_1, t_2 \in T \tag{3.9}$$

### 3.1.3. Estadísticos de muestras de dos o más funciones aleatorias

Son los encargados de estudiar la relación entre dos o más muestras de funciones aleatorias diferentes.

**Definición 3.1.10.** *Dada una muestra de n funciones* $\boldsymbol{x}_1(t), \boldsymbol{x}_2(t), ..., \boldsymbol{x}_n(t)$ *de una función aleatoria* $\boldsymbol{x}(t)$*, con* $t \in T$ *y una muestra de funciones*

$\boldsymbol{y_1}(t), \boldsymbol{y_2}(t), ..., \boldsymbol{y_n}(t)$ *de una función aleatoria* $\boldsymbol{y}(t)$*, con* $t \in T$*. La función* ***covarianza cruzada*** *de* $\boldsymbol{x}(t)$ *e* $\boldsymbol{y}(t)$ *entre* $t_1$ *y* $t_2$ *viene definida como:*

$$Cov_{\boldsymbol{x}(t),\boldsymbol{y}(t)}(t_1, t_2) = \frac{1}{n-1} \sum_{i=1}^{n} \left(\boldsymbol{x_i}(t_1) - \bar{\boldsymbol{x}}(t_1)\right) \left(\boldsymbol{y_i}(t_2) - \bar{\boldsymbol{y}}(t_2)\right), \quad t_1, t_2 \in T \tag{3.10}$$

**Definición 3.1.11.** *Dada una muestra de n funciones* $\boldsymbol{x_1}(t), \boldsymbol{x_2}(t), ..., \boldsymbol{x_n}(t)$ *de una función aleatoria* $\boldsymbol{x}(t)$*, con* $t \in T$ *y una muestra de n funciones* $\boldsymbol{y_1}(t), \boldsymbol{y_2}(t), ..., \boldsymbol{y_n}(t)$ *de una función aleatoria* $\boldsymbol{y}(t)$ *con* $t \in T$*. La función* ***correlación cruzada*** *de* $\boldsymbol{x}(t)$ *e* $\boldsymbol{y}(t)$ *entre* $t_1$ *y* $t_2$ *viene definida como:*

$$Corr_{\boldsymbol{x}(t),\boldsymbol{y}(t)}(t_1, t_2) = \frac{Cov_{\boldsymbol{x}(t),\boldsymbol{y}(t)}(t_1, t_2)}{\sqrt{Var_{\boldsymbol{x}(t)}(t_1) \cdot Var_{\boldsymbol{y}(t)}(t_2)}}, \quad t_1, t_2 \in T \tag{3.11}$$

## 3.2. Modelos de regresión para datos funcionales

En esta sección se presentan los fundamentos básicos de la técnica de regresión lineal para datos funcionales. Se ofrece la transformación necesaria de los métodos para el análisis en un espacio de dimensión infinita y se explican las distintas posibilidades que surgen al ampliar la regresión al ámbito funcional. Una de las grandes diferencias existentes entre el modelo de regresión clásico y funcional es que los coeficientes de la regresión se convierten en funciones. Las aportaciones más relevantes de este análisis llegan con Ramsay y Silverman, [18]. En esta sección se explicarán dos de los escenarios que recogen los modelos de regresión: respuesta escalar y variable independiente funcional, y respuesta funcional y variable independiente funcional.

### 3.2.1. Regresión con respuesta escalar

En muchos estudios surge la necesidad de analizar la relación que guarda una variable escalar con respecto a otra funcional. Por ejemplo, ante las continuas subidas de la luz coincidentes con la tormenta Filomena y la llegada del calor, cabe preguntarse si el coste de la luz para un período determinado depende de las temperaturas que ha habido durante ese mismo período. En este caso, el coste en el intervalo de tiempo de estudio tiene un valor escalar, mientras que las temperaturas a lo largo de dicho período responden a un comportamiento funcional.

Para el desarrollo del modelo se consideran $n$ observaciones y se toma como hipótesis que la variable independiente es funcional $\boldsymbol{x_i}(t) \in L^2$ con $i = \{1, ..., n\}$ y $t \in T$, y la variable respuesta $y_i \in \mathbb{R}$.

El ejemplo descrito anteriormente responde al siguiente modelo:

$$y_i = \alpha + \int \boldsymbol{x_i}(t)\boldsymbol{\beta}(t) + \varepsilon, \quad t \in T \tag{3.12}$$

Donde $\alpha$ es el término independiente, $\boldsymbol{\beta}(t) \in L^2$ los coeficientes de la regresión y $\varepsilon$ el término del error. Luego, determinar el modelo se reduce a encontrar la mejor estimación para los parámetros $\boldsymbol{\beta}(t)$ y $\alpha$.

Puesto que tanto $\boldsymbol{x_i}(t)$ para $i = \{1, ..., n\}$, como $\boldsymbol{\beta}(t)$ son funciones de $L^2$, ambas se aproximan mediante elementos de una base de funciones, aunque no tiene por qué ser mediante la misma:

$$\boldsymbol{\beta}(t) = \sum_{h=1}^{m_\beta} b_h \boldsymbol{\phi_h}(t) \Rightarrow \boldsymbol{\beta} = \boldsymbol{\phi' b}, \quad t \in T \tag{3.13}$$

Siendo $m_\beta$ el número de elementos de la base de funciones $\boldsymbol{\phi} = \{\boldsymbol{\phi_h}\}_{h=1}^{m_\beta}$ necesarios para conseguir un buen ajuste y donde $\boldsymbol{b} = b_1, ..., b_{m_\beta}$ son los coeficientes de la combinación lineal.

De forma análoga se expresa la aproximación de $\boldsymbol{x_i}(t)$ para $i = \{1, ..., n\}$, mediante funciones de una base:

$$\boldsymbol{x_i}(t) = \sum_{j=1}^{m_x} c_{ij} \boldsymbol{\varphi_j}(t) \Rightarrow \boldsymbol{x_i} = \boldsymbol{C_{n \times m_x} \varphi}, \quad t \in T \tag{3.14}$$

Siendo $m_x$ el número de elementos de la base de funciones $\boldsymbol{\varphi} = \{\boldsymbol{\varphi_h}\}_{h=1}^{m_x}$ necesarios para conseguir un buen ajuste y facilitar la representación de los datos, y donde $c_{i1}, ..., c_{im_\beta}$ son los coeficientes de la combinación lineal para cada $i = \{1, ..., n\}$ de la matriz $\boldsymbol{C_{n \times m_x}}$.

Mediante las aproximaciones de $\boldsymbol{\beta}(t)$ y $\boldsymbol{x_i}(t)$ en términos de una base truncada, se obtiene la siguiente expresión:

$$y_i = \int \boldsymbol{x_i}(t)\boldsymbol{\beta}(t)dt + \varepsilon_i = \int \boldsymbol{C\varphi}(t)\boldsymbol{\phi'}(t)\boldsymbol{b}dt + \varepsilon_i = \boldsymbol{CJ_{\varphi\phi} b} + \boldsymbol{\varepsilon}, \quad t \in T \tag{3.15}$$

con $\boldsymbol{J_{\varphi\phi}}$ la matriz de dimensión $m_\beta \times m_x$, definida como:

$$\boldsymbol{J_{\varphi\phi}} = \int \boldsymbol{\varphi}(t)\boldsymbol{\phi'}(t)dt \tag{3.16}$$

Eligiendo una notación adecuada, se puede concluir que $\boldsymbol{y} = \boldsymbol{Zb} + \boldsymbol{\varepsilon}$, para $\boldsymbol{y} = (y_1, ..., y_n)$ y $\boldsymbol{\varepsilon} = (\varepsilon_1, ..., \varepsilon_n)$, consiguiendo así, transformar el problema al modelo lineal clásico y $\boldsymbol{Z} = \boldsymbol{CJ_{\varphi\phi}}$ la notación simplificada para $\boldsymbol{CJ_{\varphi\phi}}$.

Por el método de los mínimos cuadrados se puede concluir que el estimador $\hat{\boldsymbol{b}}$ es:

$$\hat{\boldsymbol{b}} = (\boldsymbol{Z'Z})^{-1}\boldsymbol{Z'y} \tag{3.17}$$

Sin embargo, el cálculo de $(\boldsymbol{Z'Z})^{-1}$ puede ocasionar problemas y podría impedir determinar el modelo de regresión. Para solucionar esta limitación, Ramsay y Silverman

[18] presentan una adaptación del modelo conocida como la versión penalizada de los mínimos cuadrados. Mediante este método, la estimación de $\boldsymbol{b}$ se obtiene como:

$$\boldsymbol{b} = (\boldsymbol{Z'Z}) + \lambda \boldsymbol{R})^{-1}\boldsymbol{Z'y} \tag{3.18}$$

Siendo $\boldsymbol{R}$ la matriz de rugosidad del métdo de estimación penalizado. El lector puede revisar [18] para más detalle de la versión penalizada de los mínimos cuadrados

### 3.2.2. Regresión con respuesta funcional

Además de relacionar variables escalares y funcionales, puede ser necesario modelar las relaciones existentes entre variables funcionales. En esta sección se pretende buscar si existe alguna dependencia entre las variables independientes a lo largo de un intervalo y la evolución de la variable dependendiente durante este.

Para el desarrollo del modelo se consideran $n$ observaciones y se toma como hipótesis tanto la variable independiente $\boldsymbol{x_i}(t) \in L^2$ con $i = \{1, ..., n\}$ como la variable respuesta $\boldsymbol{y_i}(t) \in L^2$ son funcionales y $t \in T$. El modelo con respuesta funcional se puede expresar como:

$$\boldsymbol{y_i}(t) = \boldsymbol{\alpha}(t) + \int \boldsymbol{x_i}(s)\boldsymbol{\beta}(s,t)ds + \boldsymbol{\varepsilon_i}(t), \quad s,t \in T \tag{3.19}$$

Siendo $\boldsymbol{\alpha}(t)$ el término independiente, de carácter funcional al igual que la variable dependiente; $\boldsymbol{\beta}(s,t)$ la función de coeficientes de la regresión y $\boldsymbol{\varepsilon_i}(t)$ el error.

Además se podría estudiar la relación en el momento $t = s$ (la variable $\boldsymbol{y_i}(t)$ solo se vería influenciada por el valor de la variable $\boldsymbol{x_i}(t)$ para tiempo $t = s$, no por el rango de valores adoptados por la función durante el tiempo $s$). Estos modelos se denominan concurrentes, son modelos de regresión generalizados en los que se permite que los coeficientes varíen como funciones suaves de otras variables y tienen la siguiente forma:

$$\boldsymbol{y_i}(t) = \boldsymbol{\alpha}(t) + \boldsymbol{x_i}(t)\boldsymbol{\beta}(t) + \boldsymbol{\varepsilon_i}(t), \quad i = \{1, ..., n\}, \quad t \in T \tag{3.20}$$

**Nota 3.2.1.** *Ramsay y Silverman [18] demuestran que todos los modelos lineales funcionales pueden pueden ser reducidos al problema concurrente. Para conocer más en detalle los métodos concurrentes consúltese [23].*

#### Modelo de regresión funcional múltiple con respuesta funcional

Generalizar el modelo a $q$ variables independientes, con $q > 1$, requiere definir para cada una de ellas una función $\{\boldsymbol{\beta_r}(t)\}_{r=1}^{q}$ distinta para los coeficientes de la regresión.

**Definición 3.2.1.** *Sean $n$ observaciones independientes, $\boldsymbol{y}(t)$ la variable dependiente y $\{\boldsymbol{x_1}(t), ..., \boldsymbol{x_q}(t)\}$ variables independientes explicativas de $\boldsymbol{y}(t)$. Se denomina* ***modelo de regresión múltiple*** *a la función que relaciona a todas ellas:*

$$\boldsymbol{y_i}(t) = \sum_{r=1}^{q} \boldsymbol{x_{ir}}(t)\boldsymbol{\beta_r}(t) + \boldsymbol{\varepsilon_i}(t), \quad i = \{1, ..., n\}, \quad t \in T \tag{3.21}$$

*Siendo su forma matricial: $\boldsymbol{y}(t) = \boldsymbol{X}(t)\boldsymbol{\beta}(t) + \boldsymbol{\varepsilon}(t)$, para $\boldsymbol{y} = (y_1, ..., y_n)$ la variable dependiente funcional y donde $\boldsymbol{X}(t)$ es la matriz de variables, $\boldsymbol{\beta}(t)$ son los coeficientes de la combinación lineal y $\boldsymbol{\varepsilon}(t)$ es el vector de errores.*

En este caso, será cada función $\boldsymbol{\beta_r}(t)$ con $r = \{1, ..., q\}$ la que habrá que aproximar mediante funciones de una base, pudiendo ser distintas en número y en tipo de función para cada una:

$$\boldsymbol{\beta_r}(t) = \sum_{k=1}^{k_r} \boldsymbol{b_{kr}}(t)\boldsymbol{\psi_{kr}}(t) \Rightarrow \boldsymbol{\beta_r} = \boldsymbol{\psi_r'}\boldsymbol{b_r}, \quad r = \{1, ..., q\}, \quad t \in T \tag{3.22}$$

Donde $\boldsymbol{b_r}$ son los coeficientes escalares de la aproximación funcional de $\boldsymbol{\beta_r}(t)$ mediante funciones de una base $\boldsymbol{\psi_r'}$.

El total de coeficientes que hay que estimar para determinar el modelo de regresión es:

$$k_\beta = \sum_{r=1}^{q} k_r \tag{3.23}$$

Donde $k_r$, $r = \{1, ..., q\}$, es el número de funciones de una base $\{\boldsymbol{\psi_{kr}}(t)\}_{k=1}^{k_r}$ para aproximar cada una de las funciones $\{\boldsymbol{\beta_r}(t)\}_{r=1}^{q}$.

Se define la matriz $\boldsymbol{B} = (\boldsymbol{b_1'}, ..., \boldsymbol{b_q'})$, donde cada uno de los elementos $\boldsymbol{b_r'}$, con $r = \{1, ..., q\}$ es el vector de coeficientes de la aproximación funcional de $\boldsymbol{\beta_r}(t)$ mediante funciones de una base $\boldsymbol{\psi_r'}$, y se define $\boldsymbol{\Psi} = (\boldsymbol{\psi_1'}, ..., \boldsymbol{\psi_q'})$, siendo cada $\boldsymbol{\psi_r'}$, con $r = \{1, ..., q\}$ los términos de la base que aproximan las funciones $\boldsymbol{\beta_r}(t)$.

Por tanto, se podrá expresar el modelo lineal como:

$$\boldsymbol{y}(t) = \boldsymbol{X}(t)\boldsymbol{\Phi}(t)\boldsymbol{b} + \boldsymbol{\varepsilon}(t) = \boldsymbol{X^*}(t)\boldsymbol{b} + \boldsymbol{\varepsilon}(t), \quad t \in T \tag{3.24}$$

Puede ser resuelto mediante el método de los mínimos cuadrados usando una matriz de penalización de rugosidad. No es objeto de estudio en este trabajo, pero el lector puede acudir a [18] para más información.

## 3.3. Análisis de Componentes Principales Funcionales

Al contrario que PCA, el Análisis de Componentes Principales Funcional (FPCA) considera la naturaleza funcional de los fenómenos estudiados. En el método clásico, la matriz de datos $\boldsymbol{X}_{n\times p}$ recoge la información de $n$ observaciones medidas en $p$ variables, sin embargo, en el FPCA, se trabaja desde el punto de vista de $n$ funciones $\boldsymbol{x_i}(t)$, con $i = \{1, ..., n\}$.

En el contexto multivariante, para PCA se demostró que las componentes principales eran fruto de la combinación lineal de las variables originales $\{\boldsymbol{x_1}, ..., \boldsymbol{x_p}\}$. Dicha combinación lineal se construye a partir del producto escalar entre los coeficientes de la combinación y las variables originales. Para generalizar este proceso al marco funcional, se necesita recurrir al producto escalar para espacios de Hilbert (definido en la sección 3.1.1) y que está determinado por una integral en vez de por un sumatorio:

$$\langle \boldsymbol{a}, \boldsymbol{x} \rangle = \int \boldsymbol{a}(t)\boldsymbol{x}(t)dt \tag{3.25}$$

Donde $t \in T$ y tanto $\boldsymbol{a}(t)$ como $\boldsymbol{x}(t)$ son funciones. Ahora las componentes principales $\boldsymbol{y_j}(t)$, con $t \in T$ para datos funcionales quedan definidas como:

$$\begin{aligned} \boldsymbol{y_1}(t) &= \int \boldsymbol{a_1}(t)\boldsymbol{x}(t)dt \\ \boldsymbol{y_2}(t) &= \int \boldsymbol{a_2}(t)\boldsymbol{x}(t)dt \\ &\vdots \end{aligned} \tag{3.26}$$

Donde las funciones $\boldsymbol{a_j}(t)$ definen pesos normalizados que maximizan la variación de $\boldsymbol{y_j}(t)$, siendo $j = \{1, ..., j, ...\}$ y $\boldsymbol{x}(t) = \{\boldsymbol{x_1}(t), ..., \boldsymbol{x_n}(t)\}$.

El máximo de parejas formadas por valores propios y vectores propios son diferentes para el caso multivariante y funcional. Para el multivariante son tantas como variables, sin embargo para el funcional son infinitas, puesto que el espacio es $L^2[0, T]$. Además, las $n$ funciones de la muestra son linealmente independientes, por lo que la matriz de covarianzas tendrá rango n y como máximo $n$ componentes principales -máximo níumero de valores propios no nulos-. En los casos prácticos es habitual seleccionar $k$ de las $n$ con $(k < n)$ componentes principales.

Formalmente, se busca resolver, para cada $i \in \{1, ..., n\}$, la siguiente sucesión de problemas (problema del FPCA):

- $\max\limits_{\boldsymbol{a_1}\in L^2} \{\sum_{i=1}^{n} \langle \boldsymbol{a_1}, \boldsymbol{x_i} \rangle^2\}$ s.a $\|\boldsymbol{a_1}\| = 1$

- $\max_{\boldsymbol{a_2}\in L^2} \{\sum_{i=1}^{n}\langle \boldsymbol{a_2}, \boldsymbol{x_i}\rangle^2\}$ s.a $\|\boldsymbol{a_2}\| = 1$ y $\langle \boldsymbol{a}_1, \boldsymbol{a_2}\rangle = 0$
- $\vdots$
- $\max_{\boldsymbol{a_j}\in L^2} \{\sum_{i=1}^{n}\langle \boldsymbol{a_j}, \boldsymbol{x_i}\rangle^2\}$ s.a $\|\boldsymbol{a_j}\| = 1$ y $\langle \boldsymbol{a_k}, \boldsymbol{a}_j\rangle = 0$ para $k < j$

En el caso del FPCA, se verá como cada valor propio está asociado a una función propia en vez de a un vector propio. Al igual que se ha demostrado para el PCA, se comprobará para el FPCA que las funciones propias albergan la variabilidad del conjunto de datos.

Otro cambio significante es con respecto a la matriz de covarianzas, que es sustituida por la función bivariante:

$$\boldsymbol{v}(s,t) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x_i}(s)\boldsymbol{x_i}(t) \tag{3.27}$$

Se consideran las funciones $\boldsymbol{x_i}$ centradas.

Para el PCA se ha verificado que las componentes principales son combinaciones lineales de las variables originales, donde los coeficientes se corresponden con los vectores propios de la matriz de covarianzas. A continuación se muestra la generalización del problema para datos funcionales.

**Proposición 3.3.1.** *Dadas $\boldsymbol{x_i}(t)$ funciones centradas, $i = \{1, ..., n\}$ y definidas sobre el espacio $L^2$. Sea la función $\boldsymbol{v}(s,t)$ donde:*

$$\boldsymbol{V}: \begin{array}{ccc} T\times T & \longrightarrow & \mathbb{R} \\ (s,t) & \longmapsto & \boldsymbol{v}(s,t) \end{array} \quad \textit{siendo} \quad \boldsymbol{v}(s,\cdot): \begin{array}{ccc} T & \longrightarrow & \mathbb{R} \\ s & \longmapsto & \int \boldsymbol{v}(s,t)\boldsymbol{y}(t)dt \end{array} \tag{3.28}$$

*donde $\boldsymbol{v}(s,\cdot)$ es función de $s$ e $\boldsymbol{y}(t)$ es función de $t$. Entonces, las componentes principales funcionales, calculadas como la solución al problema del FPCA, son resultado de la siguiente ecuación:*

$$\int \boldsymbol{v}(s,t)\boldsymbol{a}(t)dt = \lambda \boldsymbol{a}(t) \tag{3.29}$$

*donde $\lambda$ es el valor propio de $\boldsymbol{V}$ y $\boldsymbol{a}(t)$ su correspondiente función propia.*

La ecuación 3.29 es equivalente a la ecuación del problema clásico multivariante 1.29. Sin embargo, en el marco funcional, su resolución no es inmediata y por motivos de extensión no se ha incluído. En [12] y [7] se discuten varias alternativas para la resolución y el lector puede encontrar en [18] la que se ha tomado para este trabajo. Asumiendo que la ecuación 3.29 puede resolverse siguiendo los pasos de [18] y que los valores propios que verifican la ecuación se disponen en orden descendiente como $\lambda_1 > ... > \lambda_k$, entonces los coeficientes correspondientes a la combinación lineal de la primera componente principal funcional se construyen con $\boldsymbol{a_1}(t)$, siendo $\boldsymbol{a_1}(t)$ la función propia de

$\boldsymbol{V}$ asociada al primer valor propio. De la misma manera, los coeficientes de la segunda componente principal son los elementos de la función $\boldsymbol{a_2}(t)$, siendo $\boldsymbol{a_2}(t)$ la función propia de $\boldsymbol{V}$ asociada al valor propio $\lambda_2$ y además, como las componentes principales funcionales deben formar un sistema ortogonal, se tiene que cumplir:

$$\int \boldsymbol{a_j}(t)\boldsymbol{a_k}(t)dt = 0, \quad j < k \tag{3.30}$$

Siguiendo el mismo razonamiento, los coeficientes de la componente $j$-ésima son los elementos de la función propia $j$-ésima $\boldsymbol{a_j}$ de $\boldsymbol{V}$, siempre respetando la reestricción de ortogonalidad como ya se ha mencionado.

Por tanto, se concluye que las puntuaciones de las observaciones en la componente $j$-ésima:

$$y_{ij} = \int \boldsymbol{a_j}(t)\boldsymbol{x_i}(t)dt, \quad i = \{1, ..., n\} \tag{3.31}$$

### 3.3.1. Variabilidad de las componentes principales funcionales

En la sección 1.3.2 quedó demostrado que el total de la variabilidad que recogen las componentes principales es equivalente a la traza de la matriz de covarianzas para $\boldsymbol{Y}$. La extrapolación al caso funcional es inmediata mediante la siguiente definición:

**Definición 3.3.1.** *La **variabilidad** total del modelo funcional está determinada por la siguiente expresión:*

$$\frac{1}{n}\sum_{i=1}^{n}\int \boldsymbol{x_i}^2(t)dt = \sum_{i=1}^{n}\lambda_i \tag{3.32}$$

**Definición 3.3.2.** *La **proporción de varianza acumulada** por las $k$ primeras componentes principales funcionales se puede calcular mediante:*

$$\frac{\sum_{i=1}^{k}\lambda_i}{\sum_i \lambda_i} \tag{3.33}$$

En cuanto a la elección del número de componentes, es una decisión que debe tomar el analista apoyándose en criterios mencionados en el capítulo uno.

## 3.4. Análisis de cluster para datos funcionales

Numerosos métodos se han publicado para el análisis de cluster funcional. Jacques y Preda [9] recogen en 2014 una clasificación con las distintas corrientes con las técnicas principales del análisis de cluster funcional. Una vertiente se denomina agrupación de datos en bruto y consiste en agrupar los datos discretos sin recuperar su naturaleza

funcional; otras técnicas, conocidas como métodos en dos pasos, consisten en una primera fase, de reducción de la dimensionalidad, y una segunda de clasificación; Jacques y Preda [9] también mencionan los métodos no parámetricos, basados en métodos no probabilísticos diseñados para espacios finitos. Actualmente se pueden diferenciar dos categorías en el enfoque no paramétrico; métodos que aplican técnicas ya conocidas de clustering jerárquico o no jerárquico como K-Medias, para distancias elegidas a priori, y métodos que introducen nuevos criterios geométricos para agrupar los datos funcionales. De la primera categoría resalta [22] donde se demuestra que los centroides de los clusters son combinación lineal de las funciones propias del FPCA. Se prestará especial atención al método de clasificación no jerárquico K-Medias para datos funcionales. De la segunda categoría se podrían destacar trabajos como los de Yamamoto [24] que desarrolla un nuevo proceso para detectar clusters óptimos de funciones y subespacios óptimos para el clustering simultáneamente.

### 3.4.1. Métodos no jerárquicos: K-Medias funcional

El método no jerárquico de K-Medias en el contexto de datos funcionales, al igual que el método clásico, busca agrupar $n$ observaciones en $m$ clusters mediante sucesivas reasignaciones hasta que el método converja. Se ha tomado como referencia [20].

En el contexto funcional, las $n$ observaciones son funciones $\boldsymbol{x_i}(t)$ con $i = \{1, ..., n\}$ para $t \in T$, y son los elementos que habrá que clasificar en $m$ clusters $h_1, ..., h_m$, donde $|h_j|$ con $j = \{1, ..., m\}$ será el número de elementos de cada cluster.

**Definición 3.4.1.** *Para cada cluster $h_j$, con $j = \{1, ..., m\}$, se define el **centroide** como la función $\boldsymbol{c_j}(t)$ para $t \in T$, siendo el centroide la función equidistante a todas las funciones del cluster.*

Además se define el valor binario $u_{ji}(t)$ con $i = \{1, ..., n\}$, $j = \{1, ..., m\}$, $t \in T$ como la asignación del elemento $i$-ésimo al cluster $j$-ésimo durante una de las iteraciones del proceso. De la misma forma se define $\hat{u}_{ji}(t)$, para la iteración anterior y se denotará como $\hat{\boldsymbol{c}}_{\boldsymbol{j}}(t)$ al centroide calculado en la iteración anterior a $\boldsymbol{c_j}(t)$.

El método tiene como objetivo minimizar la siguiente función:

$$\boldsymbol{J} = \int \sum_{i=1}^{n} \sum_{j=1}^{m} u_{ji}(t) \|\boldsymbol{x_i}(t) - \boldsymbol{c_j}(t)\|^2 dt \tag{3.34}$$

Y para ello se recurre a un proceso iterativo (con un número máximo de iteraciones prefijadas). Dadas $\varepsilon_1$ y $\varepsilon_2$ dos constantes cercanas a cero, en el diagrama de la figura 3.1 se presenta el algoritmo explicado:

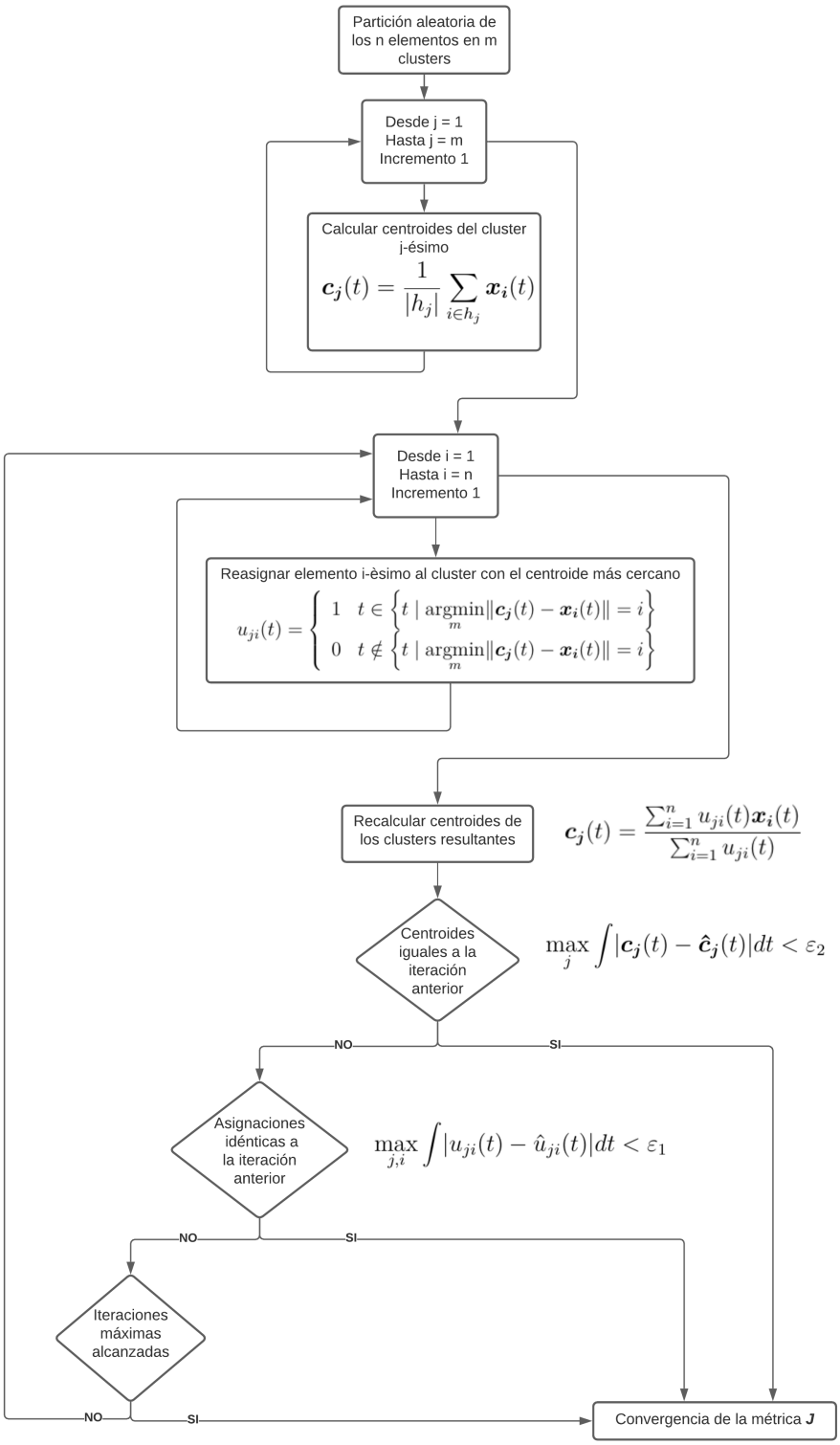



Figura 3.1: Algoritmo K-Medias Funcional. Fuente: elaboración propia.

# Capítulo 4

# Aplicación a datos reales

En este último capítulo se pretende ilustrar la utilidad de las técnicas estadísticas en el análisis de datos funcionales mediante un caso práctico. Se llevará a cabo el estudio de un conjunto de datos reales de la AEMET disponibles en el repositorio [1]. Es un dataset de registros de temperaturas máximas, medias y mínimas durante el año 2017 para veinte estaciones meteorológicas españolas.

Los datos no presentan ninguna clasificación a priori, pero atendiendo a su situación geográfica, se ha incluído manualmente el clima esperado para cada estación. Los climas que intervienen en la aplicación son el clima de interior mediterráneo, marcado por las diferencias más bruscas de temperaturas durante el año y situado en la mayor parte del interior de la península; el clima oceánico, caracterizado por tener unas curvas algo más estables durante el año que el clima de interior y situado en el norte de la península; y el clima subtropical, sin apenas variaciones durante todo el año y que afecta a las Islas Canarias. En el cuadro 4.1 se presentan las estaciones meteorológicas implicadas en el análisis agrupadas según su clima esperado por situación geográfica.

| Interior Mediterráneo | | Oceánico | Subtropical |
|---|---|---|---|
| Albacete | Ciudad Real | A coruña | Güimar |
| Cuenca | Guadalajara | Bilbao aeropuerto | El Hierro |
| Madrid aeropuerto | Salamanca | Santander aeropuerto | Lanzarote |
| Navalmoral de la Mata | Toledo | Santander | Santa Cruz de Tenerife |
| Torrejón de Ardoz | Valdepeñas | Santiago de Compostela | |
| | | Zumaia | |

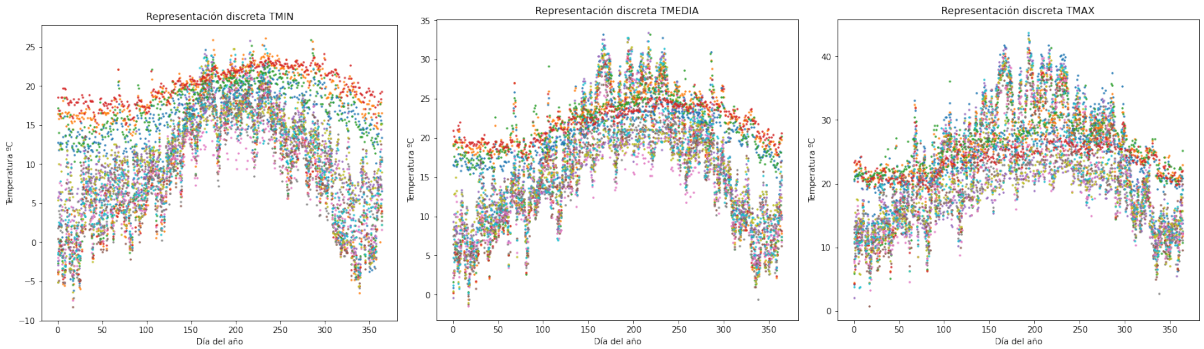
Cuadro 4.1: Clasificación de las estaciones. Fuente: elaboración propia.

El objetivo perseguido es analizar los patrones de las tendencias almacenadas para cada estación y poder demostrar que la clasificación autoḿatica de este tipo de datos es posible.

El entorno de trabajo utilizado para llevar a cabo el caso práctico ha sido un cuaderno de Jupyter de Anaconda y el lenguaje de programación escogido, Python. El lenguaje Python destaca por su sencillez y potencial en representación de datos gracias a librerías como *matplotlib*. Además, permite trabajar de manera sencilla en la fase de minería de datos con librerías como *numpy* y *pandas*. Para el análisis de datos funcionales se ha recurrido a la librería *skfda*, que dispone de las herramientas principales tanto en representación, como para el análisis. En este capítulo se comentarán los resultados obtenidos, aunque el código de python para la realización del análisis y representación de gráficas se encuentra en el Anexo I explicado.

### Muestra discreta de datos

En la figura 4.1 se puede observar para cada variable del análisis (temperatura mínima, media y máxima) los datos discretos del conjunto de datos. Para cada uno de ellos, el rango de valores oscila entre valores diferentes; siendo el gráfico de temperaturas mínimas el que tiene los valores mínimos alcanzados, entorno a -7ºC y el gráfico de temperaturas altas registra el valor máximo con aproximadamente 45ºC. Como se ha mencionado en múltiples ocasiones en el cuerpo teórico, los datos a pesar de tener una naturaleza funcional, se almacenan como una muestra discreta y posteriormente se procesan para recuperar la esencia funcional. En la figura 4.1 se presenta la muestra discreta original de los datos de partida.

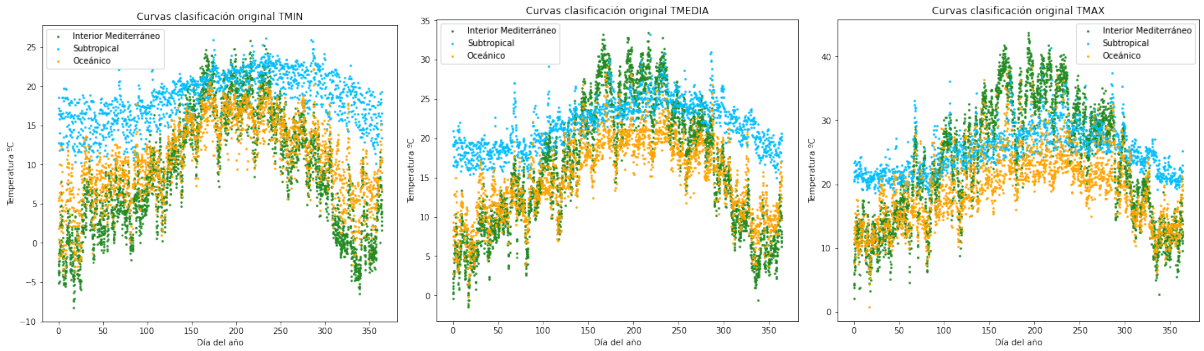



(a) Mínimas (b) Medias (c) Máximas

Figura 4.1: Representación discreta por variable. Fuente: elaboración propia.

Como primer paso del análisis, en la figura 4.2, se aplica sobre los datos discretos la clasificación de las estaciones de la figura 4.1 para reforzar la hipótesis de que existe

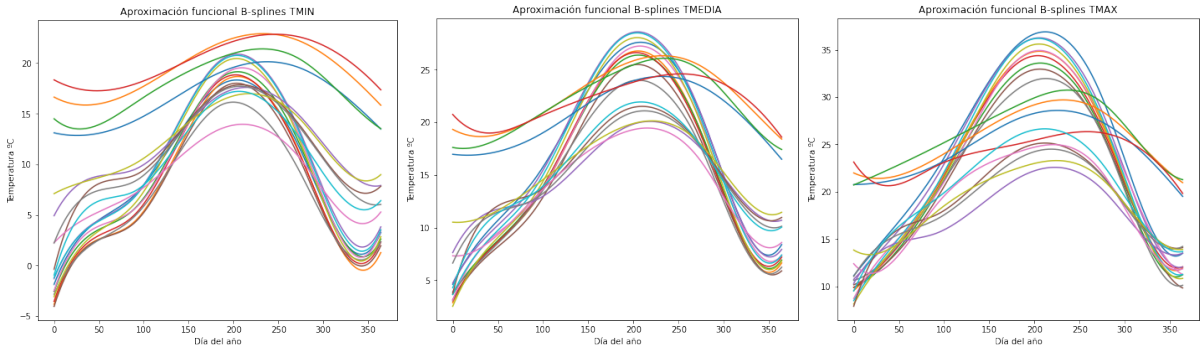
algún patrón en el comportamiento de las temperaturas por zona climática. La muestra del grupo interior mediterráneo (representada en color verde) no registra apenas cambios a lo largo del año, con una tendencia muy plana; la muestra del clima subtropical (color azul) contiene las mayores diferencias entre los mínimos y máximos alcanzados para cada variable; y la muestra del grupo oceánico (naranja), aunque presente grandes diferencias, ni sus mínimos, ni sus máximos son tan extremos como para la verde.




(a) Mínimas (b) Medias (c) Máximas

Figura 4.2: Datos discretos clasificados por variable. El color verde representa el clima interior mediterráneo, el azul el clima subtropical y el naranja, el clima oceánico. Fuente: elaboración propia.

En primer lugar, teniendo en cuenta que los datos originales han sido importados como una muestra discreta, se ha realizado la aproximación de los datos a funciones mediante una base de splines, dado que aporta un buen ajuste de los datos con un número pequeño de funciones.




(a) Mínimas (b) Medias (c) Máximas

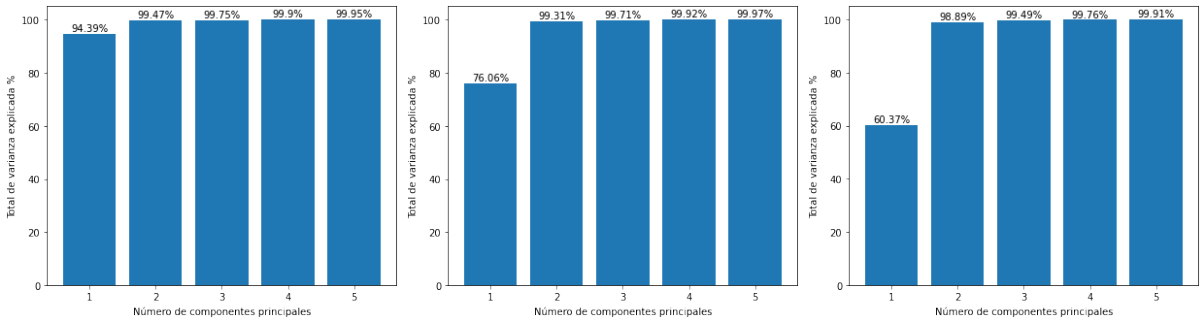
Figura 4.3: Aproximación funcional de los datos. Fuente: elaboración propia.

Con la conversión se consigue obtener por cada estación y variable una función dependiente del tiempo. Se trata de una aproximación suavizada de los datos originales. La aproximación varía según el número de funciones de la base elegidos como ya se mostró en la figura 2.2 en el capítulo dos. Se busca obtener un buen ajuste de los datos con el número mínimo de funciones.

### Análisis de Componentes Principales Funcionales

En segundo lugar se ha llevado a cabo el análisis de componentes principales funcionales de los datos. Se pretende proyectar los datos sobre un espacio de dimensión menor para facilitar la representación y comprensión de la información.

En la teoría se afirma que las componentes principales están ordenadas de mayor a menor en cuanto a cantidad de variabilidad almacenada. A continuación, en la figura 4.4 se muestra para este conjunto de datos, la variabilidad acumulada por las cinco primeras componentes para cada variable.




(a) Mínimas (b) Medias (c) Máximas

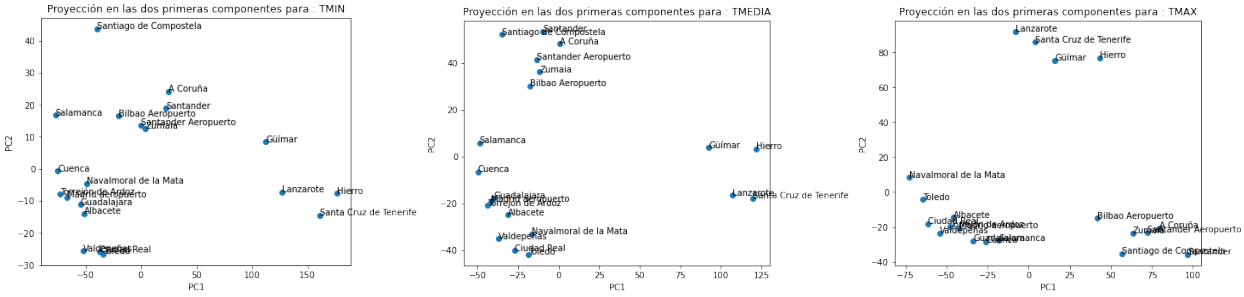
Figura 4.4: Porcentaje de varianza acumulada explicada por número de componentes principales para cada parámetro. Fuente: elaboración propia.

Se puede observar que en todos los casos, tomando las dos primeras componentes principales se supera el 90 % de variabilidad de los datos y además, a partir de la segunda componente se estabiliza la variabilidad acumulada. Esto implica que tanto el criterio del porcentaje, como el criterio del bastón roto mencionados en el capítulo uno avalan la decisión de escoger las dos primeras componentes principales para la representación de los datos.

En concreto, en la figura 4.5 se muestra la proyección de los datos en las dos primeras componentes principales funcionales. Con la representación de la figura 4.2, ya se podía intuir el comportamiento para cada tipo de clima, pero la proyección en las dos primeras componentes principales genera tres grupos claramente diferenciados y en los que las

estaciones meteorológicas situadas en zonas con el mismo tipo de clima aparecen más próximas. Tomando como referencia la tabla de la clasificación de las estaciones de la figura 4.1, se puede observar que en cada grupo aparecen las estaciones de cada zona climatológica. Se puede apreciar que para las temperaturas medias y máximas se diferencian perfectamente los grupos, sin embargo, para las temperaturas mínimas el gráfico tiene algo más de dispersión. Los grupos para las tres variables son:

1. Grupo interior: Albacete, Ciudad Real, Cuenca, Guadalajara, Madrid aeropuerto, Salamanaca, Navalmoral de la Mata, Toledo, Torrejón de Ardoz y Valdeñas.
2. Grupo oceánico: A Coruña, Bilbao aeropuerto, Santander aeropuerto, Santander, Santiago de Compostela y Zumaia.
3. Grupo subtropical: Güimar, El Hierro, Lanzarote y Santa Cruz de Tenerife.



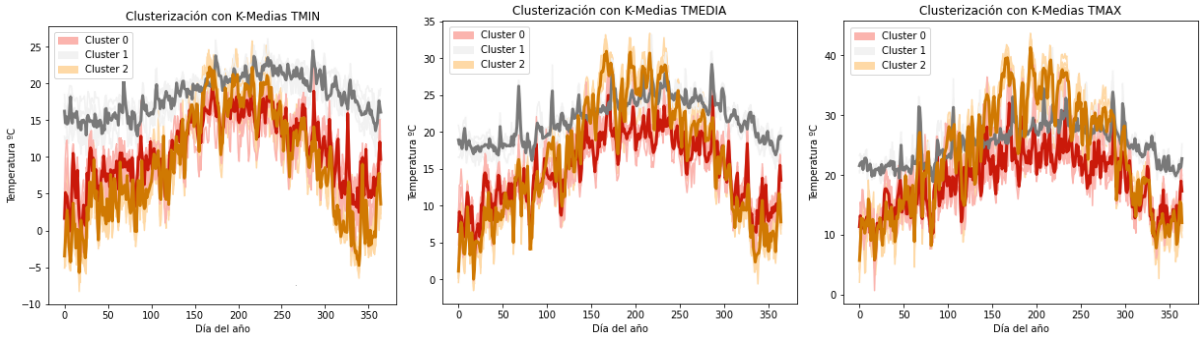

(a) Mínimas (b) Medias (c) Máximas

Figura 4.5: Proyección en las dos primeras componentes principales. Fuente: elaboración propia.

### Análisis de Cluster Funcional

Por último, se ha llevado a la práctica un ejemplo de análisis de cluster funcional utilizando el método de K-Medias para datos funcionales. Como ya se detalló en la parte teórica, K-Medias requiere especificar a priori el número de clusters. En el proceso teórico se planteaba la posibilidad de realizar varias repeticiones con distinto número de grupos, no obstante, con el análisis previo de componentes principales no ha sido necesario y se ha realizado una única repetición en busca de los tres grupos correspondientes a los tres climas.

Seguidamente, en la figura 4.6 se presentan los resultados de la clusterización. Para cada tipo de temperaturas se pueden ver tres clusters; de color gris correspondiente al cluster 1, de color rojo al cluster 0 y de color naranja al cluster 2. Para cada cluster aparecen coloreadas las funciones de ese grupo del mismo color y en un tono más oscuro, también queda representado el centroide de cada cluster.




(a) Mínimas (b) Medias (c) Máximas

Figura 4.6: Representación de clusters por K-Medias para cada parámetro. Fuente: elaboración propia.

El resultado de la predicción del cluster de cada temperatura es un *array* numérico en el que se guarda la asignación del cluster al que pertenece cada función correspondiente a una estación meteorológica. En el cuadro 4.2 se muestra dicha información acompañada del nombre de la estación meteorológica y la clasificación por zona climática que se hizo a priori. Se presenta una única tabla porque el resultado para las tres temperaturas ha sido idéntico.

| | Estación Meteorológica | Tipo de clima original | CLUSTER | | Estación Meteorológica | Tipo de clima original | CLUSTER |
|---|---|---|---|---|---|---|---|
| 0 | Navalmoral de la Mata | Interior Mediterráneo | 2 | 10 | Güímar | Subtropical | 1 |
| 1 | Torrejón de Ardoz | Interior Mediterráneo | 2 | 11 | Santa Cruz de Tenerife | Subtropical | 1 |
| 2 | Guadalajara | Interior Mediterráneo | 2 | 12 | Lanzarote | Subtropical | 1 |
| 3 | Madrid aeropuerto | Interior Mediterráneo | 2 | 13 | Hierro | Subtropical | 1 |
| 4 | Toledo | Interior Mediterráneo | 2 | 14 | Santander | Oceánico | 0 |
| 5 | Cuenca | Interior Mediterráneo | 2 | 15 | Zumaia | Oceánico | 0 |
| 6 | Albacete | Interior Mediterráneo | 2 | 16 | Santiago de Compostela | Oceánico | 0 |
| 7 | Salamanca | Interior Mediterráneo | 2 | 17 | Santander Aeropuerto | Oceánico | 0 |
| 8 | Valdepeñas | Interior Mediterráneo | 2 | 18 | A Coruña | Oceánico | 0 |
| 9 | Ciudad Real | Interior Mediterráneo | 2 | 19 | Bilbao Aeropuerto | Oceánico | 0 |

Cuadro 4.2: Clasificación detallada. Fuente: elaboración propia.

Se concluye que el cluster 0 se corresponde con las estaciones del clima oceánico; el cluster 1, con las del clima subtropical; y el cluster 2 se corresponde con las estaciones del clima de interior mediterráneo. Todos ellos con una tasa de acierto de clasificación del 100 % para cada variable.

# Conclusiones

En la memoria se ha revisado el estado del arte de las técnicas estadísticas multivariantes con el objetivo de demostrar su capacidad en el análisis de datos. Se ha destacado el modelo de regresión lineal en la tarea de estudiar las relaciones entre varias variables explicativas y dependientes; la descomposición en valores singulares, así como su aplicación en la reducción de la dimensionalidad, con su uso en la formulación de las componentes principales. Se ha explicado mediante el análisis de componentes principales cómo el proceso de expresar la máxima información con el menor conjunto de datos culmina con la proyección de los datos sobre un espacio de dimensión inferior a la original. Además, con el análisis de cluster se ha ilustrado cómo agrupar los datos según un criterio establecido y la importancia en la detección de patrones. Con la ayuda de la geometría ultramétrica se ha introducido el método jerárquico como proceso de clustering y para la vertiente no jerárquica se han presentado métodos de reasignación como el de K-Medias y algunas de sus resoluciones definiendo como condición de parada la convergencia de la métrica elegida.

Para justificar la evolución hacia datos funcionales, se han señalado las limitaciones del análisis multivariante como la no homogeneización de las mediciones y la necesidad de avanzar en dirección a modelos que recogen la naturaleza continua de cierta información. Se presenta una breve introducción a la adaptación de algunas de las técnicas multivariantes clásicas al contexto funcional. Se ha expuesto la metodología de recuperación de la forma funcional de los procesos por medio de aproximaciones de bases de funciones, ya que se hace patente que aunque se trabaje con procesos continuos, las observaciones no dejan de ser discretas.

La representación mediante bases proporciona una "discretización" de los datos con el conocimiento continuo subyacente. Es aquí donde se comienza a vislumbrar la utilidad de las técnicas multivariantes en el análisis de datos funcionales.

Se concluye que la regresión lineal para datos funcionales es capaz de modelar situaciones con respuesta escalar y variable independiente funcional, así como escenarios con respuesta funcional y variable independiente funcional. Para abordar el FPCA se ha generalizado el problema de PCA a espacios de Hilbert y se ha conseguido obtener una

expresión del problema equivalente al problema clásico multivariante. Como última técnica, se explica Análisis de Cluster para datos funcionales, en especial, el método de K-Medias funcional.

Por último, se muestra la utilidad de las técnicas mencionadas para el análisis de un conjunto de datos real. El software utilizado ha sido Python en el entorno de Anaconda. En la aplicación con datos reales se han analizado los datos de veinte estaciones meteorológicas españolas en las que se han registrado los datos de temperaturas mínimas, medias y máximas durante todo el año de 2017 con un intervalo de un día. En primer lugar, se han representado los datos tal y como han sido almacenados; en forma discreta. En segundo lugar, se ha recuperado la naturaleza funcional de los datos haciendo uso de las herramientas proporcionadas por Python. Posteriormente, para facilitar la representación de los datos en una dimensión menor se ha recurrido al FPCA y se ha visto que la representación en dos componentes principales funcionales retenía más del 90 % de la variabilidad de los datos. Por último se ha realizado el análisis de cluster funcional, aplicando el método de K-Medias funcional, en busca de patrones latentes en los datos y se concluye demostrando la capacidad de estas técnicas en la identificación de observaciones con comportamientos similares en el estudio de procesos esencialmente continuos. En concreto, se ha logrado con FPCA representar los grupos diferenciados de las distintas zonas climáticas (mediterráneo interior, oceánico y subtropical) y mediante el análisis de cluster, ofrecer una predicción para cada una de las estaciones coincidente con su situación geográfica.

# Anexos

# Apéndice A

# Código en Python

Se importan las librerías necesarias para el análisis. Las librerías *pandas* y *numpy*, muy útiles para el tratamiento de datos; la librería *matplotlib* para representación de gráficas; y la librería *skfda* es la que almacena todas las técnicas de análisis de datos funcionales.

```
#Importar librerias de programacion necesarias
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import skfda as fda
```

Se importan todos los archivos para cada una de las estaciones meteorológicas en formato .csv mediante la librería *pandas*. En total se importan datos para veinte estaciones y para cada una de ellas datos referentes a las temperaturas mínimas, medias y máximas alcanzadas para cada fecha. Para cada archivo se formatea la columna de la fecha para darle el conocimiento de año, mes y día.

```
# Importar conjunto de datos y formatear fecha
toledo = pd.read_csv('/Users/anaro/Downloads/DatosPorEstacion/toledo.csv↩
    ', sep=";")
toledo['FECHA'] = pd.to_datetime(toledo['FECHA'], format="%Y/%m/%d")

guadalajara = pd.read_csv('/Users/anaro/Downloads/DatosPorEstacion/↩
    guadalajara.csv', sep=";")
guadalajara['FECHA'] = pd.to_datetime(guadalajara['FECHA'], format="%Y/%↩
    m/%d")

salamanca = pd.read_csv('/Users/anaro/Downloads/DatosPorEstacion/↩
    salamanca.csv', sep=";")
salamanca['FECHA'] = pd.to_datetime(salamanca['FECHA'], format="%Y/%m/%d↩
    ")
```

```
albacete = pd.read_csv('/Users/anaro/Downloads/DatosPorEstacion/albacete↩
    .csv', sep=";")
albacete['FECHA'] = pd.to_datetime(albacete['FECHA'], format="%Y/%m/%d")

ciudad_real = pd.read_csv('/Users/anaro/Downloads/DatosPorEstacion/↩
    ciudad_real.csv', sep=";")
ciudad_real['FECHA'] = pd.to_datetime(ciudad_real['FECHA'], format="%Y/%↩
    m/%d")

valdepenas = pd.read_csv('/Users/anaro/Downloads/DatosPorEstacion/↩
    valdepenas.csv', sep=";")
valdepenas['FECHA'] = pd.to_datetime(valdepenas['FECHA'], format="%Y/%m↩
    /%d")

lanzarote_aeropuerto = pd.read_csv('/Users/anaro/Downloads/↩
    DatosPorEstacion/lanzarote_aeropuerto.csv', sep=";")
lanzarote_aeropuerto['FECHA'] = pd.to_datetime(lanzarote_aeropuerto['↩
    FECHA'], format="%Y/%m/%d")

hierro_aeropuerto = pd.read_csv('/Users/anaro/Downloads/DatosPorEstacion↩
    /hierro_aeropuerto.csv', sep=";")
hierro_aeropuerto['FECHA'] = pd.to_datetime(hierro_aeropuerto['FECHA'], ↩
    format="%Y/%m/%d")

santiago_aeropuerto = pd.read_csv('/Users/anaro/Downloads/↩
    DatosPorEstacion/santiago_aeropuerto.csv', sep=";")
santiago_aeropuerto['FECHA'] = pd.to_datetime(santiago_aeropuerto['FECHA↩
    '], format="%Y/%m/%d")

acoruna = pd.read_csv('/Users/anaro/Downloads/DatosPorEstacion/acoruna.↩
    csv', sep=";")
acoruna['FECHA'] = pd.to_datetime(acoruna['FECHA'], format="%Y/%m/%d")

santander_aeropuerto = pd.read_csv('/Users/anaro/Downloads/↩
    DatosPorEstacion/santander_aeropuerto.csv', sep=";")
santander_aeropuerto['FECHA'] = pd.to_datetime(santander_aeropuerto['↩
    FECHA'], format="%Y/%m/%d")

santander = pd.read_csv('/Users/anaro/Downloads/DatosPorEstacion/↩
    santander.csv', sep=";")
santander['FECHA'] = pd.to_datetime(santander['FECHA'], format="%Y/%m/%d↩
    ")

cuenca = pd.read_csv('/Users/anaro/Downloads/DatosPorEstacion/cuenca.csv↩
    ', sep=";")
cuenca['FECHA'] = pd.to_datetime(cuenca['FECHA'], format="%Y/%m/%d")

zumaia = pd.read_csv('/Users/anaro/Downloads/DatosPorEstacion/zumaia.csv↩
    ', sep=";")
zumaia['FECHA'] = pd.to_datetime(zumaia['FECHA'], format="%Y/%m/%d")

madrid_aeropuerto = pd.read_csv('/Users/anaro/Downloads/DatosPorEstacion↩
    /madrid_aeropuerto.csv', sep=";")
madrid_aeropuerto['FECHA'] = pd.to_datetime(madrid_aeropuerto['FECHA'], ↩
    format="%Y/%m/%d")

guimar = pd.read_csv('/Users/anaro/Downloads/DatosPorEstacion/guimar.csv↩
    ', sep=";")
guimar['FECHA'] = pd.to_datetime(guimar['FECHA'], format="%Y/%m/%d")

vigo_aeropuerto = pd.read_csv('/Users/anaro/Downloads/DatosPorEstacion/↩
    vigo_aeropuerto.csv', sep=";")
```

```
vigo_aeropuerto['FECHA'] = pd.to_datetime(vigo_aeropuerto['FECHA'], ↩
    format="%Y/%m/%d")

santa_cruz_de_tenerife = pd.read_csv('/Users/anaro/Downloads/↩
    DatosPorEstacion/santa_cruz_de_tenerife.csv', sep=";")
santa_cruz_de_tenerife['FECHA'] = pd.to_datetime(santa_cruz_de_tenerife[↩
    'FECHA'], format="%Y/%m/%d")

torrejon_de_ardoz = pd.read_csv('/Users/anaro/Downloads/DatosPorEstacion↩
    /torrejon_de_ardoz.csv', sep=";")
torrejon_de_ardoz['FECHA'] = pd.to_datetime(torrejon_de_ardoz['FECHA'], ↩
    format="%Y/%m/%d")

navalmoral_de_la_mata = pd.read_csv('/Users/anaro/Downloads/↩
    DatosPorEstacion/navalmoral_de_la_mata.csv', sep=";")
navalmoral_de_la_mata['FECHA'] = pd.to_datetime(navalmoral_de_la_mata['↩
    FECHA'], format="%Y/%m/%d")

bilbao_aeropuerto = pd.read_csv('/Users/anaro/Downloads/DatosPorEstacion↩
    /bilbao_aeropuerto.csv', sep=";")
bilbao_aeropuerto['FECHA'] = pd.to_datetime(bilbao_aeropuerto['FECHA'], ↩
    format="%Y/%m/%d")
```

Se definen las variables start, end y parametro para poder seleccionar un período de estudio y la variable. El código se ha ejecutado en tres ocasiones para obtener el resultado para cada variable [TMIN, TMEDIA, TMAX]. Una vez definidos las variables, se filtra cada conjunto de datos.

```
# Parametrizacion del perido a analizar y variable
start = '2017-01-01'
end = '2017-12-31'
parametro = 'TMIN'

# Filtrado de datos segun la parametrizacion
toledo_period =toledo.loc[toledo["FECHA"].between(start, end)][parametro↩
    ].to_numpy()
guadalajara_period =guadalajara.loc[guadalajara["FECHA"].between(start, ↩
    end)][parametro].to_numpy()
salamanca_period =salamanca.loc[salamanca["FECHA"].between(start, end)][↩
    parametro].to_numpy()
albacete_period =albacete.loc[albacete["FECHA"].between(start, end)][↩
    parametro].to_numpy()
ciudad_real_period =ciudad_real.loc[ciudad_real["FECHA"].between(start, ↩
    end)][parametro].to_numpy()
valdepenas_period =valdepenas.loc[valdepenas["FECHA"].between(start, end↩
    )][parametro].to_numpy()
lanzarote_aeropuerto_period = lanzarote_aeropuerto.loc[↩
    lanzarote_aeropuerto["FECHA"].between(start, end)][parametro].↩
    to_numpy()
hierro_aeropuerto_period = hierro_aeropuerto.loc[hierro_aeropuerto["↩
    FECHA"].between(start, end)][parametro].to_numpy()
santiago_aeropuerto_period = santiago_aeropuerto.loc[santiago_aeropuerto↩
    ["FECHA"].between(start, end)][parametro].to_numpy()
acoruna_period = acoruna.loc[acoruna["FECHA"].between(start, end)][↩
    parametro].to_numpy()
santander_aeropuerto_period = santander_aeropuerto.loc[↩
    santander_aeropuerto["FECHA"].between(start, end)][parametro].↩
    to_numpy()
```

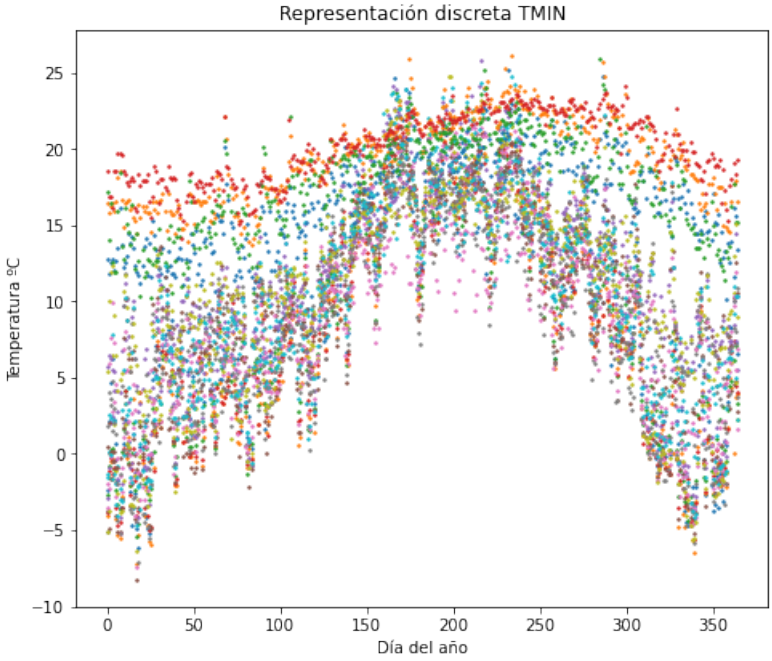
```
santander_period = santander.loc[santander["FECHA"].between(start, end)↩
    ][parametro].to_numpy()
cuenca_period = cuenca.loc[cuenca["FECHA"].between(start, end)][↩
    parametro].to_numpy()
zumaia_period = zumaia.loc[zumaia["FECHA"].between(start, end)][↩
    parametro].to_numpy()
madrid_aeropuerto_period = madrid_aeropuerto.loc[madrid_aeropuerto["↩
    FECHA"].between(start, end)][parametro].to_numpy()
guimar_period = guimar.loc[guimar["FECHA"].between(start, end)][↩
    parametro].to_numpy()
vigo_aeropuerto_period = vigo_aeropuerto.loc[vigo_aeropuerto["FECHA"].↩
    between(start, end)][parametro].to_numpy()
santa_cruz_de_tenerife_period = santa_cruz_de_tenerife.loc[↩
    santa_cruz_de_tenerife["FECHA"].between(start, end)][parametro].↩
    to_numpy()
torrejon_de_ardoz_period = torrejon_de_ardoz.loc[torrejon_de_ardoz["↩
    FECHA"].between(start, end)][parametro].to_numpy()
navalmoral_de_la_mata_period = navalmoral_de_la_mata.loc[↩
    navalmoral_de_la_mata["FECHA"].between(start, end)][parametro].↩
    to_numpy()
bilbao_aeropuerto_period = bilbao_aeropuerto.loc[bilbao_aeropuerto["↩
    FECHA"].between(start, end)][parametro].to_numpy()
```

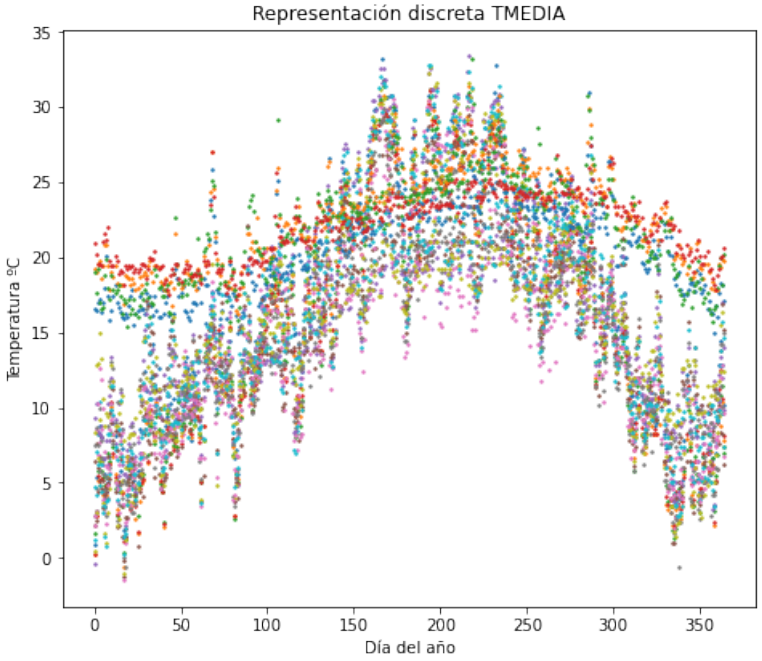
Para poder utilizar las funciones que ofrece Python para el tratamiento de datos funcionales es necesario tener una matriz de tipo *array* y para ello previamente hay que concatenar todos los datos.

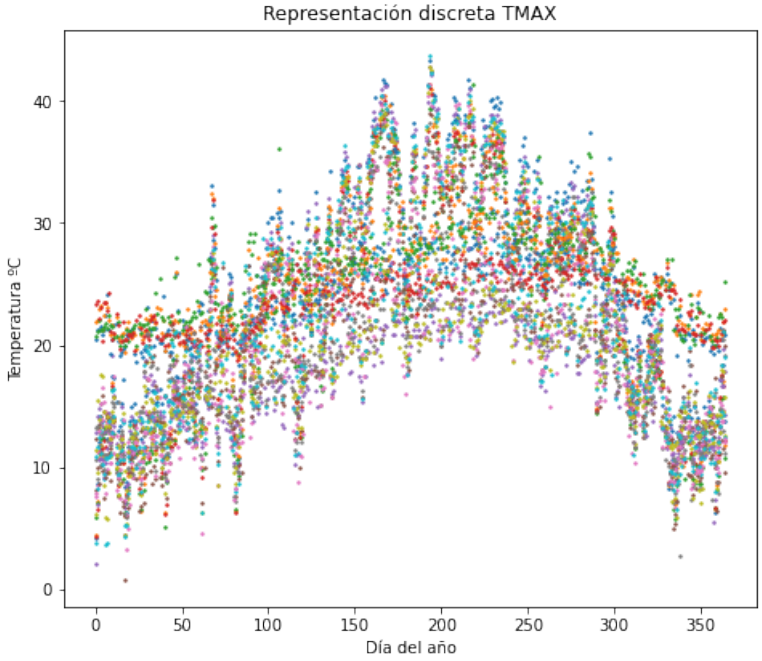
```
# Agrupacion de datos
data = np.concatenate(([navalmoral_de_la_mata_period], [↩
    torrejon_de_ardoz_period], [guadalajara_period], [↩
    madrid_aeropuerto_period], [toledo_period], [cuenca_period], [↩
    albacete_period], [salamanca_period], [valdepenas_period], [↩
    ciudad_real_period],
[guimar_period], [santa_cruz_de_tenerife_period], [↩
    lanzarote_aeropuerto_period], [hierro_aeropuerto_period],
[santander_period], [zumaia_period], [santiago_aeropuerto_period], [↩
    santander_aeropuerto_period], [acoruna_period], [↩
    bilbao_aeropuerto_period]), axis=0)

# Construccion elemento funcional
# Definicion de dominio y objeto
grid_points = range(0,data.shape[1])
fd = fda.FDataGrid(data_matrix=data, grid_points=grid_points, ↩
    axes_labels=['Dia del anio', 'Temperatura C'])
```

Se grafican los datos discretos originales para conocer cómo se distribuyen.

```
fig_discreta = plt.figure(figsize=(6,5))
fig_funcional = plt.figure(figsize=(6,5))

grid_points = range(0,data.shape[1])
fd = fda.FDataGrid(data_matrix=data, grid_points=grid_points,   ↩
    axes_labels=['Dia del anio', 'Temperatura C'])
fd.plot(fig_funcional)
fd.scatter(fig_discreta, s=2)
```


Representación discreta TMIN
Temperatura ºC
Día del año
25
20
15
10
5
0
−5
−10
0
50
100
150
200
250
300
350


Representación discreta TMEDIA
Temperatura ºC
Día del año
35
30
25
20
15
10
5
0
0
50
100
150
200
250
300
350




Se realiza la representación de las temperaturas de cada estación para cada variable etiquetas con el clima correspondiente de acuerdo a su situación geográfica.

```
# Representacion de los datos originales en forma discreta con la ↩
    clasificacion por situacion geografica
data1 = albacete_period
data2 = toledo_period
data3 = ciudad_real_period
data4 = valdepenas_period
data5 = navalmoral_de_la_mata_period
data6 = guadalajara_period
data7 = torrejon_de_ardoz_period
data8 = madrid_aeropuerto_period
data9 = salamanca_period
data10 = cuenca_period
data11 = guimar_period
data12 = lanzarote_aeropuerto_period
data13 = hierro_aeropuerto_period
data14 = santa_cruz_de_tenerife_period
data15 = bilbao_aeropuerto_period
data16 = santiago_aeropuerto_period
data17 = zumaia_period
data18 = acoruna_period
data19 = santander_aeropuerto_period
data20 = santander_period

fig = plt.figure(figsize=(6,5))
ax1 = fig.add_axes([0,0,1,1])
ax1.set_title('Curvas clasificacion original' + ' ' + str(parametro))
ax1.set_xlabel('Dia del anio')
ax1.set_ylabel('Temperatura C')
```

```
grid_points = range(0,data1.shape[0])

fd1 = fda.FDataGrid(data_matrix=data1, grid_points=grid_points)
fd2 = fda.FDataGrid(data_matrix=data2, grid_points=grid_points)
fd3 = fda.FDataGrid(data_matrix=data3, grid_points=grid_points)
fd4 = fda.FDataGrid(data_matrix=data4, grid_points=grid_points)
fd5 = fda.FDataGrid(data_matrix=data5, grid_points=grid_points)
fd6 = fda.FDataGrid(data_matrix=data6, grid_points=grid_points)
fd7 = fda.FDataGrid(data_matrix=data7, grid_points=grid_points)
fd8 = fda.FDataGrid(data_matrix=data8, grid_points=grid_points)
fd9= fda.FDataGrid(data_matrix=data9, grid_points=grid_points)
fd10 = fda.FDataGrid(data_matrix=data10, grid_points=grid_points)
fd11 = fda.FDataGrid(data_matrix=data11, grid_points=grid_points)
fd12 = fda.FDataGrid(data_matrix=data12, grid_points=grid_points)
fd13 = fda.FDataGrid(data_matrix=data13, grid_points=grid_points)
fd14 = fda.FDataGrid(data_matrix=data14, grid_points=grid_points)
fd15 = fda.FDataGrid(data_matrix=data15, grid_points=grid_points)
fd16 = fda.FDataGrid(data_matrix=data16, grid_points=grid_points)
fd17 = fda.FDataGrid(data_matrix=data17, grid_points=grid_points)
fd18 = fda.FDataGrid(data_matrix=data18, grid_points=grid_points)
fd19= fda.FDataGrid(data_matrix=data19, grid_points=grid_points)
fd20 = fda.FDataGrid(data_matrix=data20, grid_points=grid_points)


fd1.plot(fig, color='forestgreen', label="Interior Mediterraneo")
fd2.plot(fig, color='forestgreen')
fd3.plot(fig, color='forestgreen')
fd4.plot(fig, color='forestgreen')
fd5.plot(fig, color='forestgreen')
fd6.plot(fig, color='forestgreen')
fd7.plot(fig, color='forestgreen')
fd8.plot(fig, color='forestgreen')
fd9.plot(fig, color='forestgreen')
fd10.plot(fig, color='forestgreen')
fd11.plot(fig, color='deepskyblue')
fd12.plot(fig, color='deepskyblue')
fd13.plot(fig, color='deepskyblue')
fd14.plot(fig, color='deepskyblue', label="Subtropical")
fd15.plot(fig, color='orange', label="Oceanico")
fd16.plot(fig, color='orange')
fd17.plot(fig, color='orange')
fd18.plot(fig, color='orange')
fd19.plot(fig, color='orange')
fd20.plot(fig, color='orange')

leg = ax1.legend()
```

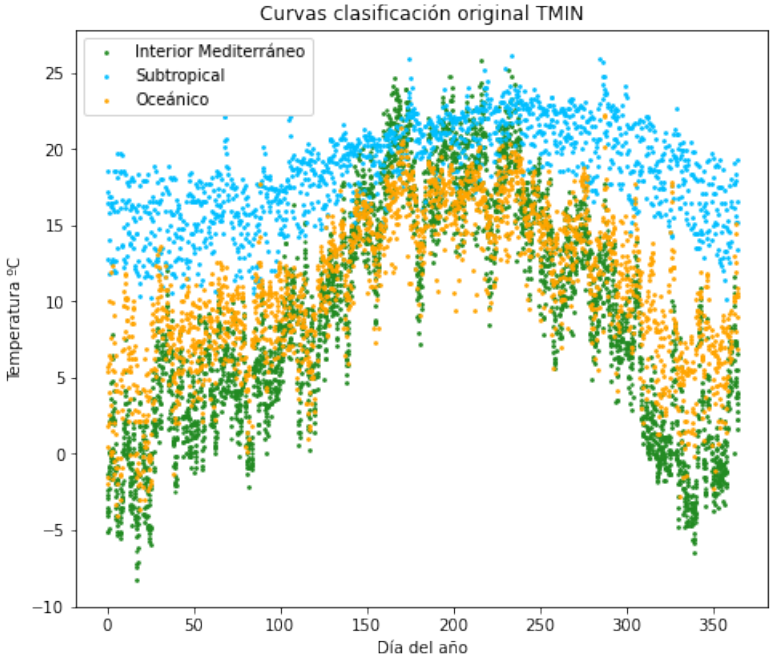



Figura A.1: Temperaturas mínimas. Fuente: elaboración propia.

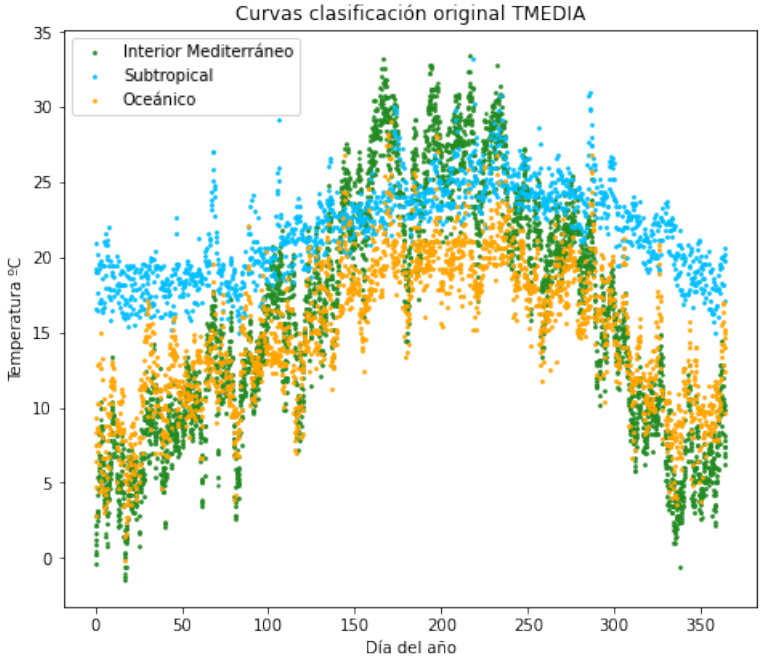



Figura A.2: Temperaturas medias. Fuente: elaboración propia.

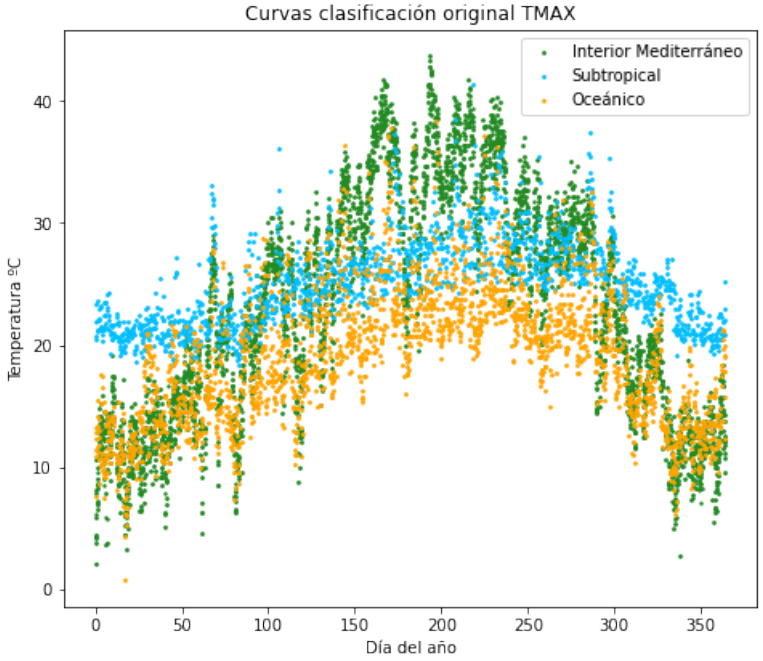
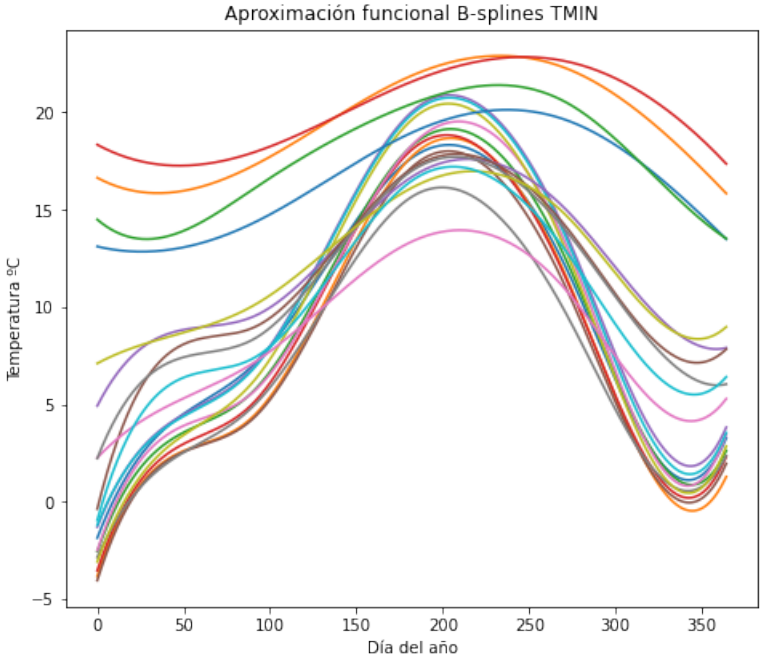
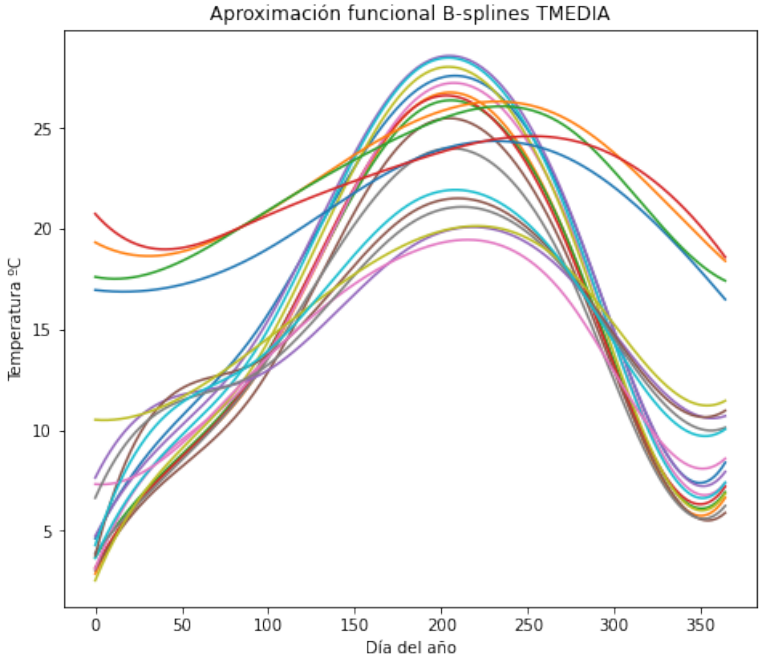


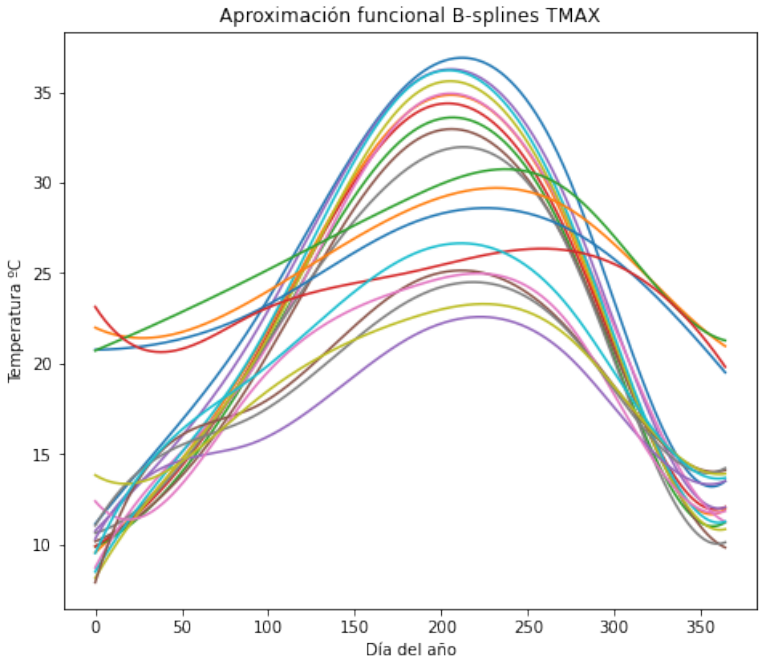

Figura A.3: Temperaturas máximas. Fuente: elaboración propia.

Aproximación de los datos mediante una base de funciones de B-Splines truncada a siete términos.

```
# Aproximacion de los datos mediante base funcional B-splines
# Representacion truncada a 7 terminos de base
basis1 = fda.representation.basis.BSpline(n_basis = 7)
basis1_fd = fd.to_basis(basis1)
basis1_fd.plot(fig_basis1)
```


Aproximación funcional B-splines TMIN
Temperatura ºC
Día del año
20
15
10
5
0
−5
0
50
100
150
200
250
300
350

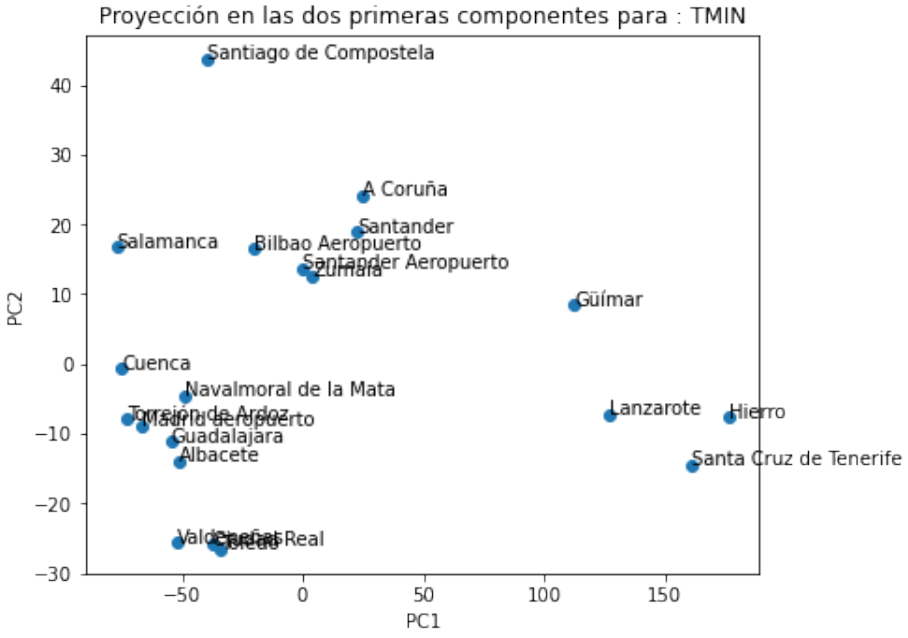

Aproximación funcional B-splines TMEDIA
Temperatura ºC
Día del año
25
20
15
10
5
0
50
100
150
200
250
300
350

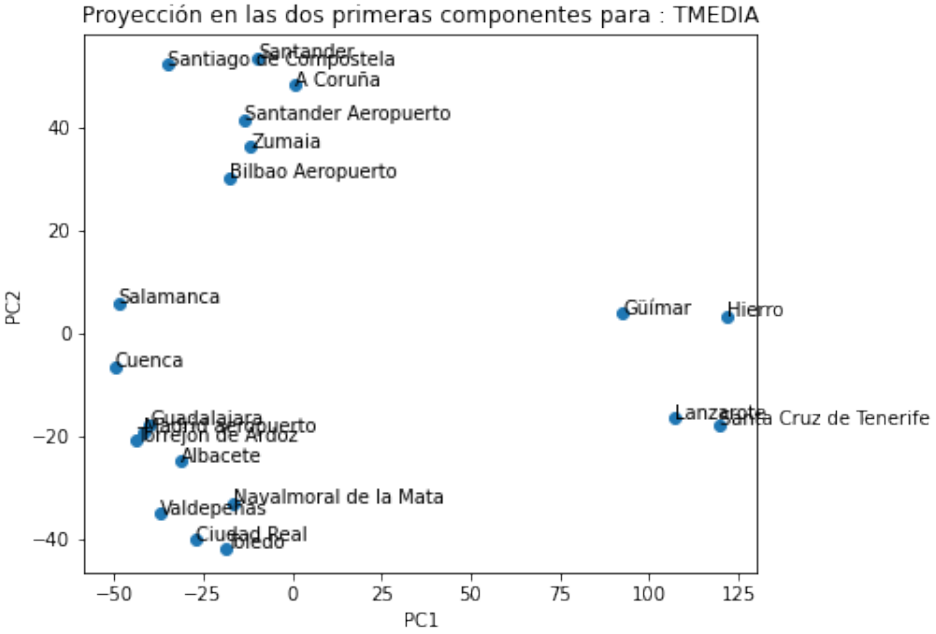


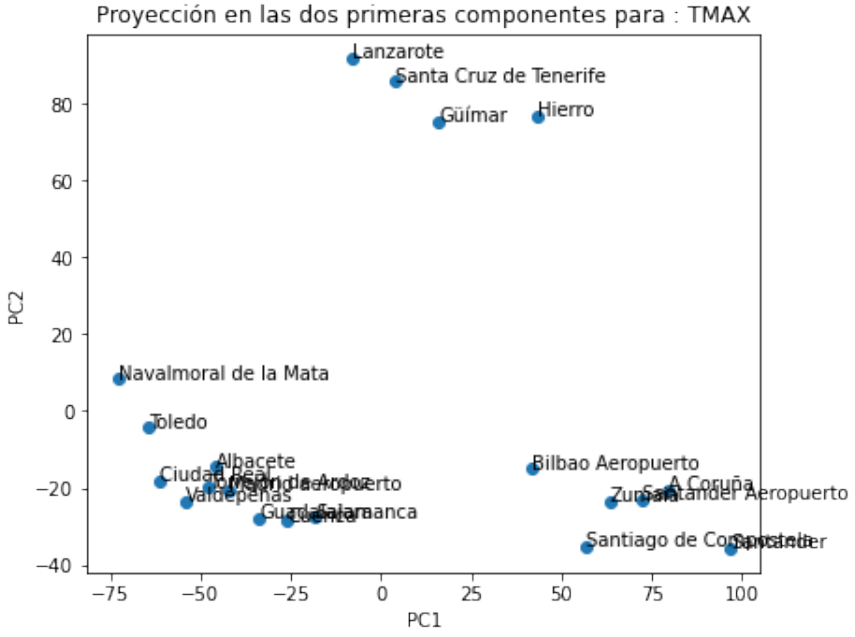

Se realiza el análisis de componentes principales funcionales y para ello se calculan las componentes principales, la proyección de los datos en las dos primeras componentes y la variabilidad acumulada por las componentes principales. Se representa tanto la proyección como los histogramas en los que se puede analizar el porcentaje de variabilidad acumulada por las componentes principales.

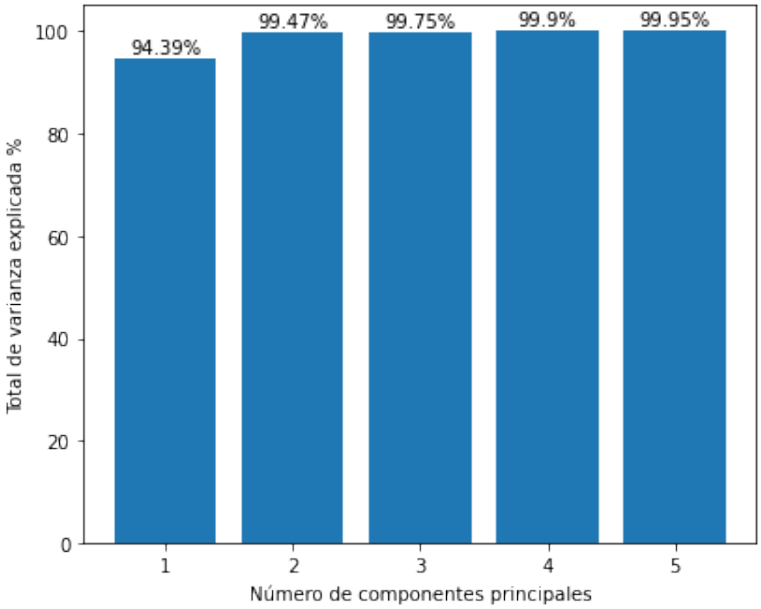
```
# Calculo de las componentes principales
num_components = []
total_varianza_explicada = []
for N_components in range(1, 6):
    fpca = fda.preprocessing.dim_reduction.projection.FPCA(n_components ↩
        = N_components)
    fpca.fit(basis1_fd)
    num_components.append(str(N_components))
    total_varianza_explicada.append(round(fpca.explained_variance_ratio_↩
        .sum()  100,2))

# Proyeccion de los datos en las dos primeras componentes principales
fig_proyeccion = plt.figure(figsize=(5,4))
ax = fig_proyeccion.add_axes([0,0,1,1])

basis = fda.representation.basis.BSpline(n_basis=7)
basis_fd = fd.to_basis(basis)
fpca = fda.preprocessing.dim_reduction.projection.FPCA(n_components=2)
fpca.fit(basis_fd)

transform = fpca.fit_transform(basis_fd)
ax.scatter(transform[:, 0], transform[:, 1])
ax.set_xlabel('PC1')
```

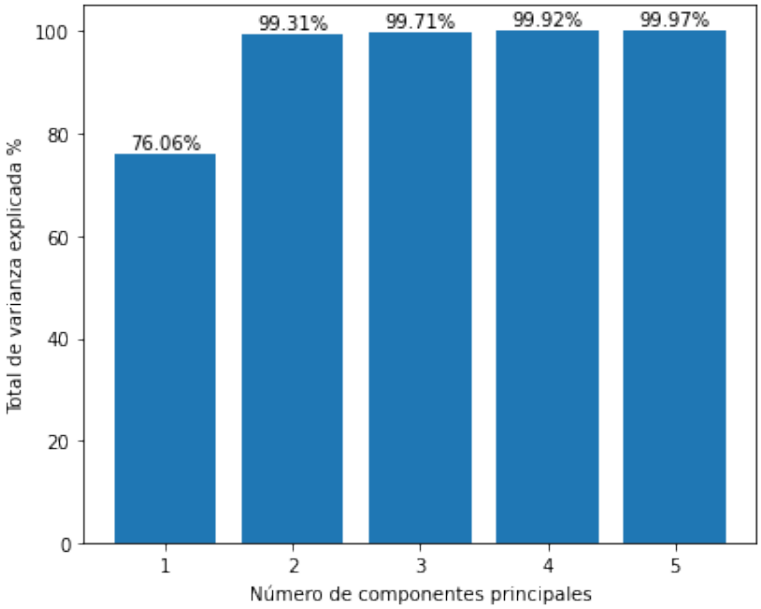
```
ax.set_ylabel('PC2')
ax.set_title('Proyecciun en las dos primeras componentes para :' + ' ' +↩
     str(parametro))

estaciones = ('Navalmoral de la Mata', 'Torrejon de Ardoz', 'Guadalajara↩
    ', 'Madrid aeropuerto', 'Toledo',   'Cuenca', 'Albacete', 'Salamanca↩
    ', 'Valdepenas', 'Ciudad Real', 'Guimar', 'Santa Cruz de Tenerife', ↩
    'Lanzarote', 'Hierro', 'Santander', 'Zumaia', 'Santiago de ↩
    Compostela', 'Santander Aeropuerto', 'A Coruna', 'Bilbao Aeropuerto'↩
    )

for i, etiqueta in enumerate(estaciones):
plt.annotate(etiqueta, (transform[i, 0], transform[i, 1]))

# Histograma de variabilidad de componentes
fig_varianza = plt.figure(figsize=(5,4))
ax = fig_varianza.add_axes([0,0,1,1])

ax.bar(num_components, total_varianza_explicada)
ax.set_ylabel('Total de varianza explicada %')
ax.set_xlabel('Numero de componentes principales')

for i, etiqueta in enumerate(total_varianza_explicada):
plt.annotate(str(etiqueta) + '%', (num_components[i], ↩
    total_varianza_explicada[i] + 1), ha='center')
```

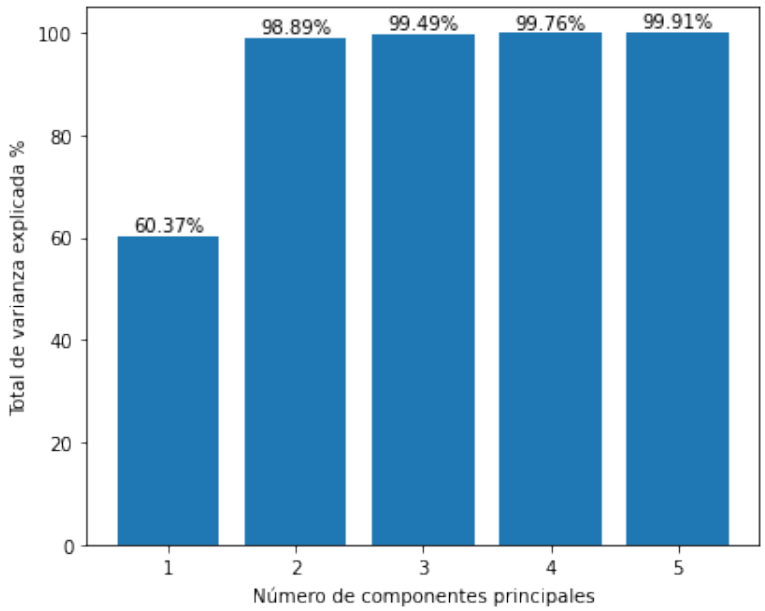



Proyección en las dos primeras componentes para : TMEDIA
Santander
Santiago de Compostela
A Coruña
Santander Aeropuerto
Zumaia
Bilbao Aeropuerto
Salamanca
Güímar
Hierro
Cuenca
Lanzarote
Santa Cruz de Tenerife
Guadalajara
Madrid aeropuerto
Torrejón de Ardoz
Albacete
Navalmoral de la Mata
Valdepeñas
Ciudad Real
Toledo
PC1
PC2


Proyección en las dos primeras componentes para : TMAX
Lanzarote
Santa Cruz de Tenerife
Güímar
Hierro
Navalmoral de la Mata
Toledo
Albacete
Bilbao Aeropuerto
A Coruña
Santander Aeropuerto
Santiago de Compostela
PC1
PC2


94.39%
99.47%
99.75%
99.9%
99.95%
Total de varianza explicada %
Número de componentes principales

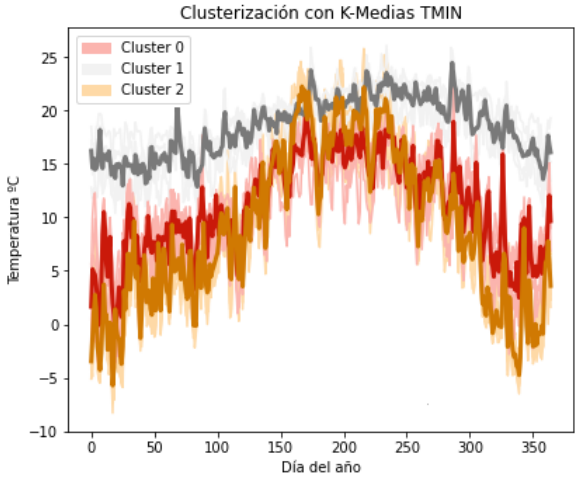

76.06%
99.31%
99.71%
99.92%
99.97%
Total de varianza explicada %
Número de componentes principales

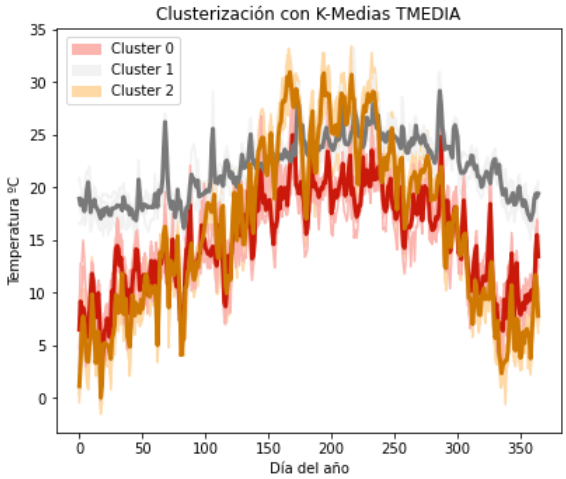


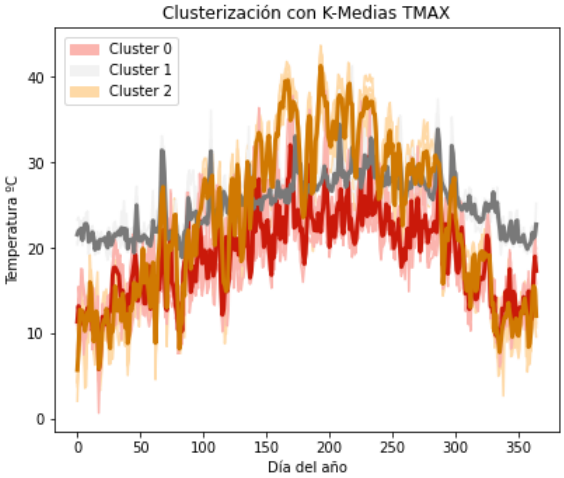

Posteriormente se procede al análisis cluster de los datos. Se utiliza el método K-medias ofrecido por Python. Con la información obtenida del análisis de componentes principales, se efectúa una única iteración en busca de los tres grupos climatológicos.

```
# Cluster K-Medias sobre los datos funcionales

n_clusters = 3
# Requiere establecer la semilla para que para todas las iteraciones se ↩
    obtenga el mismo resulatdo
kmeans = fda.ml.clustering.KMeans(n_clusters=n_clusters, random_state↩
    =10)
fd_cluster = fda.FDataGrid(data_matrix=data, grid_points=grid_points)
kmeans.fit(fd_cluster)

clasificacion_climas = ['Interior Mediterraneo', 'Interior Mediterraneo'↩
    , 'Interior Mediterraneo', 'Interior Mediterraneo','Interior ↩
    Mediterraneo','Interior Mediterraneo','Interior Mediterraneo','↩
    Interior Mediterraneo','Interior Mediterraneo', 'Interior ↩
    Mediterraneo', 'Subtropical', 'Subtropical','Subtropical', '↩
    Subtropical', 'Oceanico', 'Oceanico', 'Oceanico', 'Oceanico', '↩
    Oceanico','Oceanico']

colormap = plt.cm.get_cmap('Pastel1')
n_climates = 3
climate_colors = colormap(np.arange(n_climates) / (n_climates - 1))

cluster_colors = climate_colors[np.array([0, 2, 1])]
cluster_labels = ['Cluster 0 = Oceanico', 'Cluster 1 = Subtropical','↩
    Cluster 2 = Interior Mediterraneo']

fig_cluster = plt.figure(figsize=(5,4))
ax1 = fig_cluster.add_axes([0,0,1,1])
```

```
fda.exploratory.visualization.clustering.plot_clusters(kmeans, ↩
    fd_cluster, fig=fig_cluster, cluster_colors=cluster_colors,
cluster_labels=cluster_labels)
ax1.set_title('Clusterizacion con K-Medias' + ' ' + str(parametro))
ax1.set_xlabel('Dia del anioo')
ax1.set_ylabel('Temperatura C')
```






Aunque en los gráficos anteriores se intuye que la clusterización ha sido exitosa, por último se ha obtenido la tabla de resultados en la que se detalla el nombre de la estación meteorológica, el clima esperado y la predicción proporcionada por el modelo de clusterización.

```
# Tabla detallada con los resultados del Analisis de cluster
cluster = []
for index in range(len(clasificacion_climas)):
    cluster.append([estaciones[index], clasificacion_climas[index], ↩
        kmeans.predict(fd)[index]])

data_df = pd.DataFrame(cluster, columns = ['Estacion Meteorologica', '↩
    Tipo de clima original', 'CLUSTER'])
print(data_df)
```

| | Estación Meteorológica | Tipo de clima original | CLUSTER |
|---|---|---|---|
| 0 | Navalmoral de la Mata | Interior Mediterráneo | 2 |
| 1 | Torrejón de Ardoz | Interior Mediterráneo | 2 |
| 2 | Guadalajara | Interior Mediterráneo | 2 |
| 3 | Madrid aeropuerto | Interior Mediterráneo | 2 |
| 4 | Toledo | Interior Mediterráneo | 2 |
| 5 | Cuenca | Interior Mediterráneo | 2 |
| 6 | Albacete | Interior Mediterráneo | 2 |
| 7 | Salamanca | Interior Mediterráneo | 2 |
| 8 | Valdepeñas | Interior Mediterráneo | 2 |
| 9 | Ciudad Real | Interior Mediterráneo | 2 |
| 10 | Güímar | Subtropical | 1 |
| 11 | Santa Cruz de Tenerife | Subtropical | 1 |
| 12 | Lanzarote | Subtropical | 1 |
| 13 | Hierro | Subtropical | 1 |
| 14 | Santander | Oceánico | 0 |
| 15 | Zumaia | Oceánico | 0 |
| 16 | Santiago de Compostela | Oceánico | 0 |
| 17 | Santander Aeropuerto | Oceánico | 0 |
| 18 | A Coruña | Oceánico | 0 |
| 19 | Bilbao Aeropuerto | Oceánico | 0 |

# Apéndice B

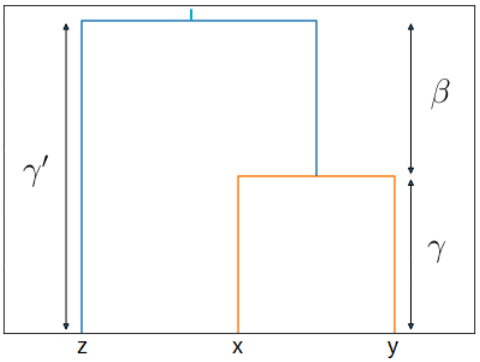
# Figuras

## B.1. Figuras del capítulo 1

Código de programación de la figura del teorema 1.4.1

```
# Importar librerias
import numpy as np
from scipy.cluster.hierarchy import dendrogram, linkage
from matplotlib import pyplot as plt

# Generar datos aleatorios
randomMatrix = np.array([[5],[10],[15]])

# Definir proceso de union
linked = linkage(randomMatrix, "complete")

# Representar figura
ax= plt.figure(figsize=(7, 5))
dendrogram(
linked,
orientation="top",
labels=labelList,
distance_sort="ascending",
show_leaf_counts=True,
get_leaves=True,
leaf_font_size=16
)

# Ocultar etiquetas ejes
plt.yticks(color='w')
plt.xticks(color='w')

# Generar flechas
plt.annotate(text='',xy=(26, 5), xycoords='data', xytext=(26, 10),textcoords↩
    ='data',
arrowprops=dict(arrowstyle="<|-|>"),va='center')
plt.annotate(text='',xy=(4, 0), xycoords='data', xytext=(4, 10),textcoords='↩
    data',
```

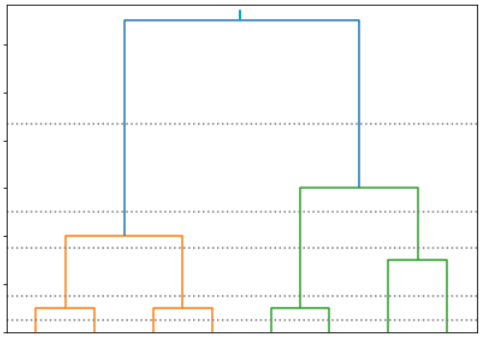
```
arrowprops=dict(arrowstyle="<|-|>"),va='center')
plt.annotate(text='',xy=(26, 0), xycoords='data', xytext=(26, 5),textcoords=↩
    'data',
arrowprops=dict(arrowstyle="<|-|>"),va='center')

# Mostrar figura
plt.show()
```




Código de programación de la subsección 1.4.2:

```
# Importar librerias
import numpy as np
from scipy.cluster.hierarchy import dendrogram, linkage
from matplotlib import pyplot as plt

# Generar datos aleatorios
randomMatrix = np.array([[5],[10],[15]])

# Definir proceso de union
linked = linkage(randomMatrix, "complete")

# Representar figura
ax= plt.figure(figsize=(7, 5))
dendrogram(
linked,
orientation="top",
labels=labelList,
distance_sort="ascending",
show_leaf_counts=True,
get_leaves=True,
leaf_font_size=16
)

# Ocultar etiquetas ejes
plt.yticks(color='w')
plt.xticks(color='w')

# Generar flechas
plt.annotate(text='',xy=(26, 5), xycoords='data', xytext=(26, 10),textcoords↩
    ='data',
arrowprops=dict(arrowstyle="<|-|>"),va='center')
```

```
plt.annotate(text='',xy=(4, 0), xycoords='data', xytext=(4, 10),textcoords='↩
    data',
arrowprops=dict(arrowstyle="<|-|>"),va='center')
plt.annotate(text='',xy=(26, 0), xycoords='data', xytext=(26, 5),textcoords=↩
    'data',
arrowprops=dict(arrowstyle="<|-|>"),va='center')

# Mostrar figura
plt.show()
```



Algoritmo de K-Medias propuesto por McQueen [13]:

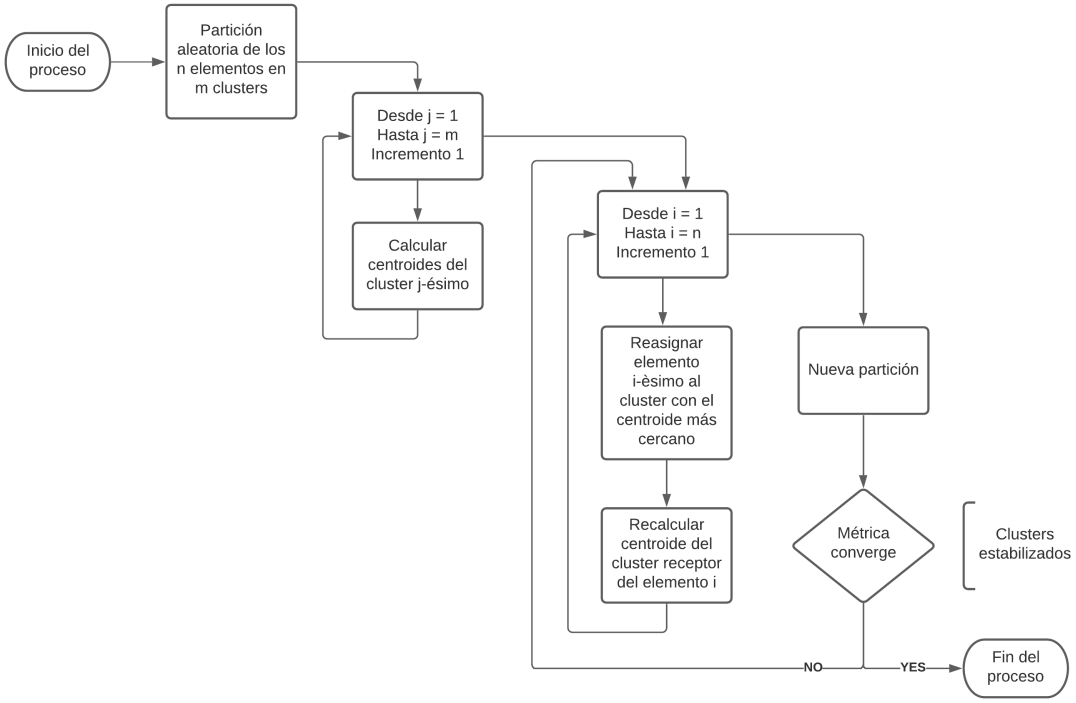



Figura B.1: Algoritmo K-Medias McQueen. Fuente: elaboración propia.

Algoritmo de K-Medias propuesto por Forgy [5]:

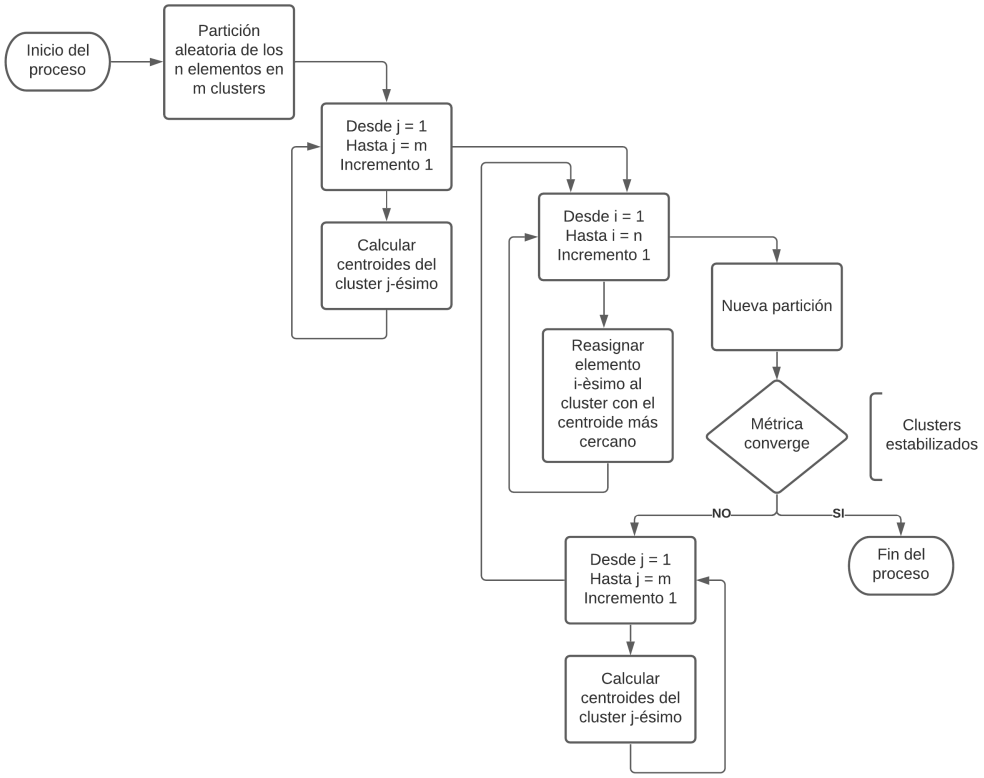
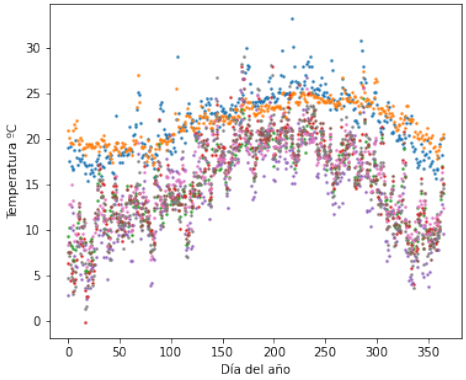


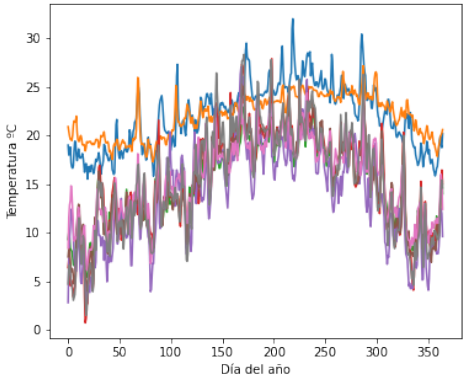

Figura B.2: Algoritmo K-Medias Forgy. Fuente: elaboración propia.

## B.2. Figuras del capítulo 2

Código de programación de la subsección 2.2:

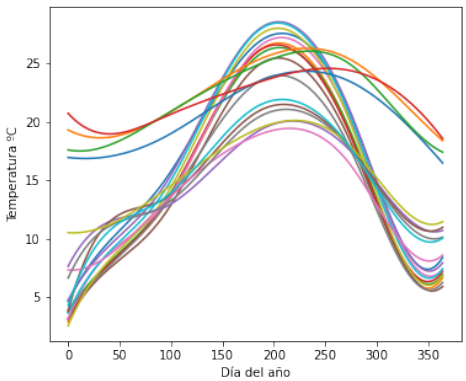
```
data = np.concatenate(([lanzarote_aeropuerto_period], [↩
    hierro_aeropuerto_period],
[santander_period], [zumaia_period], [santiago_aeropuerto_period], [↩
    santander_aeropuerto_period], [acoruna_period], [↩
    bilbao_aeropuerto_period]), axis=0)

fig_discreta = plt.figure(figsize=(6,5))
fig_funcional = plt.figure(figsize=(6,5))

grid_points = range(0,data.shape[1])
fd = fda.FDataGrid(data_matrix=data, grid_points=grid_points, axes_labels=['↩
    Dia del anio', 'Temperatura C'])
fd.plot(fig_funcional)
fd.scatter(fig_discreta, s=2)
```

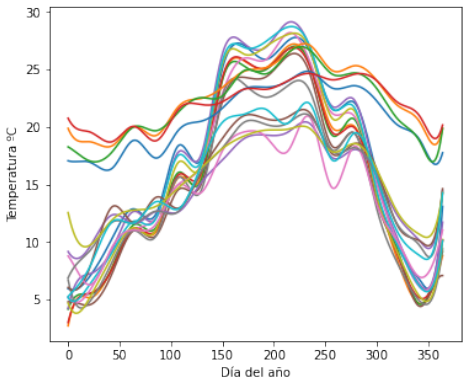



Código de programación de la subsección 2.2.1:

```
# Creacion de las figuras para la representacion
fig_basis1 = plt.figure(figsize=(6,5))
fig_basis2 = plt.figure(figsize=(6,5))

# Aproximacion de los datos mediante base funcional B-splines
# Representacion truncada a 7 terminos de base
basis1 = fda.representation.basis.BSpline(n_basis = 7)
basis1_fd = fd.to_basis(basis1)
basis1_fd.plot(fig_basis1)

# Representacion truncada a 20 terminos de base
basis2 = fda.representation.basis.BSpline(n_basis = 20)
basis2_fd = fd.to_basis(basis2)
basis2_fd.plot(fig_basis2)
```



# Bibliografía


[1] AEMET, *Descarga datos fuente aemet - open data - (291 estaciones)*, https://datosclima.es/Aemethistorico/Descargahistorico.html, Accedido 01-08-2021.

[2] G. Young C. Eckart, *The approximation of one matrix by another of lower rank*, Psychometrika **1** (1936).

[3] Carles M. Cuadras, *Nuevos métodos de análisis multivariante*, CMC Editions, 2012.

[4] José Javier Martínez Fernández de las Heras, *La descomposición en valores singulares (svd) y algunas de sus aplicaciones*, La Gaceta de la RSME **8.3** (2005), no. 4.

[5] Forgy, *Cluster analysis of multivariate data: Efficiency versus interpretability of classifications*, Biometrics (1965), no. 21.

[6] H. Hotelling, *Analysis of a complex of statistical variables into principal components*, (1933), no. 24.

[7] T. Hsing and R. Eubank, *Theoretical foundations of functional data analysis, with an introduction to linear operators*, Wiley: New York, NY, USA, 2013.

[8] Ramsay J. O. Dalzell C. J., *Some tools for functional data analysis*, Journal of the Royal Statistical Society. Series B. Methodological **53** (1991), no. 3.

[9] J. Jacques and C. Preda, *Functional data clustering: a survey*, Advances in Data Analysis and Classification **8** (2014), no. 3.

[10] R. A. Johnson and D. W. Wichern, *Applied multivariate statistical analysis, 4th ed.*, Prentice Hall, 1998.

[11] Richard Arnold Johnson, *Applied multivariate statistical analysis*, Prentice Hall, 2002.

[12] P. Kokoszka and M. Reimherr, *Introduction to functional data analysis*, Chapman and Hall: New York, NY, USA, 2017.

[13] J McQueen, *Some methods for classification and analysis of multivariate observations*, Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability (1967).

[14] Daniel Peña, *Análisis de datos multivariantes*, Mc Graw Hilll, 2002.

[15] K. Pearson, *On lines and planes of closest fit to systems of points in space*, (1901), no. 6.

[16] César Pérez, *Técnicas de análisis multivariante de datos*, Pearson Educación, 2004.

[17] J. O. Ramsay, *When the data are functions*, Psychometrika **47** (1982).

[18] James Ramsay and B. W. Silverman, *Functional data analysis, 2nd edition*, Springer-Verlag New York, 2005.

[19] Klaus Schwab, *La cuarta revolución industrial*, Debate, 2016.

[20] Hiroshi Yadohisa Shuichi Tokushige and Koichi Inada, *Crisp and fuzzy k-means clustering algorithms for multivariate functional data*, Computational Statistics **22** (2007), no. 1.

[21] G. Strang, *Linear algebra and its applications, 3rd edition*, Harcourt Brace, 1988.

[22] T. Tarpey and K.J. Kinateder, *Clustering functional data*, Journal of Classification **20** (2003), no. 1.

[23] Hastie TJ and Tibshirani RJ, *Varying-coefficient models*, Jour Roy Statist Soc B (1993), no. 55.

[24] M. Yamamoto, *Clustering of functional data in a low-dimensional subspace*, Advances in Data Analysis and Classification (2012), no. 6.