UNIVERSITAT JAUME I

Grado en Matemática Computacional

Trabajo Final de Grado

Técnicas de clasificación para datos funcionales. Aplicación a series temporales de número de positivos en Covid 19 por departamento de salud de la Comunitat Valenciana

Autor:
Juan Pons González

Tutora académica:
María Victoria Ibáñez Gual

Fecha de lectura: Julio de 2022
Curso académico 2021/2022

Resumen

En este trabajo se analizan los métodos básicos de clasificación supervisada y no supervisada para datos funcionales.

Hemos partido de una muestra de series temporales, en concreto la "Serie de casos con PDIA positiva en la Comunidad Valenciana, según la fecha en la que el laboratorio notificó el diagnóstico". A partir de ahí analizamos cómo pasar de series temporales a datos funcionales, y estudiamos las técnicas de clasificación para este tipo de datos.

En la memoria podemos diferenciar dos partes, una primera donde se explica la teoría necesaria para posteriormente aplicar los modelos de *clustering* a nuestros datos.

En la parte teórica explicamos como transformar nuestros datos en tiempos discretos a curvas, gracias a las bases de funciones. Además explicamos cómo funcionan los modelos de clasificación supervisada y no supervisada para este tipo de datos.

Finalmente, en la parte práctica, vamos a aplicar los modelos explicados a nuestros datos y además incorporaremos uno de estos modelos a un *dashboard* interactivo que está en una aplicación web.

Palabras clave

Análisis Datos Funcionales, Clasificación supervisada y no supervisada, K-NN, K-medias, Cluster Jerárquico, *R*, *Shiny*

Keywords

Functional Data Analysis, *Supervised and unsupervised Clustering*, K-NN, K-means, Hierarchical Clustering, *R*, *Shiny*

Índice general

Capítulo 1

Introducción

1.1. Contexto y motivación del proyecto

El estudio de series temporales cada vez es más habitual en diferentes ámbitos de la ciencia social y experimental, desde la meteorología o la economía hasta la medicina. Estas series temporales suelen ser recogidas en instantes de tiempo discretos y una posibilidad para poder analizarlas es aproximar cada serie temporal por una función, metiéndonos en el campo del Análisis de Datos Funcionales (FDA). Los datos funcionales surgen cuando una de las variables de interés en un conjunto de datos se puede ver de forma natural como una curva o una función suave. Así el Análisis de Datos Funcionales (Functional Data Analysis, FDA) se puede considerar como el análisis estadístico de muestras de curvas (*clustering*, predicción, regresión,...)

Un ejemplo de serie temporal, sacado del estudio de la evolución de la Covid 19, es la "Serie de casos con PDIA positiva en la Comunidad Valenciana, según fecha en la que el laboratorio notificó el diagnóstico"[13]. Esta base de datos recoge los casos positivos en Covid 19 para pruebas de detección de infección activa, *polymerase chain reaction* (PCR), y el test de antígenos siguiendo los documentos elaborados por el Ministerio de Sanidad. En el curso académico 2020/2021 participé en el programa "Estudia i investiga a l'UJI" y empezamos a realizar un breve estudio descriptivo de estos datos. Al finalizar el curso dejamos pendientes nuevas líneas de investigación entre las que se encontraba el estudio de los datos dentro del campo de FDA y explorar si hay clusters o grupos de departamentos de salud homogéneos en cuanto a la evolución de la pandemia. Por ello nos planteamos estudiar técnicas *cluster* para datos funcionales.

Los métodos *cluster* son muy habituales en el análisis de datos. Estos métodos consisten en clasificar datos siguiendo unas características parecidas para crear grupos cerrados y homogéneos. Como hemos comentado, los aplicaremos sobre la base de datos de PDIA de la

Comunidad Valenciana.

Capítulo 2

Objetivos

Partiendo del estudio iniciado de los casos positivos PDIA en la Comunidad Valenciana en el programa "Estudia i investiga a l'UJI", hemos decidido ampliarlo introduciéndonos en el Análisis de Datos Funcionales. Nuestro primer objetivo es definir y explicar cómo se deben preparar y tratar las series temporales para poder trabajar con ellas como funciones.

El segundo objetivo será estudiar cómo analizar estadísticamente muestras de funciones, empezando por ver cómo se definen los estadísticos descriptivos elementales.

Por otra parte, los datos están recogidos tanto para las diferentes provincias como para los diferentes departamentos de salud. El último objetivo del trabajo será estudiar métodos *cluster* supervisados y no supervisados para datos funcionales, y aplicarlos para encontrar grupos de departamentos de salud con patrones similares en la evolución del Covid.

Capítulo 3

Desarrollo del TFG

3.1. De series temporales a datos funcionales

Definición 1 *Dado un espacio de probabilidad (ω, A, P), una variable funcional es una variable aleatoria X, la cual toma valores en un espacio de dimensión infinita F (o espacio funcional). Cada una de las observaciones se llama dato funcional.*

De forma general, al hablar de series temporales (en nuestro caso de datos longitudinales), solemos hablar en el ámbito científico de observaciones tomadas en ciertos instantes de tiempo, es decir, datos en tiempos discretos. Por tanto, nuestros datos son:

$$\{x_i(t) \ : \ t \in T \ , \ i = 1, 2, ..., N\} \tag{3.1}$$

donde, $T = \{t_0, t_1, t_2, \cdots, \}$ es el intervalo en el tiempo, t el número de instantes en el que tenemos información y i indica sobre qué variable o individuo se está trabajando (en nuestro caso será sobre qué departamento de salud).

Al conjunto que forman los datos de cada individuo/departamento de salud, se les puede ver como una función a lo largo del tiempo. A este tipo de datos se les llama datos funcionales.

Una de las condiciones para poder realizar estudios con datos funcionales es que las funciones sean suaves, es decir, que la probabilidad de que los datos adyacentes a $x_i(t)$, $x_i(t-1)$ y $x_i(t+1)$, sean muy diferentes es muy baja. Por otra parte usamos el término suavidad para denotar la existencia de que al menos posee la primera derivada, pero la posible existencia de ruidos o errores pueden causar que no se tenga una correcta suavidad.

Para aproximar las series temporales por funciones, se puede representar los datos respecto a una base de funciones, $\{\phi_k\}_{k\in\mathbf{N}}$, en el espacio funcional F. De esta forma tenemos que para todo $x \in F$ se expresa como una combinación lineal finita de la siguiente forma:

$$x(t) \approx \hat{x}(t) = \sum_{k=1}^{K} c_k \phi_k(t) \quad : \quad t \in T \tag{3.2}$$

donde c_k son los coeficientes que ajustarán nuestra base a los datos y K es el número de elementos que posee nuestra base.
Se puede expresar de manera matricial si tomamos ϕ como el vector de funciones ϕ_k y c como el vector con los coeficientes c_k, por tanto obtenemos:

$$\hat{x} = c^t \phi = \phi^t c \tag{3.3}$$

Para construir las bases de funciones no solo se debe tener en cuenta como son los datos de trabajo, además se deben valorar el tipo de funciones con las que trabajar para obtener una correcta representación de los datos. Como consecuencia existe gran variedad de bases de funciones como son las bases constantes ($\phi(t) = cte$), polinómicas ($1, x, x^2, x^3, ...$), exponenciales ($e^{\lambda_1 t}, e^{\lambda_2 t}, e^{\lambda_3 t}, ...$), de *Fourier*, B-splines, etc... Una buena elección del tipo de base no solo beneficiará a una correcta aproximación sino que además aumentará el rendimiento computacional.

3.1.1. Suavizado de datos

Antes de comentar algunas bases de forma más extendida debemos ver a qué hace referencia el término suavizamiento.

Suavizar datos se refiere a aplicar una técnica para eliminar ruidos o comportamientos no esperados en los datos, además de detectar datos atípicos. A continuación vamos a mostrar algunas técnicas.

Métodos de ventana móvil

El método ventana móvil o medias móviles recibe este nombre debido a la forma de procesar los datos, en concreto consiste en analizar los datos por subconjuntos. Cada uno de los subconjuntos recibe el nombre de ventana y se le añade el término móvil puesto que el algoritmo analiza las diferentes ventanas de forma consecutiva.

El funcionamiento de esta técnica parte de determinar el número impar, V, de datos que forman una ventana. Una vez definida la ventana se toma el primer dato y se determina su

ventana, donde calcula la media de dicho subgrupo.

$$\tilde{x}_i = \frac{1}{2V+1} \sum_{j=-V}^{V} x_{i+j}(t) \tag{3.4}$$

Dicho valor sustituye al valor fijado de la ventana y seguidamente se desplaza la ventana fijando el siguiente dato.

Esta técnica tiene varios inconvenientes, uno de ellos viene definido por el tamaño de la ventana. Si la ventana es muy grande al realizar el suavizamiento quitaremos la variabilidad, mientras que si la ventana es muy pequeña no se eliminarán los datos atípicos.

Para mejorar este método se suele utilizar una variante de la ventana móvil. Esta técnica se diferencia de la anterior en la operación realizada en cada ventana. A la hora de calcular la media se realiza una media ponderada, es decir, los datos que están más cerca del dato fijado tienen mayor peso, W, que los que están más alejados. Este método recibe el nombre de ventana móvil ponderada.

$$\tilde{x}_i = \frac{1}{2V+1} \sum_{j=-V}^{V} W_j x_{i+j}(t) \quad \text{donde} \quad \sum_{j=-V}^{V} W_j = 1 \tag{3.5}$$

3.1.2. Representación mediante bases de funciones

Como ya hemos comentado, podemos representar los datos respecto a una base de funciones $\{\phi_k\}_{k\in\mathbf{N}}$ en el espacio funcional F. Las Bases de Fourier junto con las B-splines son las bases de funciones más conocidas y serán las que analizaremos en las secciones 3.1.3 y 3.1.4. Para más información sobre otras bases, sugerimos consultar [12].

Para aproximar los coeficientes de cada observación respecto a la base elegida, proponemos utilizar el método de mínimos cuadrados, que revisaremos en la sección 3.1.2. Además en la sección 3.1.5 explicamos como elegir el número de elementos que debe poseer la base.

Mínimos cuadrados ponderados

Este método es uno de los más utilizados y populares en la actualidad.

Para cada serie temporal (para cada observación) $x(t)$, fijada una base, $\phi = (\phi_1, \cdots, \phi_K)$, queremos estimar los valores de los parámetros $\{c_k\}_{k=1,\cdots,K}$ t.q. $x(t) \approx \hat{x}(t) = \sum_{k=1}^{K} c_k \phi_k$. El método de mínimos cuadrados se basa en estimar estos parámetros minimizando la suma de los

cuadrados de las diferencias entre los valores observados $x(t)$ y los valores generados $\hat{x}(t)$, i.e. se busca minimizar:

$$ECM(x|c) = (x - \phi c)^t (x - \phi c) \tag{3.6}$$

Que si derivamos respecto a c se obtiene:

$$2\phi\phi^t c - 2\phi^t x = 0 \tag{3.7}$$

Por tanto el estimador mínimos cuadrados de c es:

$$\hat{c} = (\phi\phi^t)^{-1}\phi^t x \tag{3.8}$$

Al sustituir las constantes estimadas en la ecuación 3.3 se tienen:

$$\hat{x} = \phi\hat{c} = \phi(\phi^t\phi)^{-1}\phi^t x \tag{3.9}$$

Por tanto la representación de la función en la base con los coeficientes estimados por mínimos cuadrados se expresa como:

$$\hat{x} = \hat{c}^t\phi \tag{3.10}$$

Acabamos de ver la técnica de mínimos cuadrados, pero buscamos mínimos cuadrados ponderados, para ello se necesita aportar un peso diferente a los datos. Esto se consigue de la siguiente forma:

$$ECM(x|c) = (x - \phi c)^t W (x - \phi c) \tag{3.11}$$

donde W es una matriz simétrica definida positiva. Si se conoce la matriz de varianzas-covarianzas entonces $W = \Sigma^{-1}$.
Si no se puede estimar Σ, se asume que las covarianzas son cero y por tanto W es diagonal con las varianzas. Por tanto el estimador de c por mínimos cuadrados ponderados es:

$$\hat{c} = (\phi^t W\phi)^{-1}\phi^t W x \tag{3.12}$$

3.1.3. Bases de Fourier

En el caso de las Bases de Fourier sus elementos son senos y cosenos con un periodo de oscilación, por lo que se suelen utilizar cuando los datos son periódicos.
Por tanto las funciones que se utilizan están definidas de la siguiente forma:

- $\phi_0(t) = 1$

- Para los elementos impares $\phi_{2j-1} = sen(jwt)$
- Para los elementos pares $\phi_{2j} = cos(jwt)$

donde $w = \frac{2\pi}{P}$ con periodo P.

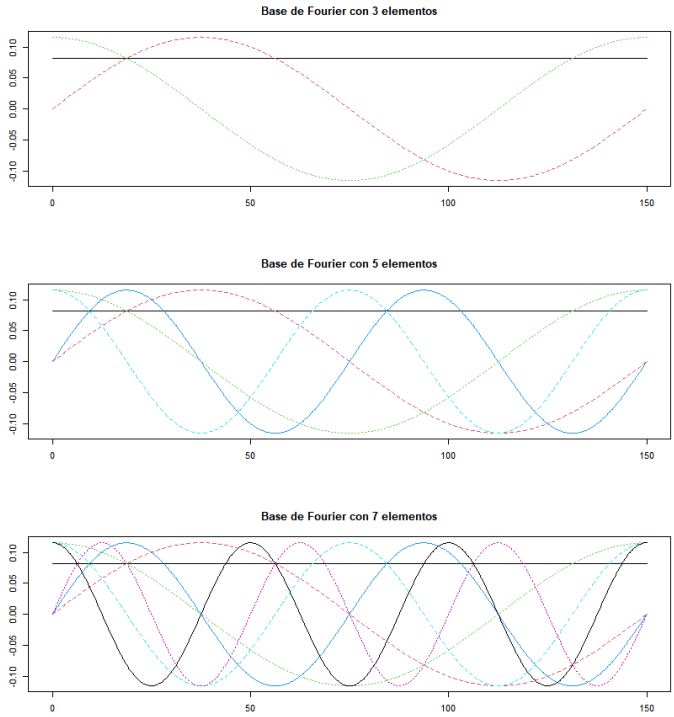


Figura 3.1: Bases de Fourier

Podemos observar en la Figura 3.1 tres ejemplos para bases con 3, 5 y 7 elementos. Se observa que todas poseen un elemento constante y las correspondientes funciones senos y cosenos. La diferencia entre las diferentes bases se encuentra en las funciones senos y cosenos, puesto que al aumentar el número de elementos estamos añadiendo mayor cantidad de senos y cosenos con su translación correspondiente.

Para representar una serie temporal respecto a una base de Fourier (Ec. 3.2), necesitaremos estimar los coeficientes por mínimos cuadrados (Ec. 3.11)

Como ejemplo, en la Figura 3.2 podemos ver como ajusta las bases de Fourier a los datos de la Comunidad Valenciana de la base de datos de la "Serie de casos con PDIA positiva en la C.V"[13]. En ella podemos observar como al aumentar el número de elementos de las bases la

suavización de los datos se ajusta más. Pero aún así, se puede apreciar que nuestras bases no forman funciones que representen correctamente los datos.

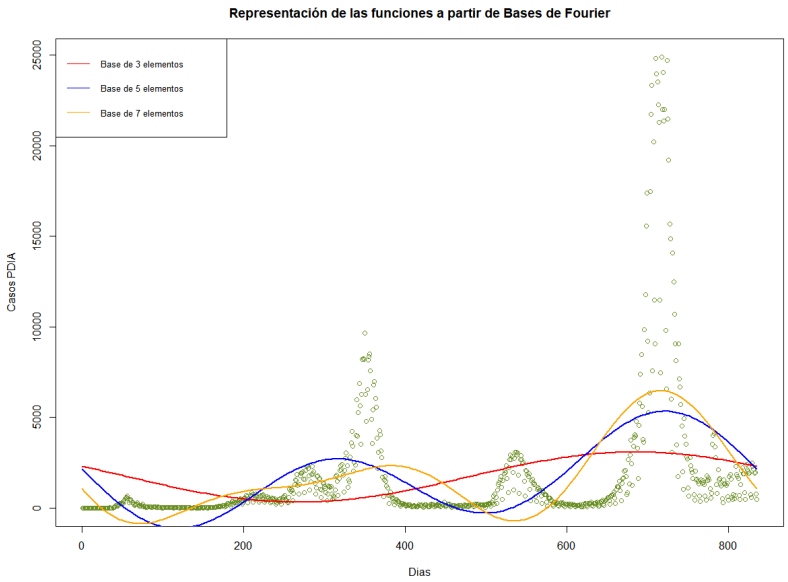


Figura 3.2: Representación de las funciones a partir de las Bases de Fourier de la Figura 3.1 de los datos de la Comunidad Valenciana de la base de datos estudiada [13].

3.1.4. Bases B-splines

Las Bases B-splines ("penalización de rugosidad") son las bases más utilizadas cuando tenemos datos no periódicos. Esto se debe a la similitud con las bases polinómicas que proporcionan una gran representación a datos no periódicos y además tienen un reducido coste computacional.

Antes de explicar qué es una B-splines vamos a definir que es un spline. Para definirlo necesitamos conocer unos términos previos.

Dada una serie temporal $x(t)$ definida en el intervalo $(0, T)$, subdividimos este intervalo en L subintervalos. Los puntos de corte que dividen el intervalo se representan como τ_l con $l = 1, 2, ..., L-1$. Diremos que cada uno de los subintervalos es un nodo y cada uno de los nodos se conecta a otro de forma suave, estos puntos se denominan nudos.

En cada uno de estos nodos, un spline es un polinomio con grado m que pasan por los puntos que las determinan. De esta forma se tienen splines lineales , con $m = 1$ que son de la

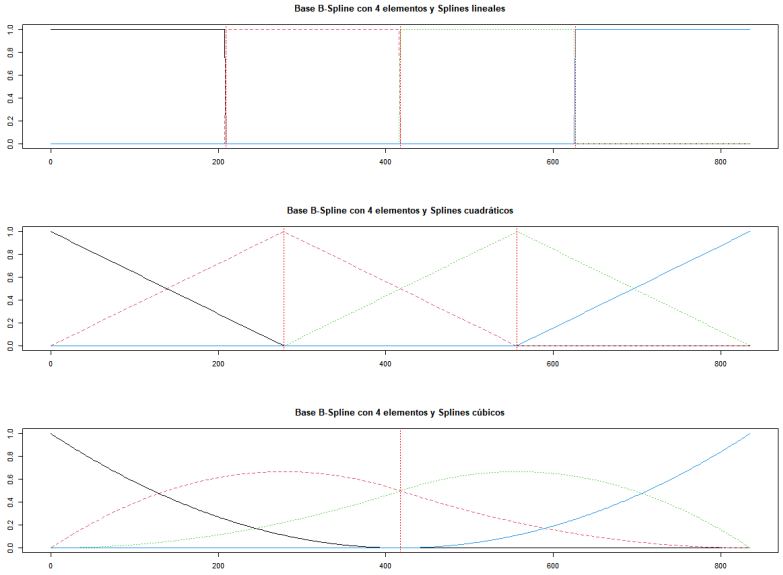


Figura 3.3: Bases B-splines con Splines Lineales, Cuadráticos y Cúbicos

forma $P(x) = ax + b$, cuadráticos, con $m = 2$ de la forma $P(x) = ax^2 + bx + c$, etc... Todos los polinomios se unen unos con otros de forma suave, esto se hace para que sean del tipo C^{m-2}. En la Figura 3.3 podemos observar la representación de B-splines lineales, cuadráticos y cúbicos. Cada una de las bases esta formada por 4 elementos, es decir por 4 splines con los órdenes comentados anteriormente.

Para mejorar las aproximaciones se puede hacer aumentando el orden de los polinomios, pero si queremos optimizar aún más estas aproximaciones se aumenta el número de nodos. El número de nodos debe aumentarse en aquellas zonas donde la variación de la función es mayor.

Una vez sabemos en qué consiste un spline podemos construir una base con dichas funciones, para ello solo necesitaremos una base que verifique lo siguiente:

- Cada $\phi_k(t)$ debe ser una función spline de orden m.
- Las combinación de funciones deben ser spline.

La forma más conocida son las B-splines, desarrolladas por Borr [5]. En ellas tenemos la función $\hat{x}(t)$ con spline de grado m y con L nodos definidos de la siguiente forma:

$$\hat{x}(t) = \sum_{k=0}^{L} c_k B_{k,m}(t) \tag{3.13}$$

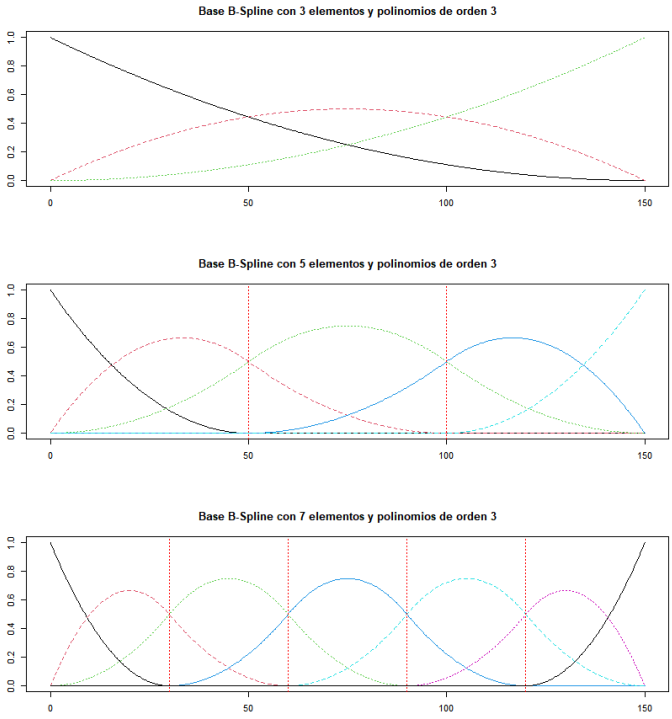


Figura 3.4: Bases B-splines

donde t son los diferentes instantes de tiempo de nuestro intervalo T a excepción de los dos extremos y donde $B_k(t)$ es el valor del B-splines en el instante t de orden m. Además c_k son los coeficientes a estimar.

Definición 2 *Fórmula de recursión Cox-de Boor*

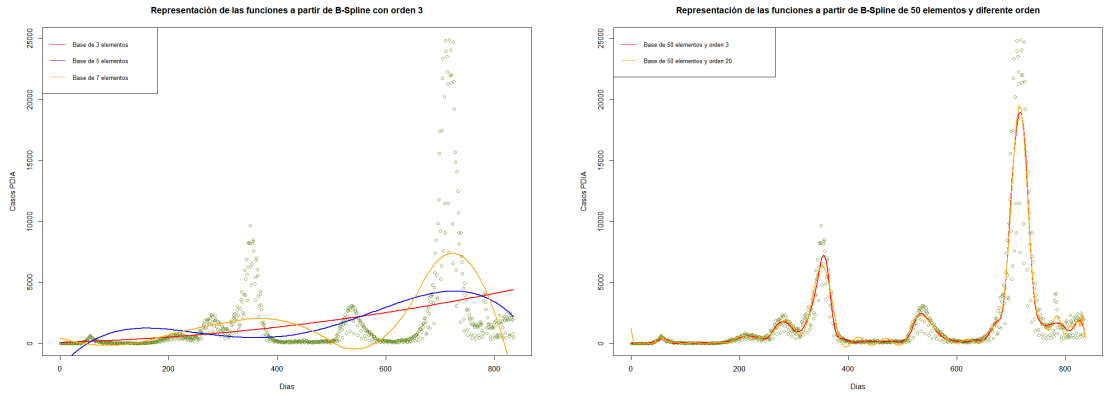
$$B_{k,0} := \begin{cases} 1 & si \quad t_k \leq t < t_{k+1} \\ 0 & resto \end{cases} \tag{3.14}$$

$$B_{k,m} := \frac{t - t_k}{t_{k+m} - t_k} B_{k,m-1}(t) + \frac{t_{k+m+1} - t}{t_{k+m+1} - t_{k+1}} B_{k+1,m-1}(t) \tag{3.15}$$

En la Figura 3.4 podemos observar la representación de 3 Bases B-splines, todas ellas de orden 3 pero con diferente número de elementos. De arriba a abajo tenemos bases con 3,5 y 7 splines respectivamente.

Por otra parte en la Figura 3.5(a) volvemos a representar los datos de la "Serie de casos con PDIA positiva en la C.V"[13], pero en este caso vamos representar las funciones que representan

los datos a partir de bases B-splines. En este caso, usaremos las bases de la Figura 3.4. Por otra parte en la Figura 3.5(b) hemos usado bases con 50 splines de orden 3 y 20 respectivamente. En esta figura apreciamos que este tipo de bases para mejorar el ajuste no solo importa el orden de los splines, sino que también se debe aumentar el número de splines.



(a) Representación de las funciones a partir de las Bases B-splines de la Figura 3.4 .

(b) Representación de las funciones a partir de Bases B-splines de 50 nodos una de orden 3 y otra de orden 20.

Figura 3.5: Representación de las funciones a partir de Bases B-splines de los datos de la Comunidad Valenciana de la base de datos estudiada [13].

3.1.5. Elección número de elementos de la base

Tras elegir el tipo de base con la que se van a representar las observaciones solo queda realizar la elección de los parámetros de la base. Por ejemplo, en las bases de Fourier se debe decidir el número de funciones que formarán la base, mientras que en las bases B-splines se debe elegir el número de funciones que formarán la base así como el orden de los spline.

Definición 3 *Se define la diferencia entre predicciones y los datos originales como el error cuadrático medio, RMSE, de la forma:*

$$RMSE = \frac{1}{T}\sum_{t=1}^{T}(\hat{x}(t) - x(t))^2 \tag{3.16}$$

De forma general queremos minimizar la cantidad de funciones K y el error cuadrático medio. Estos dos conceptos se relacionan con los grados de libertad del error, que es la cantidad

de valores iniciales necesarios para obtener el resto de forma automática, puesto que este número coincide con la cantidad de funciones.

3.2. Estadísticos descriptivos en FDA

Tomamos $t \in [0, T]$ para todos los estadísticos.

Definición 4 *Se define la función media muestral para n funciones* $x_1(t), x_2(t), ..., x_N(t)$ *como:*

$$\overline{x}(t) = \frac{1}{N} \sum_{i=1}^{N} x_i(t) \tag{3.17}$$

Definición 5 *Se define la función varianza muestral para n funciones* $x_1(t), x_2(t), ..., x_N(t)$ *como:*

$$Var_x(t) = \frac{1}{N-1} \sum_{i=1}^{N} (x_i(t) - \overline{x}(t))^2 \tag{3.18}$$

Definición 6 *Se define la función desviación estándar muestral para n funciones* $x_1(t), x_2(t), ..., x_N(t)$ *como:*

$$S_x(t) = \sqrt{Var_x(t)} \tag{3.19}$$

Definición 7 *Se define la función covarianza muestral para n funciones* $x_1(t), x_2(t), ..., x_N(t)$ *entre dos tiempos* t_0, $t_1 \in T$ *como:*

$$Cov_x(t_0, t_1) = \frac{1}{N-1} \sum_{i=1}^{N} (x_i(t_0) - \overline{x}(t_0))(x_i(t_1) - \overline{x}(t_1)) \tag{3.20}$$

Definición 8 *Se define la función correlación muestral para n funciones* $x_1(t), x_2(t), ..., x_N(t)$ *entre dos tiempos* t_0, $t_1 \in T$ *como:*

$$Corr_x(t_0, t_1) = \frac{Cov_x(t_0, t_1)}{\sqrt{Var_x(t_0) \cdot Var_x(t_1)}} = \frac{Cov_x(t_0, t_1)}{S_x(t_0) * S_x(t_1)} \tag{3.21}$$

Ahora veamos algunos estadísticos de muestras para más de una función, estas son las encargadas de estudiar la relación entre ellas. Para ello denotamos a las funciones como $x_i(t)$ respecto a la función $x(t)$ y las funciones $y_i(t)$ respecto a la función $y(t)$ con $i = 1, 2, ..., N$

Definición 9 *Se define la covarianza cruzada de $x(t)$ e $y(t)$ entre dos tiempos t_0 y $t_1 \in T$ como:*

$$Cov_{x,y}(t_0, t_1) = \frac{1}{N-1}\sum_{i=1}^{N}(x_i(t_0) - \overline{x}(t_0))(y_i(t_1) - \overline{y}(t_1)) \tag{3.22}$$

Definición 10 *Se define la función correlación cruzada de $x(t)$ e $y(t)$ entre dos tiempos t_0, $t_1 \in T$ como:*

$$Corr_{x,y}(t_0, t_1) = \frac{Cov_{x,y}(t_0, t_1)}{\sqrt{Var_x(t_0) \cdot Var_y(t_1)}} = \frac{Cov_{x,y}(t_0, t_1)}{S_x(t_0) * S_y(t_1)} \tag{3.23}$$

Finalmente vamos a comentar dos definiciones que sirven para obtener la relación entre las diferentes series temporales. En concreto vamos a comentar el variograma y el correlograma.

El variograma es una herramienta que permite estudiar cómo se comportan de forma espacial una variable sobre otra, además de la influencia de sus vecinos. Para poder definir este concepto debemos establecer unos conceptos previos.

Primero tenemos el incremento de tiempo definido entre dos instantes de tiempos como:

$$\Delta t := t_j - t_k \tag{3.24}$$

Además debemos definir la función γ:

$$\gamma(\Delta t) := \frac{1}{2}[X_i(t) - X_i(t + \Delta t)]^2 \tag{3.25}$$

Definición 11 *Se define variograma a la media de las funciones γ de las observaciones $\{X_i(t)\}_{i=1}^{N}$ para cada uno de los incrementos de tiempo, Δt, como:*

$$\hat{\gamma}(\Delta t) = promedio[\gamma(\Delta t)] \tag{3.26}$$

Por otra parte, el correlograma es una herramienta que muestra la autocorrelación. Partiendo de las funciones anteriormente definidas para el variograma podemos definir el correlgrama como:

$$correlograma := \frac{\hat{\gamma}(\Delta t)}{\gamma(0)} \tag{3.27}$$

3.3. Modelos de clasificación

En el ámbito de la estadística la clasificación consiste en dividir un conjunto de datos en varios subgrupos homogéneos de dos formas. Por una parte, cuando nuestros datos no están

agrupados de forma preestablecida y lo que se busca es definir los grupos a los que pertenecen, para ello se utilizan procedimientos de clasificación no supervisada como es el Análisis *Cluster*. La segunda forma es cuando se parte con los datos ya divididos en subgrupos y por tanto el fin del estudio es decidir a qué categoría pertenece un nuevo dato, para esto se utiliza la clasificación supervisada como es el método de los k vecinos más próximos (k-NN)o el Análisis Discriminante.

3.3.1. Modelos de clasificación supervisada

Los modelos de clasificación supervisada son frecuentes en la Inteligencia Artificial y los denominados Sistemas Inteligentes. Este tipo de modelos requieren que los datos estén previamente etiquetados, por tanto tendremos las observaciones a lo largo del tiempo, $X(t) := \{x_i(t) : t \in [0,T]\ ,\ i = 1,2,...,N\}$, y el grupo al que pertenece, $Y := \{y_i : i = 1,2,...,N\}$. Por tanto nuestros datos son el conjunto de los $(X(t), Y)$ y dada una nueva observación, el objetivo es determinar a qué grupo pertenece.

Como modelos de clasificación supervisada vamos a revisar el método de los k vecinos más próximos.

K-NN

Modelo de los K vecinos más cercanos o K-NN (K Nearest Neighbours). Es un método no paramétrico de aprendizaje supervisado que fue creado por Fix y Hodges en 1951 [7], por lo que quiere decir que los datos ya están clasificados previamente y su objetivo es clasificar los nuevos datos en el grupo correspondiente.

El funcionamiento de esta técnica consiste en clasificar cada uno de los datos en un grupo determinado, dicho grupo será al que pertenecen la mayor parte de sus k vecinos más cercanos. Para poder saber a que grupo se debe clasificar el nuevo dato se debe calcular las distancias a cada uno de los datos existentes. Una vez calculadas todas las distancias se ordenan para poder seleccionar aquellas K menores, dichas distancias proporcionan los K datos más cercanos al nuevo. En resumen, el nuevo dato se clasificará en el grupo con mayor frecuencia.

Observamos un ejemplo en la Figura 3.6, en ella tenemos un conjunto de 10 datos etiquetados en dos grupos, A y B. Se pretende clasificar un nuevo dato, marcado color rojo y para ello se toman dos casos un 3-vecinos y un 6-vecinos.
En el primer caso el nuevo dato se clasificaría en el grupo B mientras que en el segundo el nuevo dato quedaría etiquetado como A.

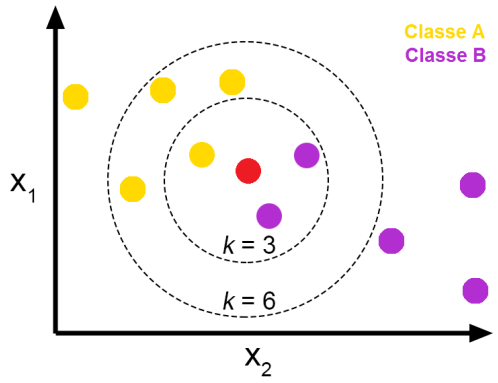
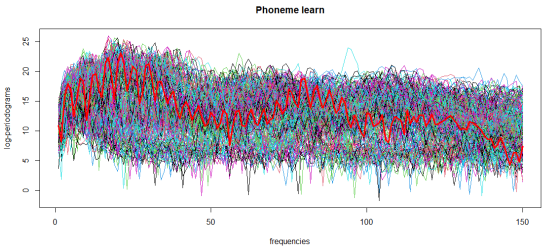


Figura 3.6: Ejemplo Clasificación K-NN

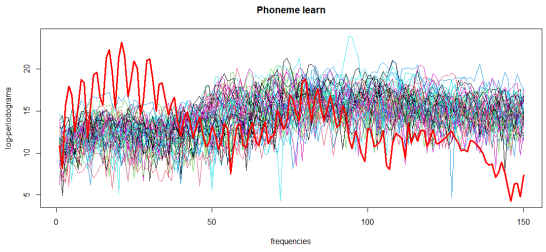
Hasta el momento hemos comentado en qué consiste el método K-NN sin tener en cuenta que vamos a tratar con datos funcionales. Para este caso la Figura 3.7 nos ayuda a entender cómo funciona el método puesto que ahora tratamos con funciones. De esta forma la distancia que utilizaremos en nuestro caso es:

$$d(x_i, x_0) = \sqrt{\sum_{t=0}^{T}(x_i(t) - x_0(t))^2} \tag{3.28}$$

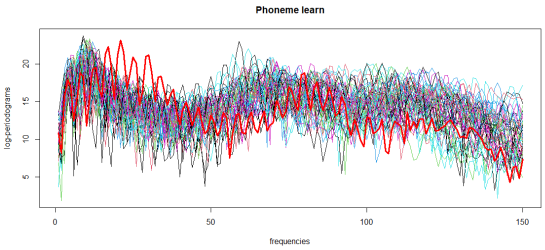
En dicha figura observamos el conjunto de datos Phoneme que esta en el paquete *fda.usc* [3] de R. Esta base de datos alamacena 250 curvas discretizadas en 150 puntos, dichas curvas están clasificadas en 5 fonemas digitales. Por tanto en la imagen superior izquierda se observa todo el conjunto de observaciones y en rojo la observación que se quiere clasificar. En el resto de figuras están las diferentes observaciones según pertenecen a una clase u a otra además de la que se pretende clasificar. Finalmente el resultado de aplicar K-NN indica que pertenece al *cluster* 5, algo que se puede apreciar en las imágenes.



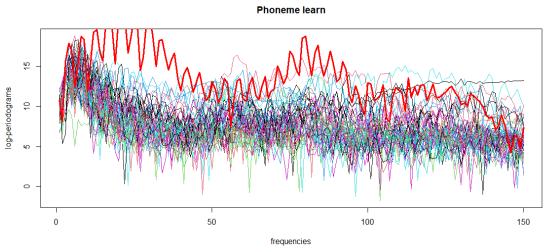
(a) Conjunto de todos los datos de entrenamiento y la observación 250 en rojo



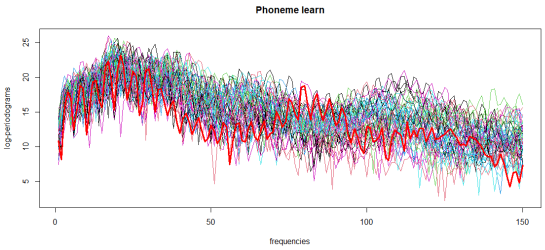
(b) Datos del *cluster* 1 y la observación 250 en rojo



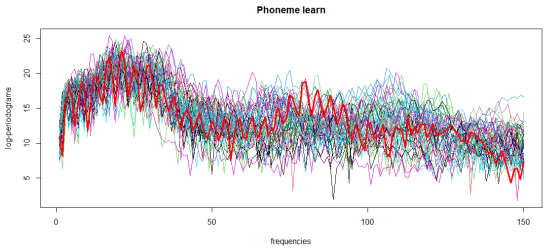
(c) Datos del *cluster* 2 y la observación 250 en rojo



(d) Datos del *cluster* 13 y la observación 250 en rojo



(e) Datos del *cluster* 4 y la observación 250 en rojo



(f) Datos del *cluster* 5 y la observación 250 en rojo

Figura 3.7: Ejemplo clasificación de la observación 250 del conjunto de Phoneme de R mediante K-NN

Así pues se obtiene el Algoritmo 1, donde observamos que consta de 4 pasos. El algoritmo recibe como parámetros de entrada los datos con las etiquetas y el nuevo dato a clasificar.

Ya se ha definido y explicado como funciona el algoritmo, pero aún falta tomar la decisión del valor de K. Si este valor es pequeño puede inducir a una mala predicción por tener poco muestreo, mientras que si se toma una K grande se puede dar el caso que se esté observando algún dato atípico. Por otra parte, una de las características de K es ser impar para no encontrarse con casos de empates.

Algorithm 1 Algoritmo K-NN

Input: Conjunto de los datos {(x_1(t),y_1),(x_2(t),y_2),...,(x_N(t),y_N)}
E grupos
T instantes de tiempo
Input: Nuevo dato a clasificar $x_0(t)$
Output: y_0

Paso 1: Calcular la distancia de los datos a $x_0(t)$ $D := \{d(x_i(t), x_0(t))/i = 1, 2, ..., N\}$
for (i = 1 hasta N) **do**
$d(x_i(t), x_0(t)) := \sqrt{\sum_{t=0}^{T}(x_i(t) - x_0(t))^2}$
end for
Paso 2: Ordenar las distancias ascendentemente
$\{indice.ordenado, D_{ordenado}\}$ = sort(D)
Paso 3: Seleccionar $\{(x_j^{NN}(t), y_j^{NN})/j = 1, 2, ..., K\}$, los K-vecinos más cercanos a x_0,
for (i = hasta K) **do**
$x_j^{NN}(t) = x_{indice.ordenado[i]}$
$y_j^{NN} = y_{indice.ordenado[i]}$
end for
Paso 4: Asignar la nueva clase por mayoria de K-vecinos
y_0 =arg $max_y \sum_{W^{NN}(t)} \delta(y = y_j^{NN})$

La forma de saber que valor de K es más adecuado es ir aplicando diferentes valores y realizar evaluaciones de los resultados. De esta forma se selecciona la K con mejores resultados, para ello se calcula la probabilidad de las buenas clasificaciones.

Siguiendo nuestro ejemplo hemos aplicado K-NN con diferente número de vecinos, en concreto hemos aplicado el modelo para 1,2,3,4,5 y 60 vecinos. Para obtener la probabilidad de bien clasificados debemos obtener el número de observaciones que se han clasificado correctamente y dividirlo por el total de observaciones.

En la Tabla 3.1 observamos los resultados para nuestro ejemplo y por tanto podemos determinar que el mejor número de vecinos para aplicar el método es 5.

K	1	2	3	4	5	60
Probabilidad	0.876	0.792	0.884	0.884	0.912	0.872

Tabla 3.1: Valores de probabilidad para estar bien clasificado para $K = 1, 2, 3, 4, 5, 60$ con los datos de Phoneme.

3.3.2. Modelos de clasificación no supervisada

Los modelos de clasificación no supervisadas no tienen en cuenta si los datos están previamente etiquetados, por tanto solo tendremos la variable aleatoria funcional $X(t) := \{x_i(t) : t \in [0,T]\ ,\ i = 1,2,...,N\}$.

Estos algoritmos se utilizan para hacer grupos a partir de los datos evaluando según patrones distintos y sus similitudes. Reciben la connotación de "no supervisado"puesto que el modelo no parte de unas etiquetas previas. Dentro de este grupo de métodos revisaremos el método de K-medias y los algoritmos jerárquicos *cluster* de aglomeración (o ascendentes).

Para evaluar los resultados de los modelos de clasificación existe varias técnicas de bondad. Una de las medidas se basa en la silueta.

Definición 12 *Se define silueta $s(i)$ para el objeto i como:*

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{3.29}$$

con $a(i) = promedio(\sum_{j:C(j)=C(i)} d(x_i(t), x_j(t)))$ *y con* $b(i) = promedio(\sum_{j:C(j)\neq C(i)} d(x_i(t), x_j(t))$.

Se tiene que $-1 \leq s(i) \leq 1$ para cualquier objeto i.
De esta forma tenemos tres posibles casos para evaluar el resultado:

- $s(i)$ es próximo a 1, por lo que el objeto i está bien clasificado.
- $s(i)$ es próximo a 0, por lo que no tenemos muy claro si el objeto i está bien clasificado o no.
- $s(i)$ es próximo a -1, por lo que el objeto i está mal clasificado.

De forma general se puede tomar la media de la silueta para todo el conjunto y estudiar su valor para hacerse una idea del resultado de la clasificación. Para considerar una clasificación suficiente la media debe ser superior a 0,2 $\sim$ 0,3 y si es superior al 0,5 se puede considerar como buena.

K-medias

Es un método que fue ideado por Hugo Steinhaus y llevado a cabo por Stuart Lloyd en 1957 [8]. Puesto que es una técnica de aprendizaje no supervisado la muestra no está previamente

clasificada y el objetivo es encontrar los K grupos, $C_1, \cdots, C_K$, en que se pueden clasificar los datos. El valor K es un parámetro que previamente se debe fijar y por tanto, cada uno de los *cluster* tendrá $|h_j|$ elementos con $j = 1, 2, ..., K$.

Definición 13 *Dado un cluster C_j, $j = 1, 2, ..., K$, se define su centroide como la función $\mu_j(t)$ para $t \in [0, T]$ como:*

$$\mu_j(t) = \frac{1}{|h_j|} \sum_{x_i \in C_j} x_i(t) \tag{3.30}$$

siendo el centroide la función que minimiza la distancia a todas las observaciones del cluster.

Una vez definido el elemento principal del método podemos construir el Algoritmo 2, cuya entrada es el conjunto de datos y proporciona una clasificación de ellos. Para realizar la mejor elección del número de *clusters* (K), la técnica más utilizada es el método del codo, el cual comentaremos posteriormente.

El algoritmo consiste en asignar los datos al grupo cuyo centroide este más cerca y recalcular los centroides, estos pasos se repiten de forma iterativa hasta que no se modifiquen los centrodies.

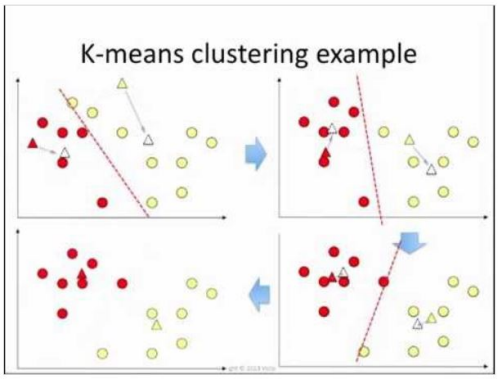


Figura 3.8: Ejemplo Clasificación K-Medias

Para entenderlo podemos observar la Figura 3.8, en ella se visualiza una representación del algoritmo genérico. En este ejemplo se parte de un conjunto de 14 datos representado por puntos y con dos categorías. Cada una de las categorías se representa con un color, rojo y blanco, además sus centroides aleatorios están representados por triángulos. En la primera vuelta del algoritmo, imagen superior izquierda, se puede visualizar como el grupo rojo posee 5 datos y el blanco otros 9 que son los datos más cerca de los centroides. Además se recalcula el centroide y se pasa a la segunda vuelta, imagen superior derecha, donde el *cluster* rojo posee 7 datos y el blanco 7 también.De esta forma se repite este proceso hasta que los centroides no se modifican.

Algorithm 2 Algoritmo K-medias

Input: Conjunto de los datos $\{x_1(t),x_2(t),...,x_N(t)\}$
T instantes de tiempo
Output: Conjunto de los datos $\{(x_1(t),y_1),(x_2(t),y_2),...,(x_N(t),y_N)\}$

Paso 1: Seleccionar el número de grupos, K ▷ Usando el método del codo
do
Paso 2:
if (Primera iteración) **then**
Tomar aleatoriamente los K centros inciales, $\{\mu_j(t)/j = 1, 2, ..., K\}$
else
$\mu_j(t) = \hat{\mu}_j(t) : j = 1, 2, ..., K$
end if
Paso 3: Calcular las distancias de los datos a los centroides D:=$\{d(x_i(t), \mu_j(t))/i = 1, 2, ..., N$ y $j = 1, 2, ..., K\}$
for (i = 1 : N) **do**
for (k = 1 : K) **do**
$D_{i,j} = d(x_i(t), \mu_j(t)) := \sqrt{\sum_{t=0}^{T}(x_i(t) - \mu_j(t))^2}$
end for
end for
Paso 4: Asignar cada $x_i(t)$ al *cluster* con centroide más cercano
y_i=arg $min_{i \leq j \leq K}$ $D_{i,j}$
Paso 5: Construir los *clusters*
$C_j = \{x_i(t) : y_i = j\}$
$h_j = \#C_j$
Paso 6: Recalculamos los centroides $\{\hat{\mu}_j(t)/j = 1, 2, ..., K\}$
$\hat{\mu}_j(t) = \frac{1}{|h_j|} \sum_{x_i \in C_j} x_i(t)$ ▷ Ecuación 3.30
while ($\mu_j(t) \neq \hat{\mu}_j(t) : j = 1, 2, ..., K$)

Una vez que entendemos la forma básica vamos a ver un ejemplo con funciones, para ello observamos la Figura 3.9. En ella se observan los datos del ejemplo que hemos usado en la Sección 3.3.1. Como podemos apreciar se ha aplicado esta técnica con 4 centroides y se ha obtenido como resultado final la figura de la derecha, donde se aprecian los centroides. Mientras que en la figura izquierda se observan como quedan clasificados los datos.

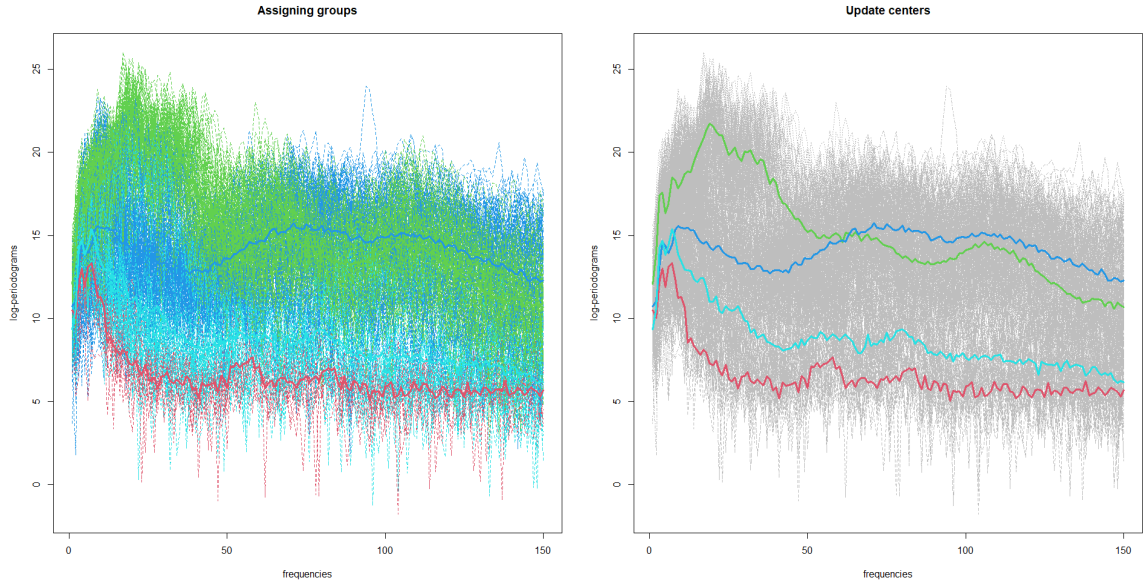


Figura 3.9: Ejemplo de *clustering* del conjunto de Phoneme de R meidante K-mean

Una vez se ha explicado todas las partes del algortimo solo queda ver que es el método del codo. Para ello previamente se debe definir el término inercia.

Definición 14 *El cálculo de la suma de las distancias al cuadrado de cada objeto a su centroide recibe el nombre de inercia.*

$$WCSSE = \sum_{i=1}^{K} \sum_{x_j \in C_i} d^2(x_j - \mu_i) \tag{3.31}$$

Por tanto esta técnica evalúa cómo de cohesionados están los grupos. Luego el algoritmo consiste en calcular la inercia para todos los *clusters* y representar dichos resultados en una gráfica, se puede ver como ejemplo la Figura 3.10.En esta figura tenemos los diferentes valores de la inercia para la clasificación de los datos de Phoneme de 1 hasta 10 *clusters*.

Una vez representados los resultados el número de grupos será aquel en el que se empiece a observar una estabilidad. De forma gráfica se toma el número K al punto que forma un "codo",

es decir, aquel punto en que la diferencia entre él y el siguiente ya no supone una gran diferencia como entre ese punto y el anterior. En el ejemplo (Fig: 3.10) este codo está en los valores K 4 y 5.

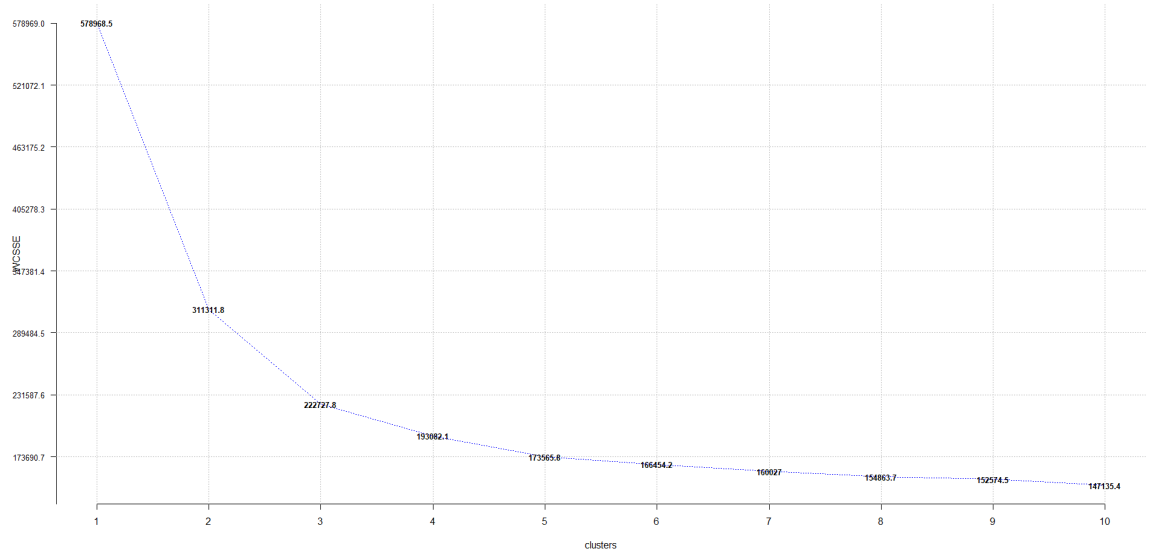


Figura 3.10: Ejemplo método del codo.

Algoritmo Jerárquico

Este algoritmo fue desarrollado por Peña (2002) [10] y parte de una matriz de distancias o similitudes. En este tipo de modelo se diferencian dos subtipos, de aglomeración y de división. El primero de ellos se parte de cada uno de los datos y se van agrupando, mientras que el segundo parte del total de datos y se van dividiendo sucesivamente.

Todos los métodos de aglomeración, también denominados ascendentes (Manly, 1990 [9]) tienen la misma estructura a excepción de la forma de recalcular la distancia entre los grupos, Medida de Vinculación, que se usa en el paso 4.
En el paso inicial se parte de los grupos individuales, y posteriormente se eligen los elementos más próximos en la matriz de distancias. Se repite este proceso hasta que no queden *clusters* por unir. En el Algoritmo 3 podemos observar todos estos pasos.

A continuación vamos a detallar las diferentes Medidas de Vinculación, que se aplican en el paso 4, donde se sustituye en la matriz de distancias las filas de dichos elementos por una nueva siguiendo uno de estos criterios:

Algorithm 3 Algoritmo jerárquico

Input: Conjunto de los datos $\{x_1(t),x_2(t),...,x_N(t)\}$
T instantes de tiempo
Output: Conjunto de los datos $\{(x_1(t),Y),(x_2(t),Y),...,(x_N(t),Y)\}$

Paso 1: Calcular la matriz de distancias, D:=$\{d(x_i(t), x_j(t))/i, j = 1, 2, ..., N\}$
for (i = 1 hasta N) **do**
for (k = 1 hasta N) **do**
$d(x_i(t), x_j(t)) := \sqrt{\sum_{t=0}^{T}(x_i(t) - x_j(t))^2}$
end for
end for
Paso 2: Asignar a cada punto su propio grupo
while (rang(D)$\neq$ 1) **do**
Paso 3: Se fusionan los 2 grupos más cercanos
Paso 4: Recalculo matriz distancias. Se aplica la Medida de Vinculación elegida
end while

- Encadenamiento simple o distancia mínima (Fig. 3.11)
  La distancia es la mínima entre los dos objetos. Se suele utilizar cuando se quieren grupos alargados que pueden incluir gran variedad de datos en los extremos.

$$d(C_i, C_j) = min(\{d(x_i, x_j)/x_i \in C_i, \quad x_j \in C_j\})$$

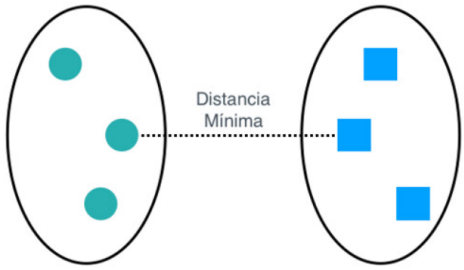


Figura 3.11: Representación encadenamiento simple

- Encadenamiento completo o distancia máxima (Fig. 3.12)
  La distancia es la máxima entre los dos objetos. Se suele utilizar cuando se quieren grupos con forma esférica.

$$d(C_i, C_j) = max(\{d(x_i, x_j)/x_i \in C_i \quad x_j \in C_j\})$$

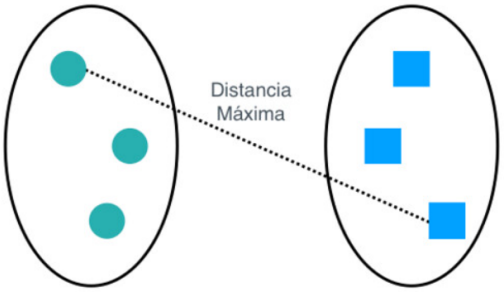


Figura 3.12: Representación encadenamiento completo

- Encadenamiento medio entre grupos (Fig. 3.13)
  La distancia es la media ponderada entre los objetos de ambos grupos. Al igual que el encadenamiento simple, es invariante a transformaciones monótonas:

$$d(C_i, C_j) = \frac{1}{h_i * h_j} \sum_{x_i \in C_i} \sum_{x_j \in C_j} d(x_i, x_j)$$

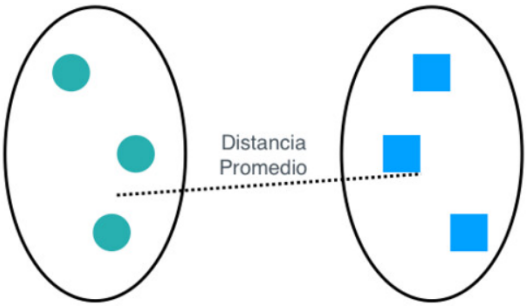


Figura 3.13: Representación encadenamiento medio

- Método del centroide (Fig. 3.14)
  Únicamente es válido para variables continuas y equivale a la distancia euclídea entre sus centros.

$$d^2(C_i, C_j) = \frac{1}{h_i * h_j} \sum_{x_i \in C_i} \sum_{x_j \in C_j} d^2(x_i, x_j) - \sum_{k=1}^{K} \sum_{x_i \neq x_j \in C_k} d^2(x_i, x_j)$$

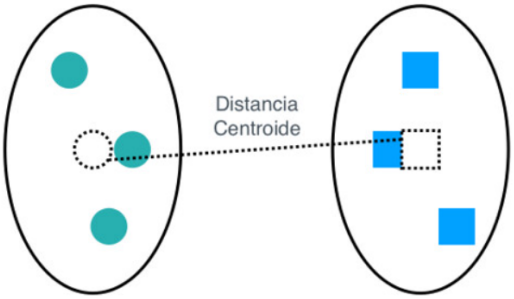


Figura 3.14: Representación método centroide

- Método de Ward (Fig. 3.15)
  Se suma la media de las distancias euclídeas al cuadrado para cada grupo.

$$W = \sum_{k=1}^{K} \sum_{x_i \in C_k} (x_i - \overline{x_k})^t (x_i - \overline{x_k})$$

  Donde $\overline{x_k}(t)$ es la media del grupo C_k

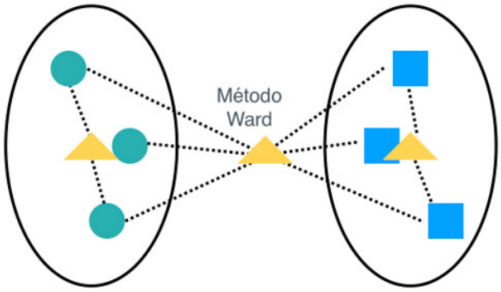


Figura 3.15: Representación método Ward

Para poder ver de forma más clara este modelo existe la representación gráfica del resultado, que recibe el nombre de dendograma. En la Figura 3.16 tenemos la representación gráfica del resultado de aplicar este tipo de algoritmo a las 50 primeras curvas de los datos Phoneme. En ella se observa en la parte inferior las 50 curvas etiquetadas por su índice de ordenación, cada una en un grupo. De cada uno de los grupos sale una línea recta que se une a otra de ellas mediante una horizontal, de esta forma los subgrupos se unen creando un único *cluster*. Entonces se repite el proceso hasta que todas quedan unidas en una línea horizontal.

Para obtener un número de K *clusters* se debe cortar el dendograma a la altura donde trazando una línea horizontal se corten K líneas verticales, de esta forma se obtiene los diferentes grupos que almacenan a las curvas que están unidas. En el ejemplo que estamos estudiando se

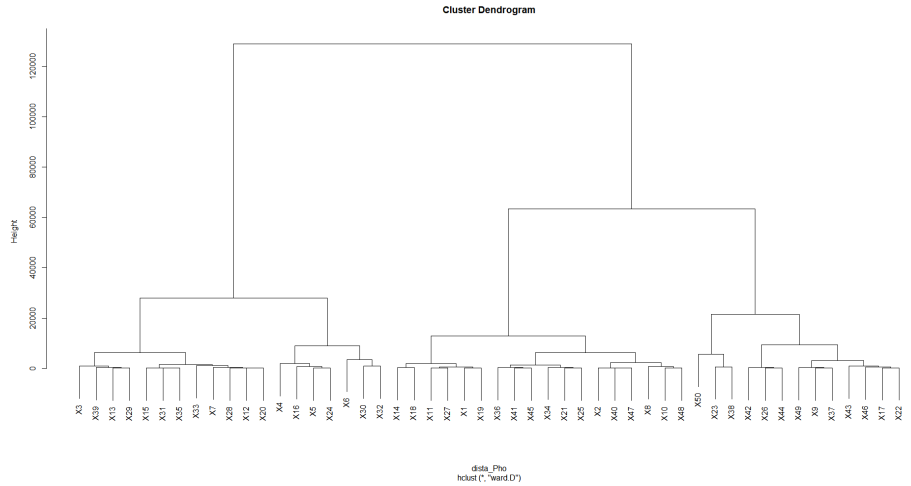


Figura 3.16: Ejemplo Dendograma de las 50 primeras curvas de la base de datos Phoneme.

puede obtener 3 *clusters*, los cuales apreciamos en la Figura 5.18 mediante la representación de rectángulos rojos. .

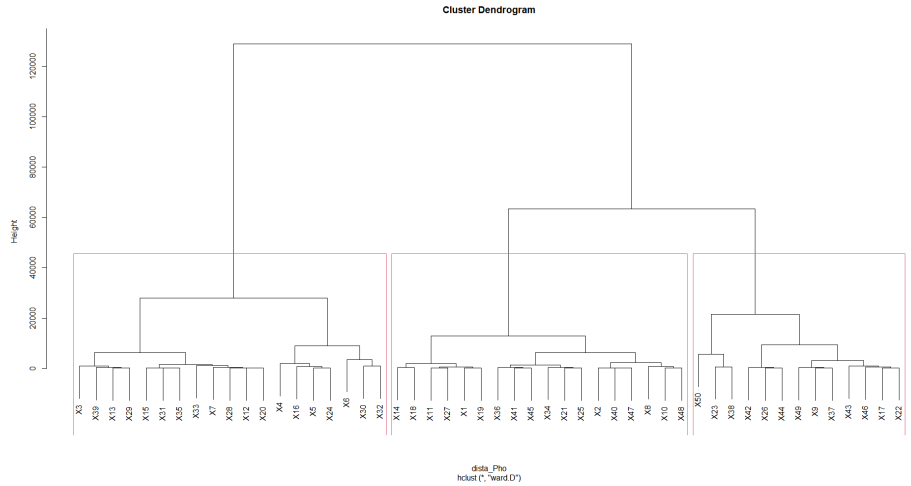


Figura 3.17: Dendograma de la Figura 3.16 cortado con 3 *clusters*

Capítulo 4

Implementación de un *Dashboard* interactivo

La motivación principal del trabajo fue el poder analizar las series temporales contenidas en la base de datos de "Serie de casos con PDIA positiva en la Comunidad Valenciana, según fecha de diagnóstico de laboratorio"[13]. En este capítulo vamos a estudiar la base de datos (Sec. 4.1) y la implementación de una interfaz para visualizar la base de datos (Sec. 4.2).

4.1. Nuestra base de datos

Tras la aparición de la enfermedad de la Covid 19 en el territorio Español y su posterior denominación como pandemia mundial las diferentes instituciones empezaron a recoger información para poder realizar estudios. Uno de los datos que recogieron la "Conselleria de Sanidad Universal y Salud Pública"son los casos positivos de dos pruebas de detección de infección activa, las cuales son el test de antígenos y la PCR. Todos los datos recogidos han seguido los procedimientos y documentación técnica elaborada por el Ministerio de Sanidad Español.

Los datos fueron recogidos y publicados desde el 1 de Febrero de 2020 hasta el 29 de mayo de 2022. Aunque la información era diaria, la actualización de los datos era semanal, los miércoles se cargaban todos los datos de la semana anterior.

En concreto, la base de datos contiene información del total de casos diarios de la Comunidad Valenciana, los casos separados por provincias, casos diarios globales en hombres y mujeres y casos diarios por cada uno de los 24 departamentos de salud. Por tanto hay un total de 30 variables. Para cada una de estas variables se ha almacenado 849 datos, los cuales corresponden

a los días que ha estado activa la recogida de datos.

En la Tabla 4.1 se puede observar el identificador de cada departamento de salud, junto con una variable que indica a qué provincia pertenece. Además hemos representado el mapa de la Comunidad Valenciana divido en sus 24 departamentos para así poder ubicarlos de forma rápida (Fig. 4.1).

Identificador	Nombre del departamento de salud	Etiqueta
1	“VINAROS”	1
2	“CASTELLO”	1
3	“LA PLANA”	1
4	“SAGUNT”	1
5	“VCIA CLINIC-LA MALVA-ROSA”	2
6	“VCIA ARNAU DE VILANOVA LLIRIA”	2
7	“VALENCIA - LA FE”	2
8	“REQUENA”	2
9	“VALENCIA -HOSPITAL GENERAL”	2
10	“VALENCIA - DOCTOR PESET”	2
11	“LA RIBERA”	2
12	“GANDIA”	2
13	“DENIA”	3
14	“XATIVA - ONTINYENT”	2
15	“’ALCOI”	3
16	“LA MARINA BAIXA”	3
17	“’ALACANT-SANT JOAN D’ALACANT”	3
18	“’ELDA”	3
19	“’ALACANT - HOSPITAL GENERAL”	3
20	“’ELX - HOSPITAL GENERAL”	3
21	“’ORIHUELA”	3
22	“TORREVIEJA”	3
23	“MANISES”	2
24	“’ELX-CREVILLENT”	3

Tabla 4.1: Identificadores y etiquetas de los departamentos de salud de la base de datos. Etiquetas: 1 - Provincia de Castellón, 2 - Provincia de Valencia, 3 - Provincia de Alicante.

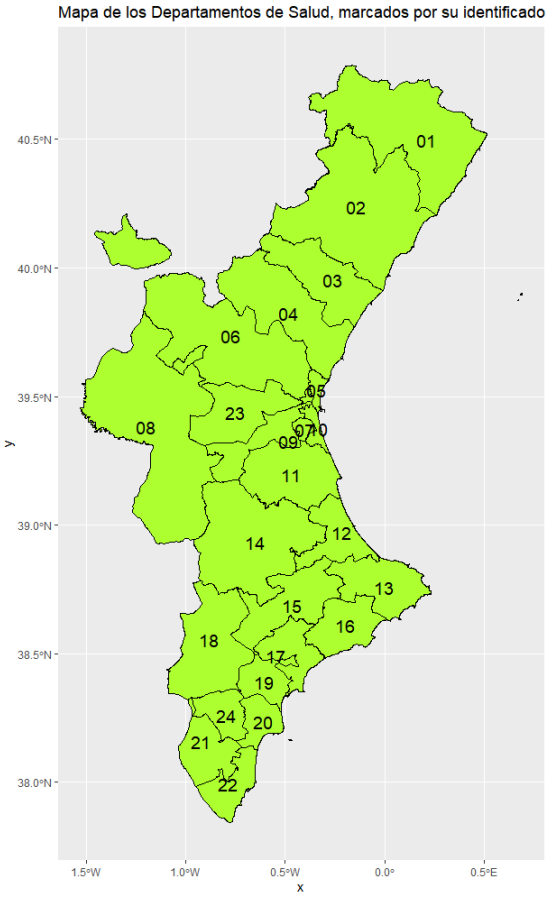


Figura 4.1: Mapa de los Departamentos de Salud con sus identificadores.

4.2. Implementación del *dashboard* con *Shiny*

Una vez tenemos clara toda la información que nos proporciona la base de datos nos planteamos el trabajo a realizar y qué nos hacia falta. Para poder visualizar los análisis de forma clara y directa y poderlos publicar en la web, decidimos implementar un *dashboard* en R. Para ello utilizamos el paquete *Shiny*[4] que ofrece una interfaz para interactuar con los datos.

A partir de los datos de contagios y de la población de cada zona, el primer paso fue calcular la media, desviación estándar y total de positivos, además también miramos la media, la suma de casos positivos de los últimos 14 días y por otra parte calculamos la tendencia e incidencia de la enfermedad por cada 100.000 habitantes en cada área de los últimos 14 días.

Definición 15 *Dada una serie temporal $X_i(t)$ se define tendencia, T, como la diferencia entre la observación pasada, $X_i(t_p)$, y la actual, $X_i(t_a)$ de la forma:*

$$T = X_i(t_p) - X_i(t_a) \tag{4.1}$$

Definición 16 *Dada una serie temporal $X_i(t)$ sobre una población P, se define incidencia, I, por cada* 100,000 *habitantes en los últimos D dias como:*

$$I = \frac{\sum_{t=1}^{D} X_i(t)}{P} \times 100000 \tag{4.2}$$

Toda la información anterior la almacenamos en una tabla (Fig. 4.2) para poder trabajar posteriormente con algunos de los estadísticos obtenidos.

	Regió	Mitjana	Máxim_Alcançat	Casos_Últim_Día	Casos_Ultims_14_Dies	Casos_Totals	Incidencia	Tendencia	Varianza
1	C. Valenciana	1695.630	24894	315	18703	1439590	373.778	8233	3667.589
2	Homes	795.380	11933	147	7435	675278	148.588	3355	1728.558
3	Dones	899.777	13181	168	11208	763911	223.991	4912	1939.964
4	Prov. Alacant	581.721	8664	114	5880	493881	316.353	3455	1263.594
5	Prov. Castello	208.041	3391	57	2665	176627	459.513	782	469.873
6	Prov. Valencia	905.862	13757	144	10158	769077	396.004	3995	1954.984
7	Dep. Vinaros	32.115	472	12	649	27266	714.632	297	63.597
8	Dep. Castello	97.225	1555	21	948	82544	334.957	223	215.409
9	Dep. La Plana	68.385	1312	24	951	58059	507.855	169	171.359
10	Dep. Sagunt	49.903	779	11	525	42368	343.139	368	110.424
11	Dep. VCIA Clinic-La Malva-Rosa	123.625	2032	16	1190	104958	344.167	350	269.167
12	Dep. VCIA Arnau de Vilanova Lliria	113.286	1755	32	1458	96180	455.928	510	246.403
13	Dep. Valencia- La Fe	100.133	1559	11	1273	85013	441.202	457	221.580
14	Dep. Requena	18.584	316	4	208	15778	403.970	176	39.822
15	Dep. Valencia Hosp. General	124.892	2062	22	1372	106033	376.905	242	281.283
16	Dep. Valencia Doctor	102.955	1569	13	1174	87409	419.944	464	211.714
17	Dep. La Ribera	85.932	1386	17	922	72956	354.962	364	186.817
18	Dep. Gandia	58.883	942	5	536	49992	301.540	446	131.934
19	Dep. Denia	51.294	793	8	412	43549	239.896	342	113.257
20	Dep. Xativa-Ontinyent	67.736	1029	5	768	57508	394.832	423	144.555
21	Dep. Alcoi	49.523	892	3	336	42045	245.164	479	116.085
22	Dep. La Marina Baixa	56.581	871	16	540	48037	291.394	269	114.376
23	Dep. Sant Joan D'Alacant	67.947	1149	15	936	57687	420.170	376	149.103
24	Dep. Elda	69.444	1205	8	597	58958	314.873	260	162.617
25	Dep. Alacant Hosp. General	85.833	1310	35	1167	72872	419.641	366	185.724
26	Dep. Elx Hosp. General	63.048	922	14	651	53528	384.493	437	138.422
27	Dep. Orihuela	53.404	970	7	595	45340	353.855	375	119.927
28	Dep. Torrevieja	40.218	653	7	329	34145	176.858	311	81.652
29	Dep. Manises	70.160	1361	8	849	59566	409.719	288	167.838
30	Dep. Elx-Crevillent	44.389	798	1	317	37686	202.548	240	106.582

Figura 4.2: Tabla con estadísticos descriptivos

El paquete *Shiny* facilita crear aplicaciones web interactivas desde R. Esto permite incorporar documentos *RMarkdown* y crear paneles, además puede ampliar sus aplicaciones con temas *CSS*, *html* y acciones con *JavaScript*.

Al crear un documento *Shiny*, realmente estas creando una aplicación web de forma local, tienes la posibilidad de elegir entre realizar todo el código en un único archivo o en varios. De esta forma se pueden diferenciar dos partes, el bloque *ui* y el bloque *server*. En el bloque *ui* se define todo lo relacionado con la interfaz, por tanto aquí se definirá el aspecto de la aplicación, los diferentes paneles, menús y desplegables que hacen interactiva la web. En el otro bloque, *server*, es donde se realizan todos los cálculos y donde se definen cómo van a ser las salidas, en esta parte se reciben los diferentes *inputs* interactivos y se pueden filtrar para definir diferentes *outputs*.

Una vez decidido el paquete empezamos a trabajar en este nuevo entorno donde teníamos parámetros interactivos. De esta forma decidimos implementar una gráfica para visualizar las series temporales del departamento de salud seleccionado. Para la representación de las series usamos la función *ggplot* del paquete *ggplot2*[14].

Tras lograr que la gráfica funcionara correctamente pasamos a implementar la representación del mapa. Dicha figura es la representación de la Comunidad Valenciana dividida por Departamentos de Salud y su objetivo era servir de referencia para localizar la zona seleccionada. Por tanto, el mapa debía resaltar e indicar la incidencia del departamento seleccionado.

Para implementar el mapa fue necesario buscar argumentos de la función *ggplot* para poder remarcar el departamento seleccionado e introducir etiquetas en los gráficos.

Finalmente, para mostrar la información de forma más visual, decidimos añadir algunos estadísticos importantes, como son la media de los casos, la media de los últimos 14 días y la incidencia de los últimos 14 días. Para hacerlo de una forma clara y visual utilizamos *widgets* de *html*, en concreto usamos *flexdashboard::renderGauge*. Además gracias a la función *shinyalert* del paquete *shinyalert*[2], integramos unas pestañas que sirven como botones para mostrar un pequeño panel con la información necesaria para entender que representa cada parte de la aplicación.

Todo lo que hemos mencionado lo implementamos de forma local, pero una parte de nuestra motivación era poder hacer accesible esta información por tanto decidimos publicar la aplicación web. Para poder publicar la aplicación web hay muchas posibilidades y nosotros nos decantamos por publicarla en un servidor que ofrecía un paquete gratuito. En concreto la web esta publicada en el servidor `https://www.shinyapps.io/`, el paquete con el que nos registramos nos ofrece 25h de actividad al mes y subir hasta un total de 5 aplicaciones web.

El último paso fue descargar el paquete *rsconnect*[1] que nos permite realizar la conexión

entre la aplicación web y el servidor *shinyapps* para cargar todo el contenido. Una visualización de como quedo la aplicación web es la Figura 4.3.

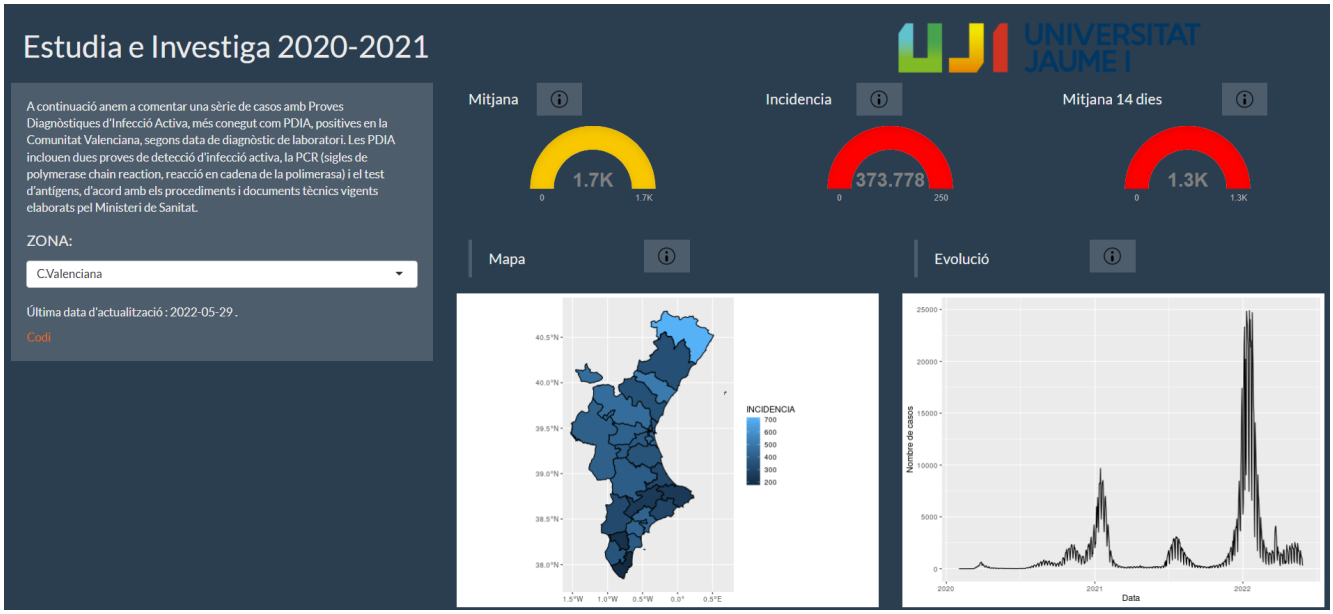


Figura 4.3: Captura de la página web

Para finalizar en la Figura 4.4 observamos la aplicación web con el desplegable para seleccionar el departamento deseado y como los datos se han modificado. Además también podemos apreciar en el mapa su localización.

Esta interfaz está publicada como pagina web en la dirección `https://juan-pg.shinyapps.io/Estudia-e-Investiga2021/` y junto con lo comentado anteriormente cuenta con una pequeña explicación de la finalidad de la web y un enlace al código para poder ver como esta creada.

En el capítulo 5 comentaremos los resultados obtenidos al aplicar los modelos de clasificación vistos en la memoria a esta base de series temporales.

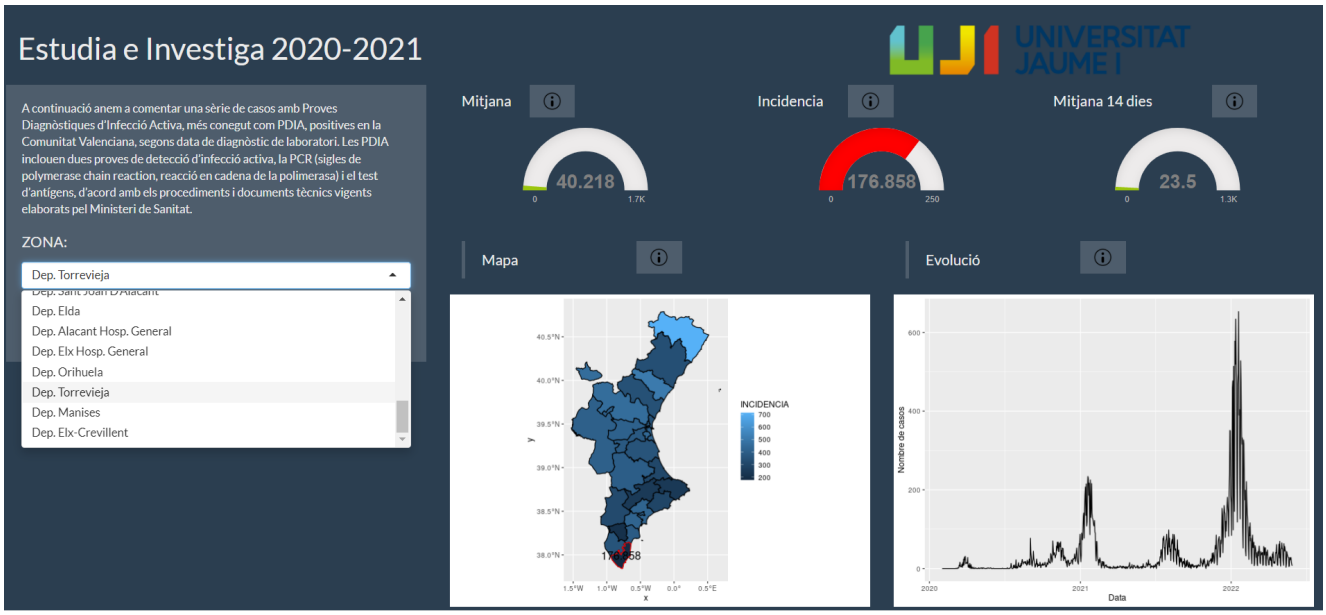


Figura 4.4: Captura de la página web desarrollada en el para el Departamento de Salud de Torrevieja

Capítulo 5

Resultados modelos de clasificación

En este capítulo vamos a aplicar las diferentes técnicas explicadas en el capítulo 3. Para ello vamos a trabajar con la base de datos de "Serie de casos con PDIA positiva en la Comunidad Valenciana, según fecha de diagnóstico de laboratorio"[13], la cual hemos visto en la sección 4.1.

En las siguientes secciones vamos a realizar una descripción de los datos (Sec. 5.1) donde estudiaremos los datos tanto en bases de Fourier como en bases B-splines. Posteriormente aplicaremos los modelos de clasificación no supervisados (Sec. 5.2.1 y Sec. 5.2.2), modelos de clasificación supervisada (Sec. 5.3.1).

Y finalmente, en la sección 5.4 implementaremos una clasificación en la aplicación web comentada en la sección 4.

5.1. Descripción datos

En la sección 4.2 ya hemos visto los descriptivos estadísticos para cada una de las series temporales, pero en esta sección vamos a obtener las funciones que representan nuestros datos en las diferentes bases y además sus estadísticos.

En la Figura 5.1 se representan todos los datos de la base de datos, en ella se encuentran el total de casos de la comunidad, los casos positivos tanto en hombres como en mujeres, el total de datos por provincia y por departamento de salud.

Antes de representar los datos vamos a realizar una modificación en nuestras series temporales. En lugar de trabajar con los datos directamente vamos a calcular las incidencias y realizar

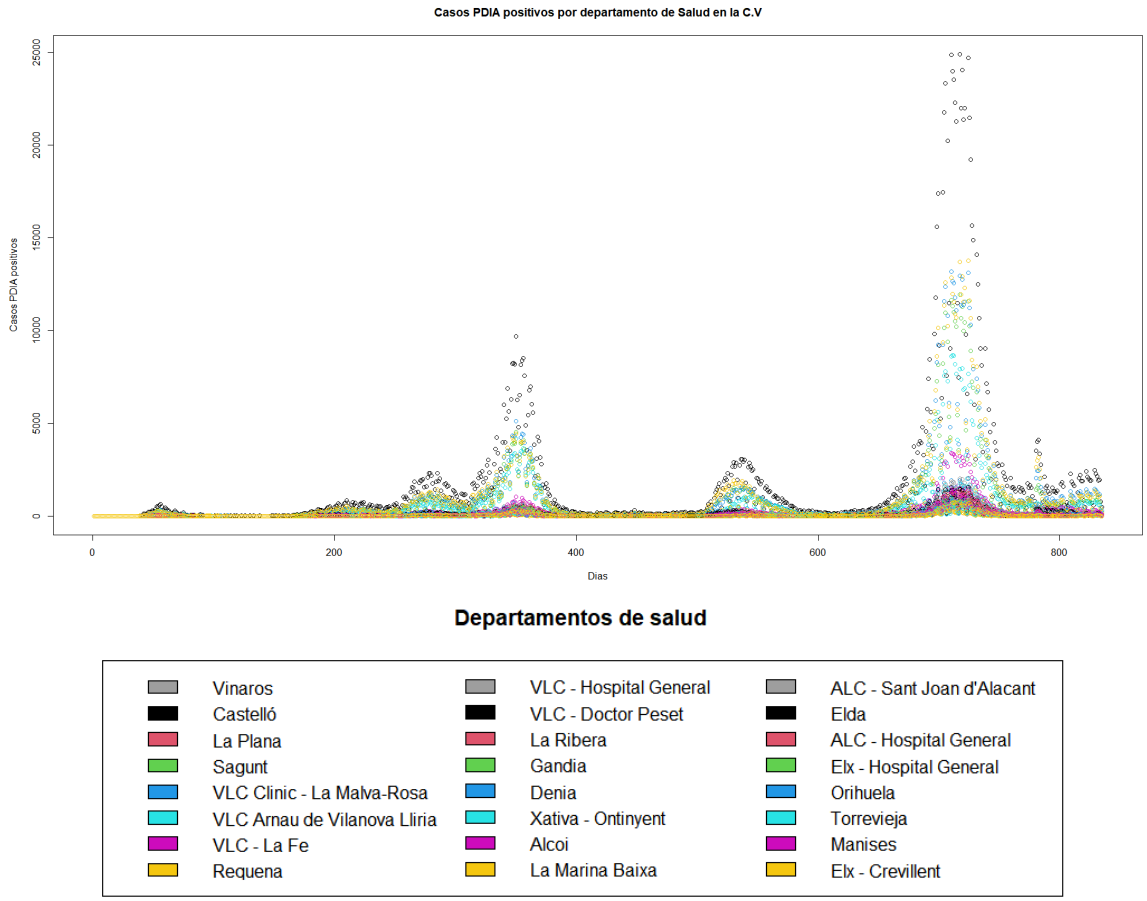


Figura 5.1: Representación de la base de datos “Serie de casos con PDIA positiva en la C.V.”.

así las operaciones con esos resultados. Por tanto aplicamos la Definición 16 para nuestras observaciones, en la Figura 5.2 podemos ver su representación.

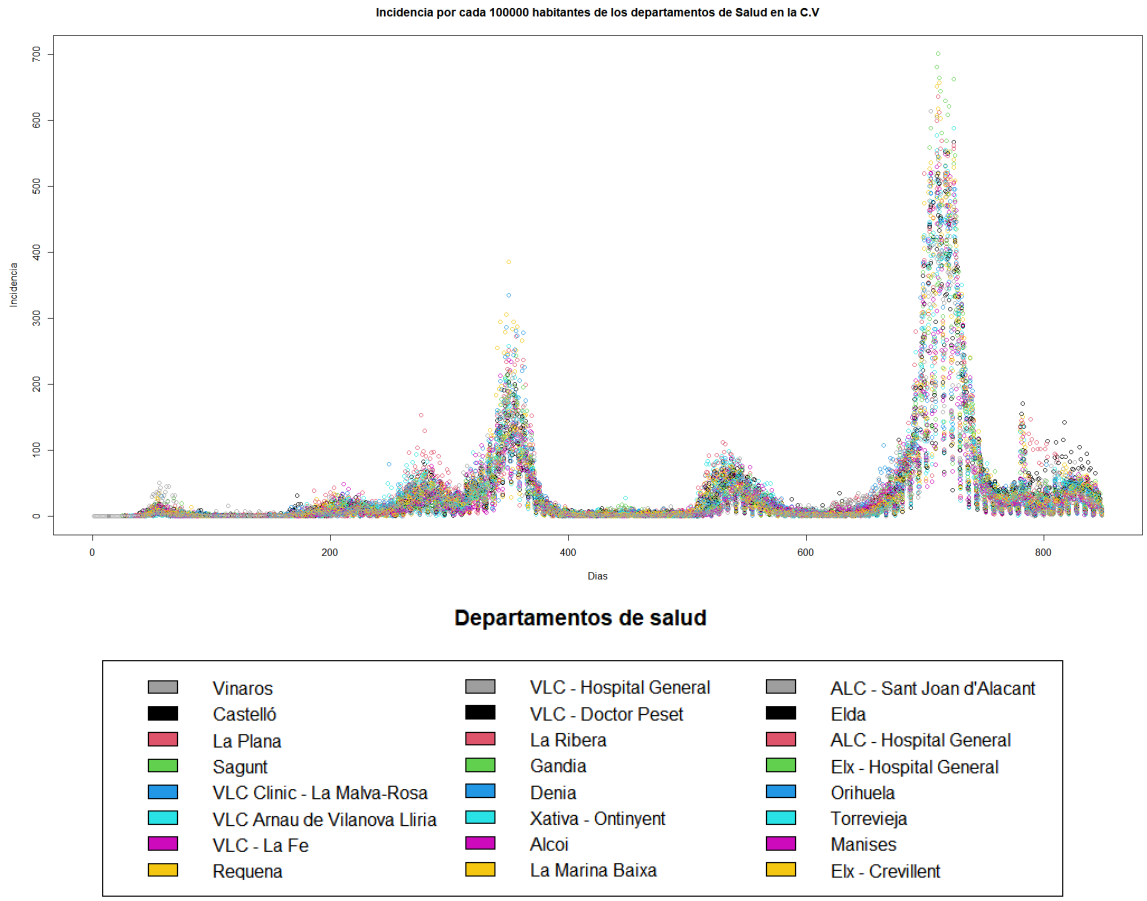
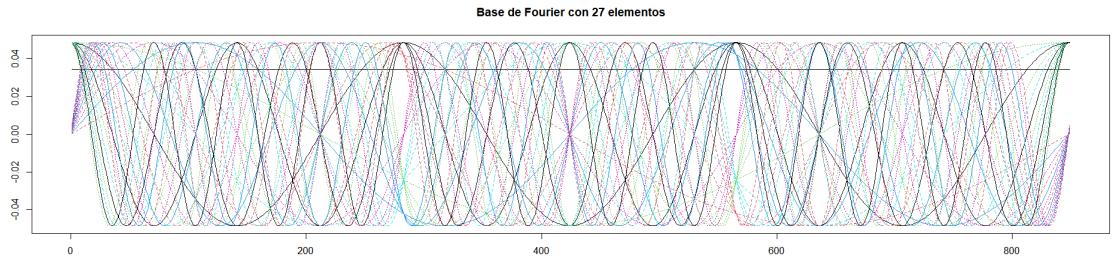


Figura 5.2: Representación de la Incidencia de los datos "Serie de casos con PDIA positiva en la C.V.".

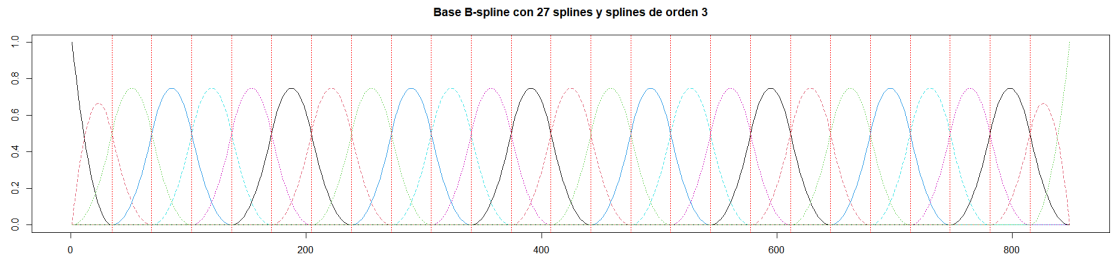
Como hemos explicado en las secciones 3.1.3 y 3.1.4 vamos a aplicar bases de Fourier y B-splines. Las funciones *create.fourier.basis* y *create.bspline.basis* del paquete *fda*[11] nos permiten construir nuestras bases. En concreto creamos una Base de Fourier formada por 27 elementos, 1 constante, 13 senos y 13 cosenos,(Fig. 5.3(a)) y por otra parte una Base B-splines formada por 27 splines de orden 3 (Fig. 5.3(b)).

Una vez tenemos las bases pasamos a representar los datos en ellas, para este paso la función *smooth.basis*, del paquete *fda*, se encarga de obtener los coeficientes de la base usando mínimos cuadrados ponderados, tras realizar una suavización de los datos.

En la Figura 5.4 podemos ver las funciones obtenidas al ajustar las Bases de Fourier, mientras que en la Figura 5.5 observamos las funciones obtenidas con las Bases B-splines.



(a) Base de Fourier de 27 elementos



(b) Base B-splines de 27 splines de orden 3

Figura 5.3: Representación una Base de Fourier y una Base B-splines.

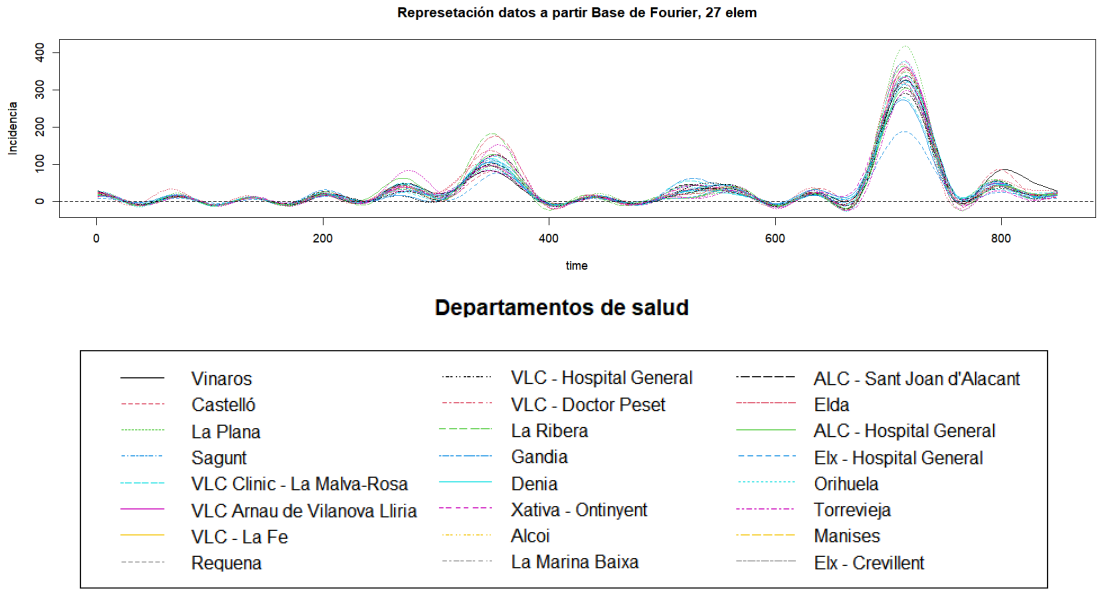


Figura 5.4: Representación datos sobre las Bases de Fourier de la Figura 5.3(a).

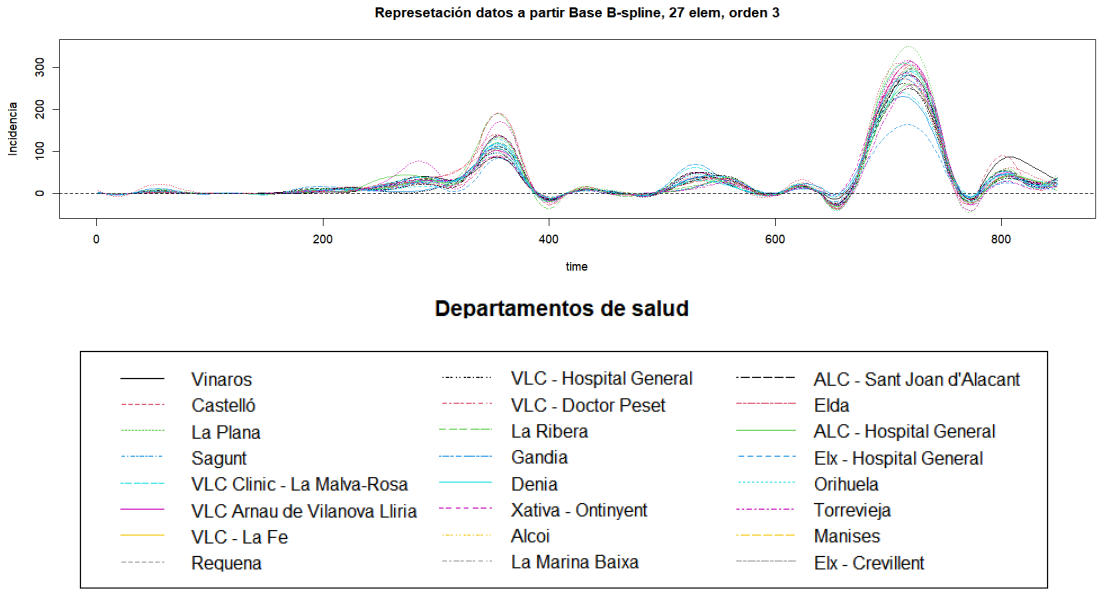
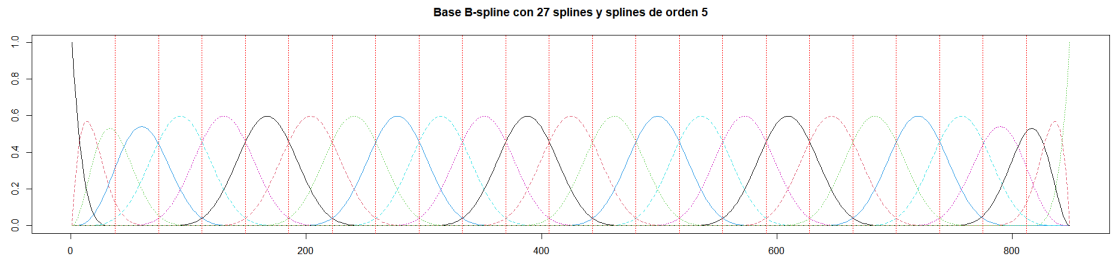


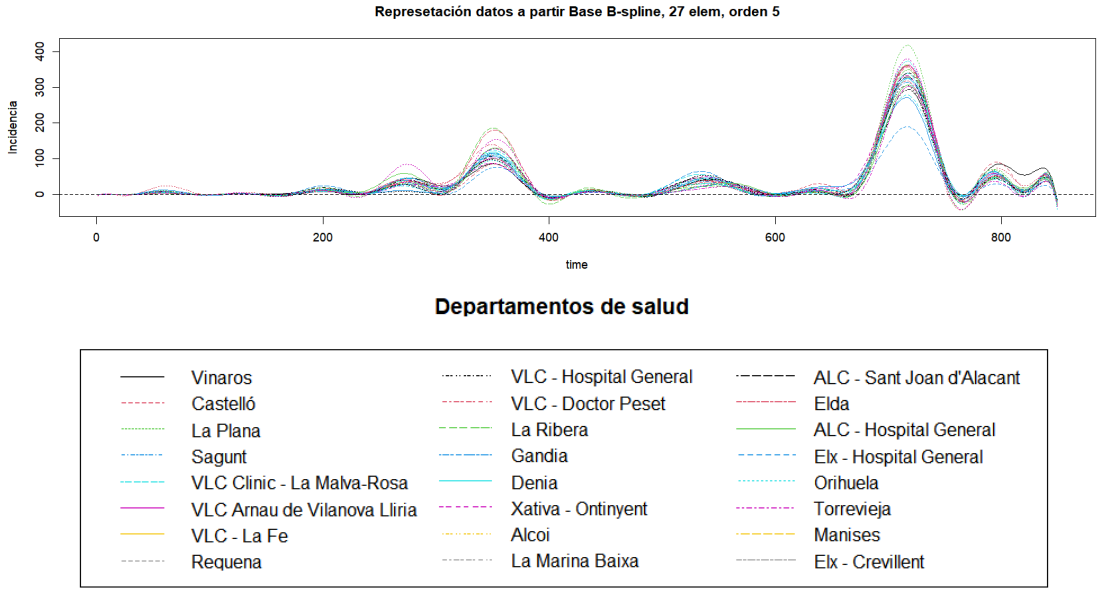
Figura 5.5: Representación datos sobre las Bases B-spline de la Figura 5.3(b).

Para ver como se ajustan estas funciones a las observaciones calculamos el RMSE, como se ha comentado en la sección 3.1.5. Para ello la función *plotfit.fd* del paquete *fda.usc*, nos proporciona una comparación gráfica de los datos reales con los ajustados mediante las funciones y además también indica el error cuadrático medio obtenido en cada caso.

En la Figura 5.7 y la Figura 5.8 podemos ver la comparación anteriormente descrita junto con el RMSE para las 2 primeras observaciones de nuestros datos. A estas dos figuras podemos añadir una tercera, la Figura 5.9, en este caso los datos se han representado mediante una Base B-splines de 27 spline, como las anteriores, pero ahora con splines de orden 5. Podemos observar la base y como se representan las funciones para las incidencias en la Figura 5.6.



(a) Base B-splines de 27 splines de orden 5



(b) Representación de la incidencia

Figura 5.6: Representación incidencias sobre Base B-splines de 27 splines de orden 5.

Puesto que hemos fijado el grado de libertad para todas las bases vamos a revisar el error cuadrático medio. De las dos primeras figuras (Fig. 5.7 y Fig. 5.8) podemos ver que las Bases de Fourier representan de una forma mejor los datos.

Ahora sí aumentamos el orden de los splines y comparamos la figura 5.7 con la figura 5.9, podemos ver que esta vez es la Base B-splines la que obtiene mejores resultados puesto que en este tipo de base también juega un factor importante el grado de los splines. Estos resultados que acabamos de comentar se aplican a todas las observaciones que estamos estudiando.

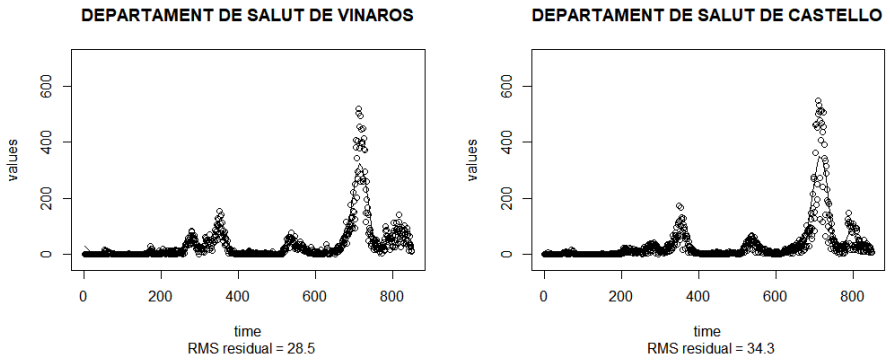


Figura 5.7: Comparación entre los datos y la función en la Base de Fourier con 27 elementos.

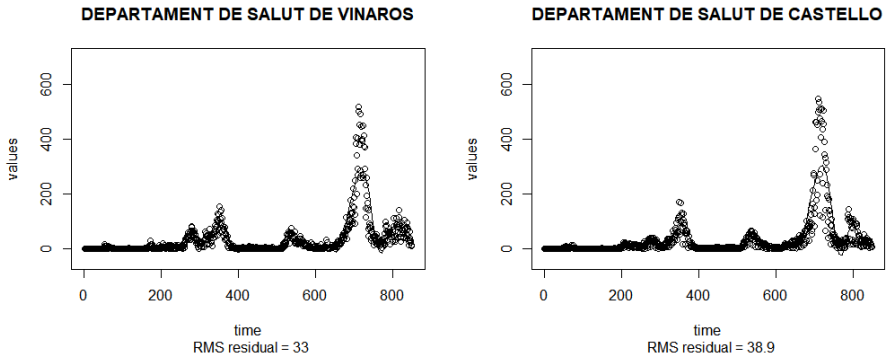


Figura 5.8: Comparación entre los datos y la función en la Base B-splines de 27 splines de orden 3.

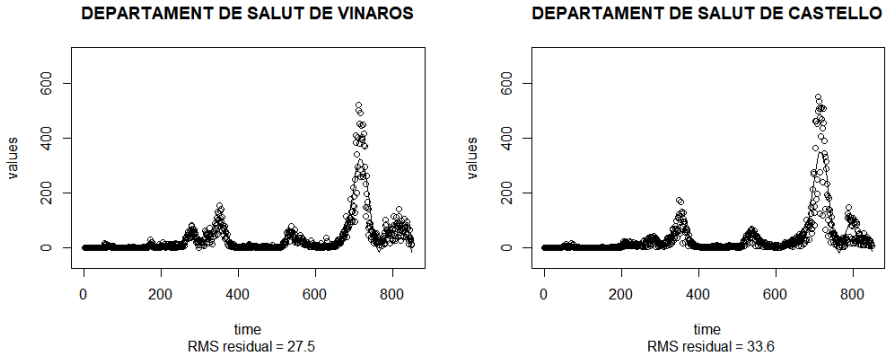
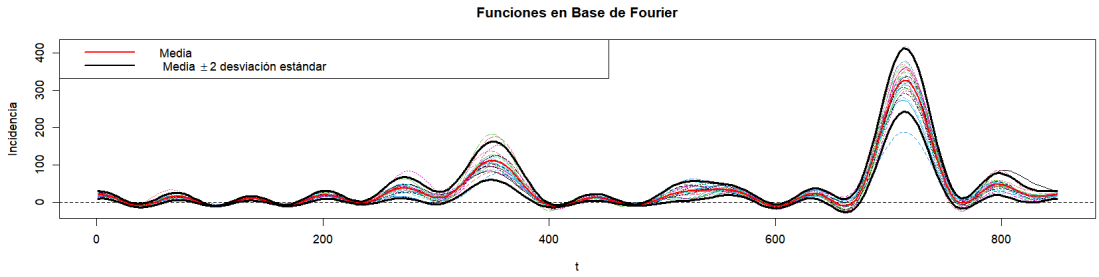


Figura 5.9: Comparación entre los datos y la función en la Base B-splines de 27 splines de orden 5.

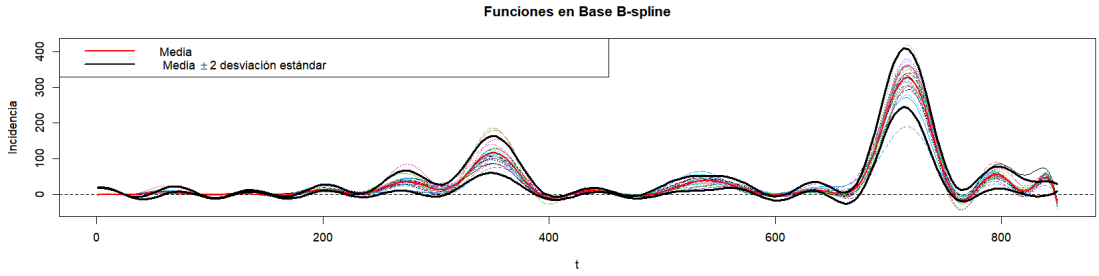
Tras estas comparaciones vamos a seguir trabajando con la Base de Fourier y entre las dos Bases B-splines que hemos presentado, seguiremos con la Base B-splines de 27 splines de orden 5.

Una vez tenemos claras nuestras bases y cómo se ajustan a nuestros datos pasamos a revisar algunos estadísticos.

En la Figura 5.10 podemos observar la media y un intervalo definido como media $\pm$ 2 veces la desviación, como se observa tanto en la subfigura (a) como en la (b) la gran parte de los datos pertenecen a este intervalo.



(a) Media e intervalo de 2 veces la desviación respecto a las Bases de Fourier.



(b) Media e intervalo de 2 veces la desviación respecto a las Bases B-splines.

Figura 5.10: Media (curva roja) e intervalo de confianza (delimitado por las curvas negras) de las funciones respecto a las bases.

La media nos permite hacernos una idea de como se comportan las series a lo largo del tiempo mientras que la desviación nos dan una idea de la variabilidad en todos los puntos.

5.2. Clasificación no supervisada

En esta sección vamos a trabajar con las incidencias de las series temporales de los 24 departamentos de salud. Con estos datos realizaremos el *clustering* de dos formas diferentes para asi comparar los resultados. En la sección 3.3.2 se han explicado dos modelos de clasificación no supervisada, dichos modelos son los que vamos a aplicar: K-medias (Sec. 5.2.1) y un algoritmo jerárquico ascendente (Sec. 5.2.2).

5.2.1. K-means

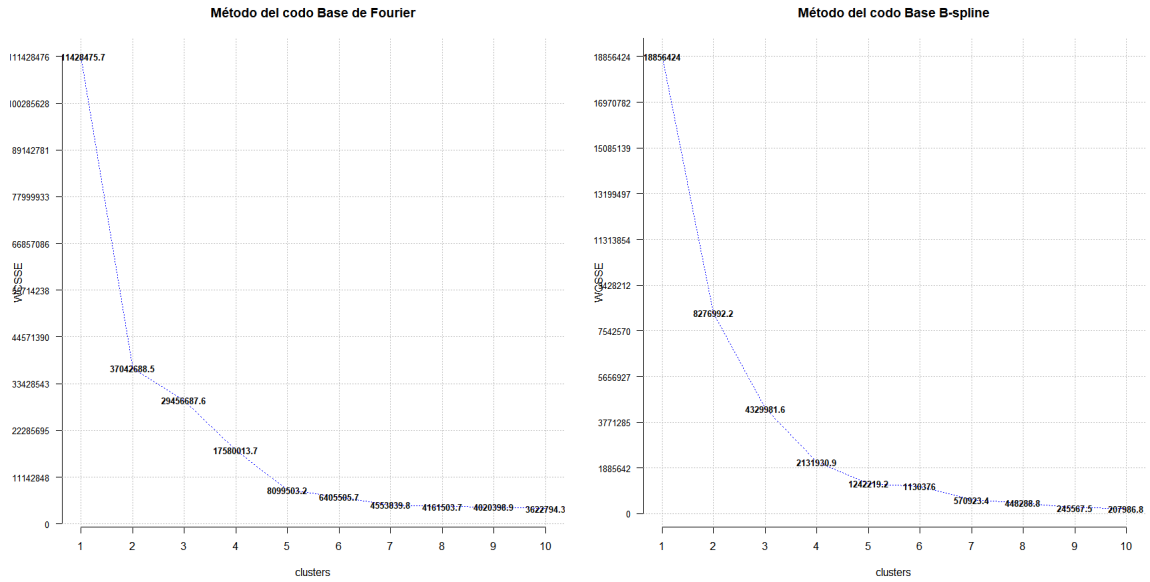
En esta sección vamos a realizar el *clustering* no supervisado aplicando K-medias (Sección 3.3.2). Partiendo de las funciones obtenidas con ambas bases (Fig. 5.4 y Fig. 5.6), el primer paso para realizar la clasificación es encontrar el número de grupos en el que se puede dividir los datos. Para ello se aplica el método del codo.

En la Figura 5.11 podemos observar la representación de los valores de la inercia para distintos números de *clusters* para los dos casos de estudio. En la Figura 5.11(a) se encuentra el resultado para las funciones en Bases de Fourier mientras que en la Figura 5.11(b) para las curvas en Bases B-splines.

En las dos figuras podemos observar un codo para el caso de $K = 4$, el codo en el caso de la Figura 5.11(b) es más pronunciado, pero podemos tomar este valor para aplicar esta técnica de *clustering*.

En la Figura 5.12 se observa el resultado de aplicar este modelo a las curvas en las Bases de Fourier con 4 centroides aleatorios iniciales. En la parte izquierda observamos cada observación coloreada con el color de su *cluster*. Mientras que en la derecha se representan los centroides, en su color correspondiente.

En la Tabla 5.1 esta la clasificación de los departamentos resultante de la clasificación anterior. En el *Cluster* 1 se han clasificado 7 observaciones, en el *Cluster* 2 se han clasificado 8 series temporales, en el *cluster* 3 se han clasificado 3 series temporales mientras que en el *cluster* 4 se han clasificado las otras 6 series temporales.



(a) WCSSE de las funciones en Bases de Fourier (b) WCSSE de las funciones en Bases B-splines

Figura 5.11: Representación del Método del Codo respecto a las bases para deducir el número de *cluster* de K-medias.

Cluster 1	**Cluster 2**
“VINAROS”	“SAGUNT”
“ CASTELLO”	“ VLC - DOCTOR PESET”
“ VLC CLINIC-LA MALVA-ROSA”	“DENIA”
“ VLC ARNAU DE VILANOVA LLIRIA”	“XATIVA - ONTINYENT”
“ VLC - LA FE”	“ALCOI”
“ VLC -HOSPITAL GENERAL”	“LA MARINA BAIXA”
“GANDIA”	“ELDA”
	“ELX - HOSPITAL GENERAL”
Cluster 3	**Cluster 4**
“LA PLANA”	“ LA RIBERA”
“REQUENA”	“ALC-SANT JOAN D’ALACANT”
“MANISES”	“ ALC - HOSPITAL GENERAL”
	“ORIHUELA”
	“TORREVIEJA”
	“ELX-CREVILLENT”

Tabla 5.1: Tabla clasificación con K-medias de las curvas de los Departamentos de Salud en Bases de Fourier

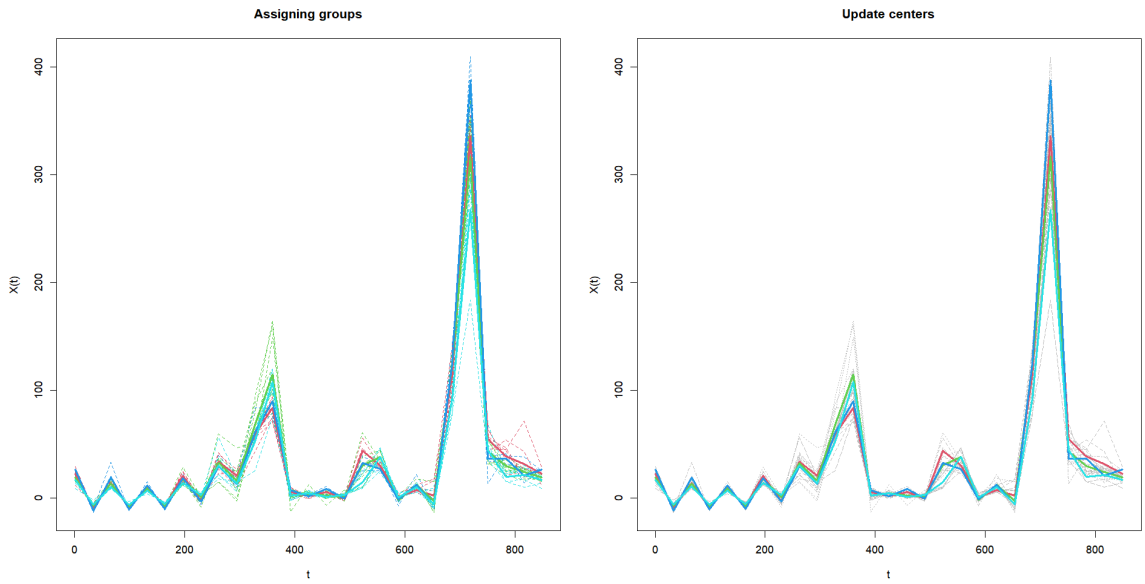


Figura 5.12: K-medias aplicado a las curvas en la Base de Fourier. Rojo: *Cluster* 1, Verde: *Cluster* 2, Cían: *Cluster* 3, Azul: *Cluster* 4

Si ahora observamos la Figura 5.13 podemos visualizar el resultado de aplicar este modelo a las curvas en las Bases B-splines con 4 centroides aleatorios iniciales. En la parte izquierda observamos cada observación coloreada con el color de su *cluster*. Mientras que en la derecha se representan los centroides, en su color correspondiente.

En la Tabla 5.2 podemos ver los *clusters* resultantes. En este hemos obtenido que el *cluster* 1 posee 6 observaciones, el *cluster* 2 posee 10 series temporales, el *cluster* 3 esta formado por 3 observaciones y el *cluster* 4 posee las otras 5 series temporales.

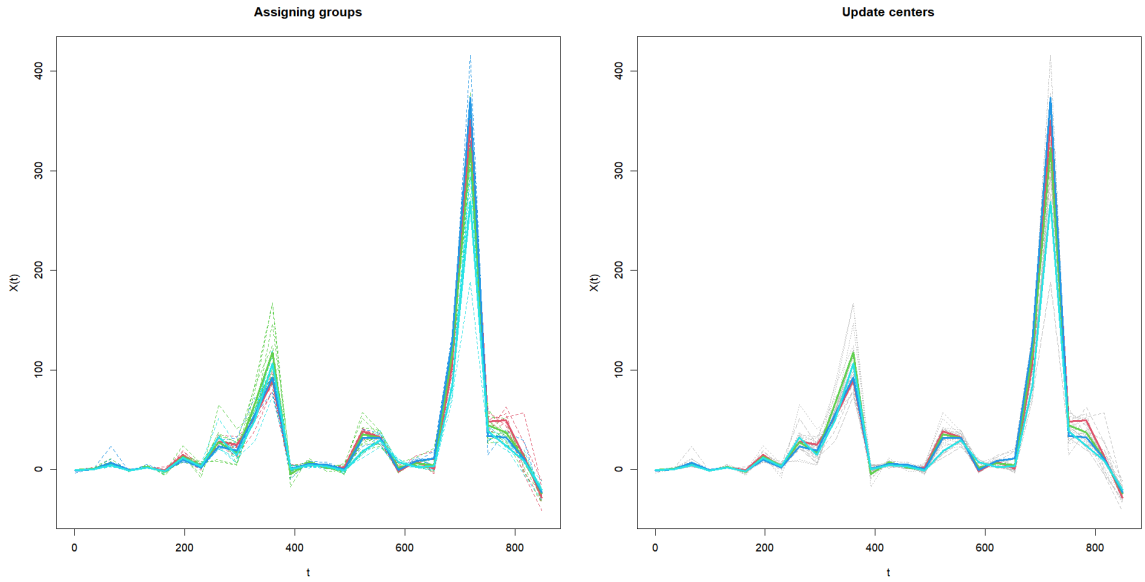
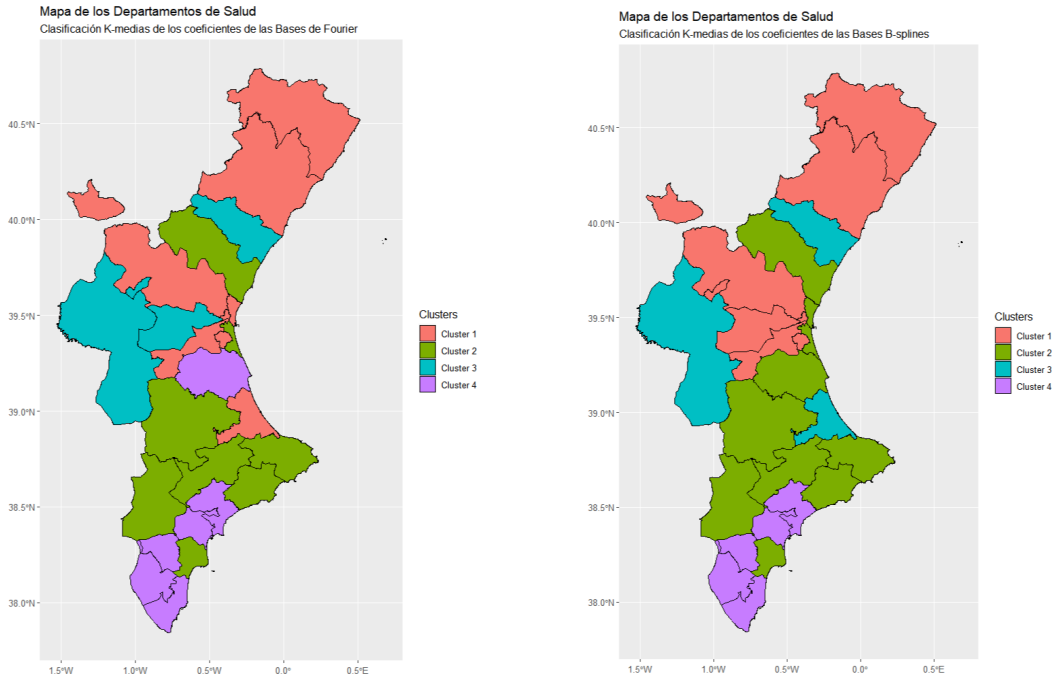


Figura 5.13: K-medias aplicado a las curvas en la Base B-spline. Rojo: *Cluster* 1, Verde: *Cluster* 2, Cian: *Cluster* 3, Azul: *Cluster* 4

Cluster 1	**Cluster 2**
“VINAROS”	“SAGUNT”
“ CASTELLO”	“ VLC CLINIC-LA MALVA-ROSA”
“ VLC ARNAU DE VILANOVA LLIRIA”	“ VLC - DOCTOR PESET”
“ VLC - LA FE”	“ LA RIBERA”
“ VLC -HOSPITAL GENERAL”	“DENIA”
“MANISES”	“XATIVA - ONTINYENT”
	“ALCOI”
	“LA MARINA BAIXA”
	“ELDA”
	“ELX - HOSPITAL GENERAL”
Cluster 3	**Cluster 4**
“LA PLANA”	“ALC-SANT JOAN D’ALACANT”
“REQUENA”	“ ALC - HOSPITAL GENERAL”
“GANDIA”	“ORIHUELA”
	“TORREVIEJA”
	“ELX-CREVILLENT”

Tabla 5.2: Tabla clasificación con K-medias de las curvas de los Departamentos de Salud en Bases B-spline

Para ver la diferencia entre las clasificaciones según las bases hemos representado los *cluster* en el mapa de los departamentos de salud de la Comunidad Valenciana (Fig. 5.14), además nos servirá para localizar su ubicación.



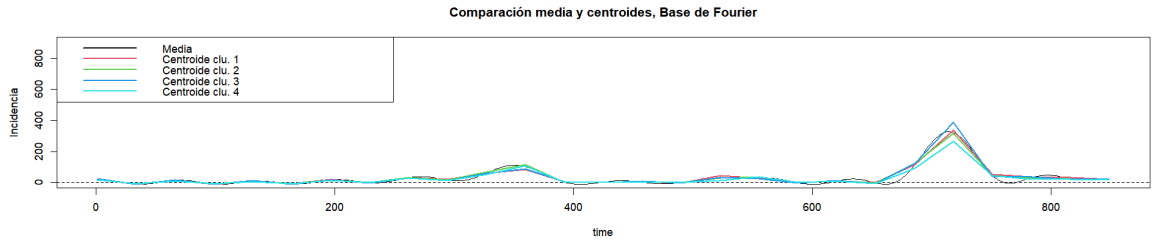
(a) Departamentos clasificados tras aplicar K-medias en las Bases de Fourier

(b) Departamentos clasificados tras aplicar K-medias en las Bases B-splines

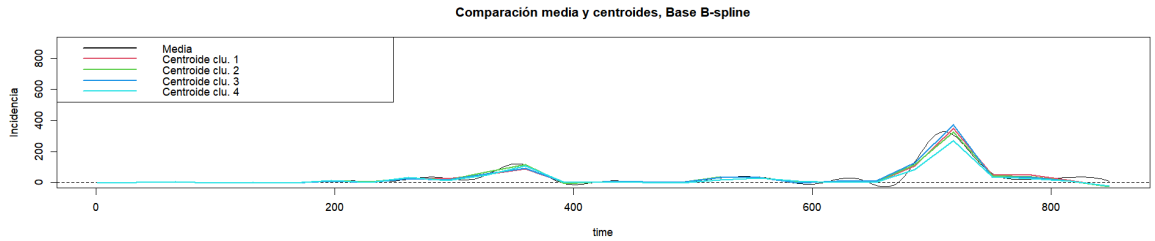
Figura 5.14: Mapa del clustering resultante de aplicar K-medias a las bases

Por último vamos a comparar la media de las funciones con los centroides obtenidos de la clasificación, para ello lo hacemos de forma gráfica en la Figura 5.15.

En las dos gráficas se puede apreciar como los centroides tienen una serie temporal muy parecida, en los dos casos el *Cluster* 2 tiene mayor incidencia en la "ola" con $t \approx 400$ mientras que en la "ola" con $t \approx 700$ no se puede apreciar que grupo tiene mayor incidencia.



(a) Comparación entre centroides y media en Bases de Fourier



(b) Comparación entre centroides y media en Bases B-spline

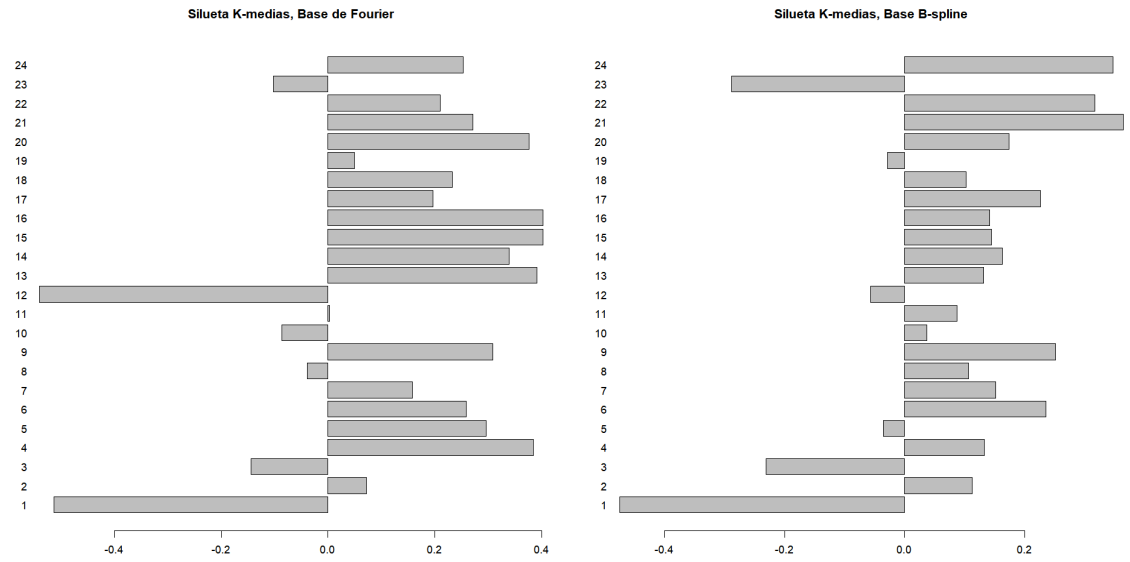
Figura 5.15: Comparación de centroides con la media.

Finalmente hemos implementado una función para obtener la silueta de nuestros *clustering*, esta función nos da como resultado la silueta de cada observación y las representamos en un gráfico de barras. En la Figura 5.16 observamos el resultado de aplicar la función a nuestros *clusters*.

Hemos obtenido dos siluetas diferentes, puesto que el resultado de ambas clasificaciones son diferentes y la silueta sólo tiene en cuenta las distancias y las clasificaciones de las observaciones.

En la Figura 5.16(a) observamos que todas las siluetas están en un intervalo entre el -0.5 y el 0.4, por tanto no podemos deducir si los *clusters* están bien construidos o si por su contrario no se ha realizado una correcta clasificación. Por otra parte tenemos la Figura 5.16(b) donde las siluetas se encuentran en un intervalo de -0.5 a 0.3, por lo que tampoco podemos decir si están bien clasificados.

Comparando las dos siluetas podemos apreciar que ambas siluetas poseen el mismo número de observaciones con siluetas inferiores a 0. De las observaciones negativas de la Figura 5.16(a) debemos destacar la observación 12 y 1 que podríamos decir que están mal clasificadas, mientras que de la Figura 5.16(b) podemos decir que la observación 1 también esta mal clasificada. Además si comparamos las medias de la siluetas obtenemos que los *clusters* con Bases B-spline tiene una silueta media de 0.13 mientras que el otro del 0.09. De esta forma ninguna de las clasificaciones se podría considerar como suficiente.



(a) Silueta *clusters* con las Bases de Fourier (b) Silueta *clusters* con las Bases B-splines

Figura 5.16: Silueta de los *clusters* obtenidos con el algoritmo K-medias

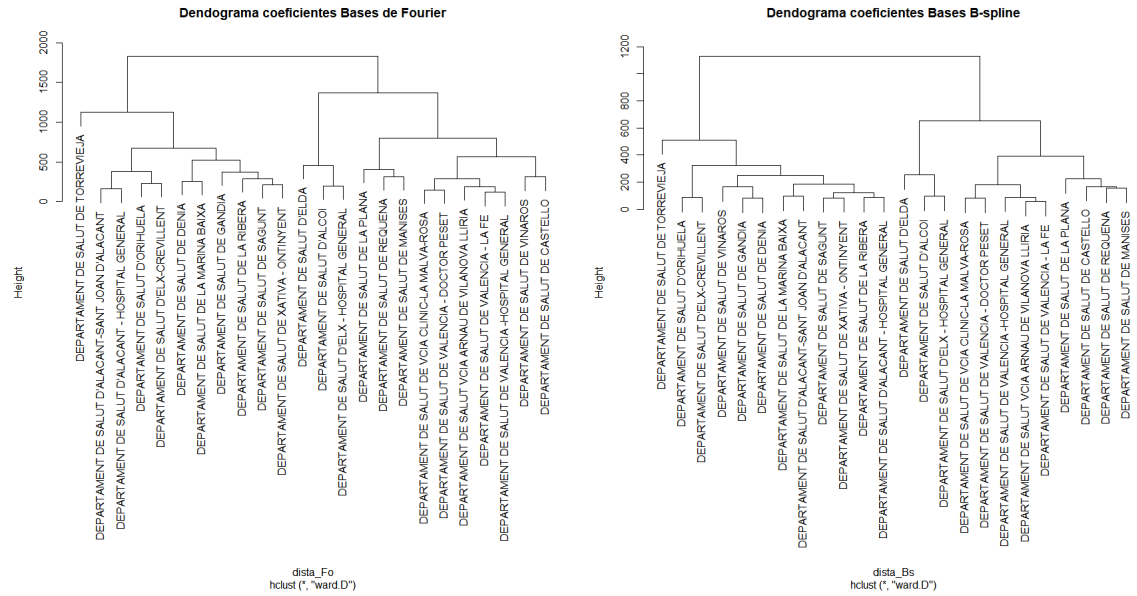
Como ya hemos mencionado las siluetas obtenidas tiene una media que no llega ni a ser suficiente, por tanto decimos realizar el *clustering* para K=3 y obtuvimos una solución análoga, con unas siluetas que no llegaban a ser suficientes. Finalmente, repetimos el proceso para K=2, en este caso si obtuvimos mejores siluetas pero los grupos que se obtuvieron diferenciaron aquellas series temporales que están por encima o por debajo de la media.

Por tanto, podemos concluir que nuestro conjunto de series temporales de las incidencias no posee una estructura clara de grupos.

5.2.2. Algoritmo jerárquico

En esta sección vamos a realizar el *clustering* no supervisado aplicando un algoritmo jerárquico, como ya se ha explicado en la sección 3.3.2. En este caso vamos a realizar la clasificación a partir de los coeficientes estimados para la representación de los datos en las bases seleccionadas. Además vamos a aplicar el algoritmo con la Medida de Vinculación Ward, que hemos explicado en la sección anteriormente mencionada.

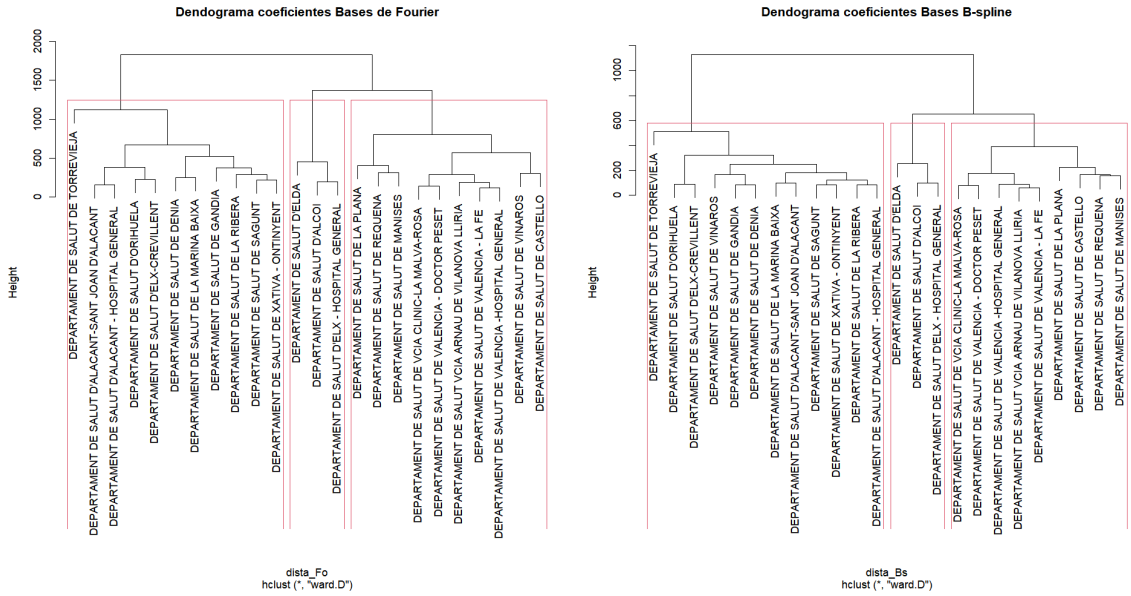
Gracias a la función *hclust* podemos aplicar este algoritmo y obtener su dendograma resultante. En la Figura 3.16 podemos observar los dendogramas resultantes. La Figura 5.17(a) es el árbol resultante de aplicar el algoritmo a los coeficientes de las Bases de Fourier, mientras que en la Figura 5.17(b) vemos el resultado de aplicarlo a los coeficientes de las Bases B-splines.



(a) Dendograma a partir de los coeficientes de las Bases de Fourier

(b) Dendograma a partir de los coeficientes de las Bases B-splines

Figura 5.17: Dendograma de los datos a partir de los coeficientes de las bases



(a) Dendograma con 2 *clusters* a partir de los coeficientes de las Bases de Fourier

(b) Dendograma con 2 *clusters* a partir de los coeficientes de las Bases B-splines

Figura 5.18: Dendograma cortado de los datos a partir de los coeficientes de las bases

Para decidir el número de *clusters* con el que queremos trabajar partimos de la elección que hemos realizado en la sección 5.2.1. En dicha sección habíamos utilizado 4 grupos pero observando las Figuras 5.17 podemos observar que si tomamos ese número de *clusters* tendremos uno que solo poseerá una observación, por tanto vamos a tomar 3 grupos.

En la Figura 5.18(a) observamos que hemos cortado el dendograma a una altura de 1200 para obtener los tres grupos (Tabla 5.3). Mientras que en el dendograma de los coeficientes de las Bases B-splines (Fig. 5.18(b)) hemos cortado en la altura 600 para obtener 3 categorías (Tabla 5.4).

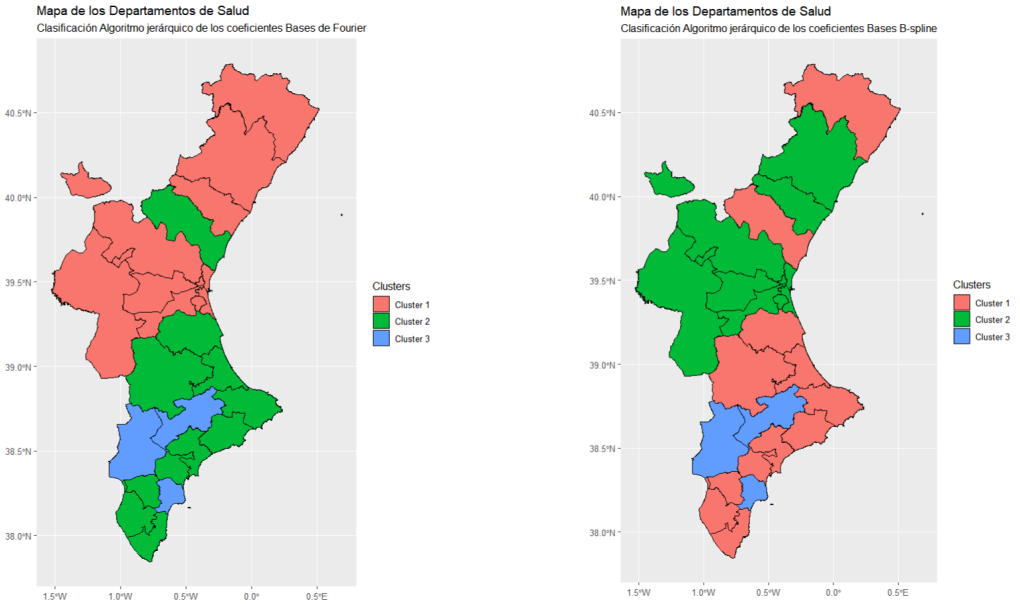
En la Tabla 5.3 están los diferentes departamentos según la clasificación obtenido, en concreto hemos obtenido que el *Cluster* 1 posee 11 series temporales, en el *Cluster* 2 se han clasificado 10 observaciones y en el tercer *Cluster* están los otros 3 departamentos. Por otra parte en la Tabla 5.4 tenemos el resultado de aplicar el algoritmo a las funciones en las Bases B-splines, para el primer *Cluster* se han clasificado 12 observaciones, en el *Cluster* 2 están 9 series temporales y el *Cluster* 3 es igual al obtenido en la Tabla 5.3.

Cluster 1	**Cluster 3**
“TORREVIEJA”	“LA PLANA”
“ALC-SANT JOAN D’ALACANT”	“REQUENA”
“ALC - HOSPITAL GENERAL”	“MANISES”
“ORIHUELA”	“VLC CLINIC-LA MALVA-ROSA”
“ELX-CREVILLENT”	“VLC - DOCTOR PESET”
“DENIA”	“VLC ARNAU DE VILANOVA LLIRIA”
“LA MARINA BAIXA”	“VLC - LA FE”
“GANDIA”	“VLC -HOSPITAL GENERAL”
“LA RIBERA”	“VINAROS”
“SAGUNT”	“CASTELLO”
“XATIVA - ONTINYENT”	
Cluster 2	
“ELDA”	
“ALCOI”	
“ELX - HOSPITAL GENERAL”	

Tabla 5.3: Tabla clasificación del Algoritmo jerárquico de las curvas en Bases de Fourier

Cluster 1	**Cluster 3**
“TORREVIEJA”	“VLC CLINIC-LA MALVA-ROSA”
“ORIHUELA”	“VLC - DOCTOR PESET”
“ELX-CREVILLENT”	“VLC -HOSPITAL GENERAL”
“VINAROS”	“VLC ARNAU DE VILANOVA LLIRIA”
“GANDIA”	“VLC - LA FE”
“DENIA”	“LA PLANA”
“LA MARINA BAIXA”	“CASTELLO”
“ALC-SANT JOAN D’ALACANT”	“REQUENA”
“SAGUNT”	“MANISES”
“XATIVA - ONTINYENT”	
“LA RIBERA”	
“ALC - HOSPITAL GENERAL”	
Cluster 2	
“ELDA”	
“ALCOI”	
“ELX - HOSPITAL GENERAL”	

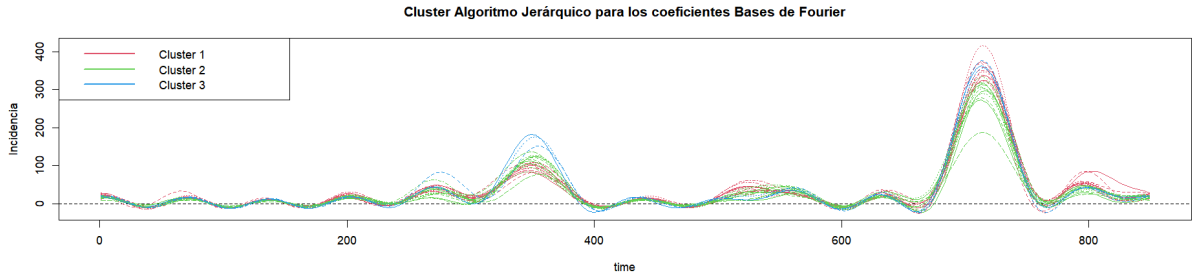
Tabla 5.4: Tabla clasificación del Algoritmo jerárquico de las curvas en Bases Bspline



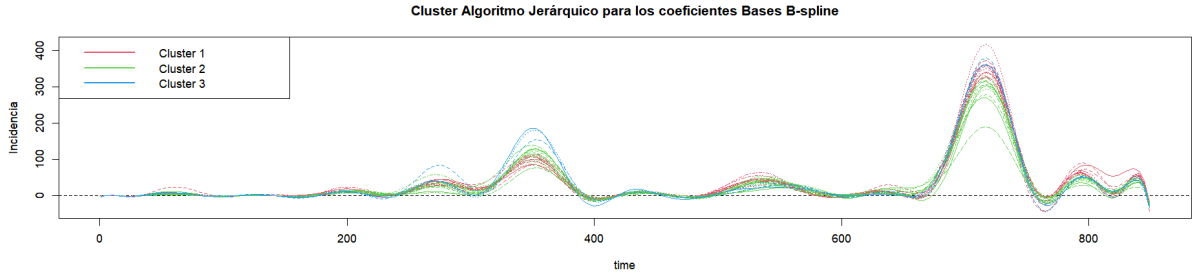
(a) Mapa *clusters* respecto Bases de Fourier (b) Mapa *clusters* respecto Bases B-splines

Figura 5.19: Mapa del *clustering* resultante de aplicar el algoritmo jerárquico respecto a los coeficientes de las bases.

Como se puede ver al comparar las dos tablas (Tab. 5.3 y Tab. 5.4) no hemos obtenidos los mismos *cluster* para las diferentes bases de funciones, en concreto el único departamento en que se diferencian es Vinaroz. Además esto lo podemos observar en la Figura 5.19, en ella podemos ver como queda la división de departamentos según se aplica el algoritmo jerárquico en los coeficientes de las bases. Además en la Figura 5.20 se tiene la clasificación de las curvas, en ambas bases.



(a) Curvas de las Bases de Fourier según *cluster* resultante de aplicar un Algoritmo Jerárquico

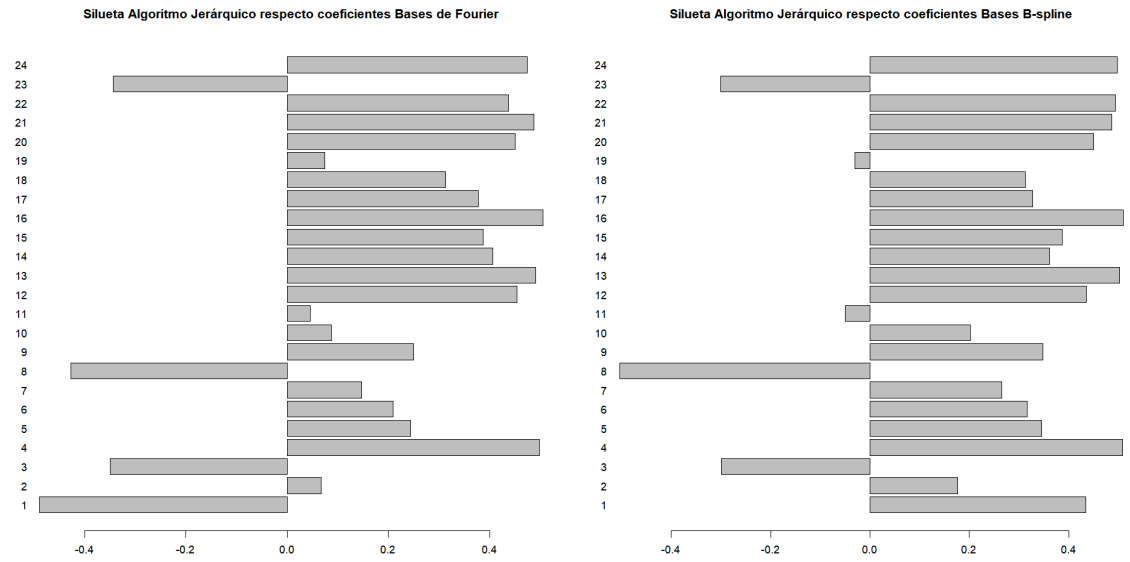


(a) Curvas de las Bases B-spline según *cluster* resultante de aplicar un Algoritmo Jerárquico

Figura 5.20: Representación de los *clusters* de los algoritmos jerárquicos respecto a los coeficientes de las bases.

Al igual que hemos realizado con el anterior modelo de clasificación no supervisada, hemos realizado la silueta de los *clusters* obtenidos, además podemos observar el resultado en la Figura 5.21. En este caso las siluetas no son iguales puesto que los *clusters* son diferentes. De forma general no podemos considerar que haber utilizado una base u otra sea mejor para la clasificación puesto que ambas tienen unas siluetas con valores entre -0.5 y 0.5.

Casualmente podemos apreciar que en las dos gráficas ambas particiones han dado valores aproximados en la silueta. En la Figura 5.21(a) se han obtenido 4 siluetas inferiores a 0 mientras que en la Figura 5.21(b) se han obtenido 5. Si solo nos fijáramos en la cantidad de observaciones con siluetas de signo negativo podríamos deducir que utilizar Bases de Fourier proporciona una mejor clasificación. Pero calculando la media de las siluetas, para las Base de Fourier 0.20 y para las Bases B-spline 0.26, obtenemos que la silueta del algoritmo aplicado a coeficientes en Bases B-spline nos da mejores resultados pero, aún así, solo podemos decir que es una clasificación suficiente.



(a) Silueta *clusters* con las Bases de Fourier (b) Silueta *clusters* con las Bases B-splines

Figura 5.21: Silueta de los *cluster* obtenidos con los algoritmos jerárquicos.

5.3. Clasificación supervisada

En esta sección volvemos a partir las incidencias de los 24 departamentos de salud. Con estos datos realizaremos una clasificación supervisada. Para esta clasificación vamos a trabajar directamente con objetos de la clase del paquete *fda*, puesto que las funciones que usaremos ya interpretan dichos objetos de la forma pertinente.

En la sección 3.3.1 se ha explicado el modelo de clasificación no supervisada que vamos a aplicar: K-NN.

5.3.1. K-NN

En esta sección vamos a realizar el *clustering* supervisado aplicando K-NN, como ya se ha explicado en la sección 3.3.1 partimos de los datos etiquetados. Tomaremos como etiqueta de cada dato, la provincia a la que pertenece, dichas etiquetas también se pueden ver en la Tabla 4.1, donde la etiqueta 1 corresponde a departamentos de salud de la provincia de Castellón, la etiqueta 2 para la provincia de Valencia y la etiqueta 3 para la de Alicante. Podemos ver una representación de la división territorial en la Figura 5.22.

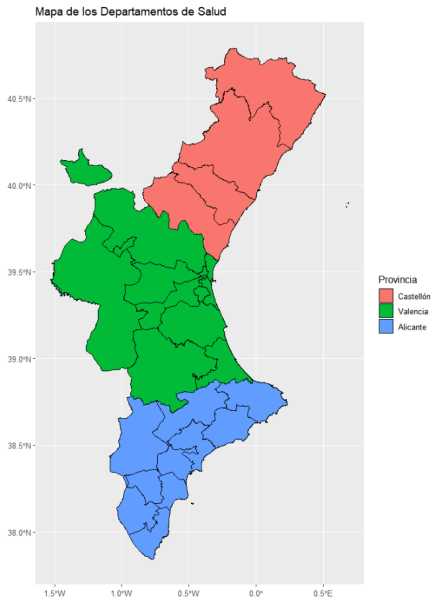


Figura 5.22: Mapa de los Departamentos de Salud de la C.V. según provincia.

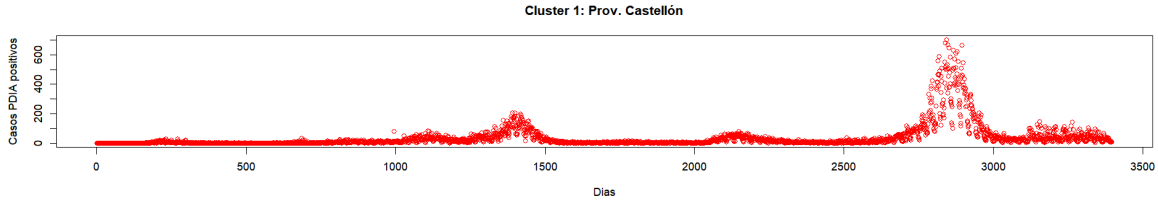
Además en la Figura 5.23 observamos para cada una de las provincias las series temporales de las incidencias de sus departamentos de salud.

Una vez tenemos los datos preparados, la función *classif.knn* del paquete *fda.usc* nos permite entrenar nuestro modelo de clasificación para obtener el valor de K y poder predecir posteriormente. El entrenamiento se realiza con los datos de las incidencias y probando a seleccionar de 2 a 9 vecinos. Y el resultado obtenido esta en la Tabla 5.5, explicado en la sección 3.3.1, en ella se observa que el mayor valor de validación cruzada lo comparten varios valores de K. Puesto que nuestro conjunto de datos no es muy grande además hay un *cluster* con una cantidad de observaciones notablemente inferior al resto vamos a escoger el menor número de K vecinos. Por tanto tenemos que $K = 3$

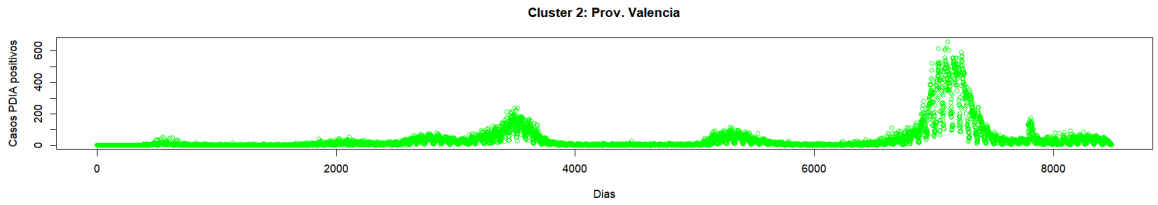
K	2	3	4	5	6	7	8	9
Probabilidad	0.7500	0.8750	0.7500	0.8333	0.7917	0.7917	0.7919	0.8333

Tabla 5.5: Valores de probabilidad para estar bien clasificado para $K = 2, 3, \cdots, 9$ con los datos de los departamentos de salud.

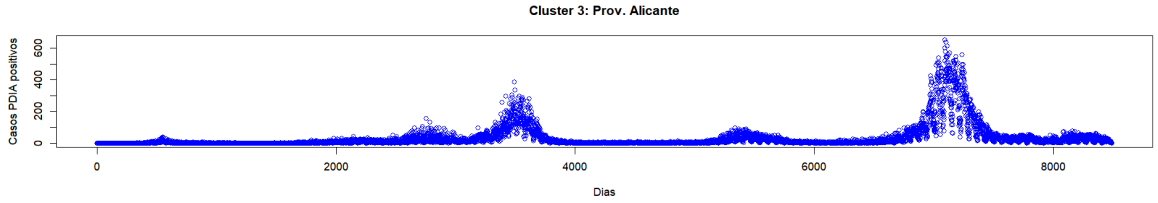
Al igual que hemos realizado en los modelos de clasificación supervisada ahora vamos a revisar la fiabilidad de nuestro modelo. Hemos elegido el número de vecinos óptimo, $K = 3$. De esta forma tenemos que el CV es de 0.875, pero no habíamos explicado como hemos obtenido



(a) Incidencias de los departamentos de Salud de la Provincia de Castellón.



(b) Incidencias de los departamentos de Salud de la Provincia de Valencia.



(c) Incidencias de los departamentos de Salud de la Provincia de Alicante.

Figura 5.23: Representación de las incidencias de las provincias de la C.V.

ese valor.

Para obtener la probabilidad una forma simple es realizar una tabla de clasificación donde las filas representan los *cluster* reales mientras que las columnas hacen referencia al *cluster* predicho. En nuestro caso hemos obtenido la Tabla 5.6, por tanto el número de observaciones correctas se corresponde con la diagonal. De esta forma obtenemos la probabilidad como casos favorables partido casos posibles: $\frac{2+9+10}{24}$. Por tanto 0.875 es la probabilidad de clasificar correctamente una nueva observación, es decir, tenemos una tasa del 87,5 % de clasificar correctamente nuevas series temporales. Además si sumamos los valores de las diferentes columnas por filas, obtenemos el número de observaciones que pertenecen al *cluster* de esa fila.

	1	**2**	**3**
1	2	2	0
2	0	9	1
3	0	0	10

Tabla 5.6: Tabla *cluster* predcido/real

5.4. Implementación de un modelo de clasificación en Shiny

En esta sección vamos explicar como se ha implementado uno de los modelos de clasificación con los que hemos trabajado en un *dashboard*. Además vamos a aprovechar la aplicación web, comentada en la sección 4.2.

El primer paso fue elegir el modelo que mejor se acoplaría a la aplicación web como la que partimos. Por tanto descartamos los modelos de clasificación supervisada y entre los dos modelos explicados de clasificación no supervisada hemos seleccionado K-media.

Una vez teníamos claro que queríamos implementar, añadimos un pequeño formulario para seleccionar el número de *cluster* tomando como referencia la sección 5.2.1. De esta forma vamos a trabajar con 3 o 4 grupos. Además como hemos explicado en la sección 5.1 vamos a trabajar con las series temporales de las incidencias de los diferentes departamentos representados en las Bases de Fourier y en las Bases B-splines.

Como ya hemos trabajado todo lo necesario en la sección 5.2.1 solo hemos tenido pasar dichas funciones a nuestra aplicación web. Gracias a las funciones *reactive* del paquete *Shiny* hemos podido implementar un K-medias interactivo que recalculé los nuevos *clusters* cuando se modifica el atributo del número de grupos.

Para trabajar con la clasificación hemos creado otra aplicación web que contiene todo lo descrito en la sección 4.2 y además hemos añadido la visualización del resultado de la clasificación. Para ello hemos implementado una nueva pestaña en la aplicación web llamada "*Clustering*", la nueva pantalla de inicio de la web la podemos ver en la Figura 5.24.

En la nueva pantalla hemos decidido realizar una comparativa entre la clasificación resultante de aplicar el algoritmo a las curvas en cada una de las bases. Para ello representamos como quedan los grupos en un mapa de la comunidad (Fig. 5.25) y a continuación unas gráficas que comparan los centroides de los grupos con la serie temporal de la incidencia del departamento seleccionado (Fig. 5.26).

De esta forma hemos implementado una nueva aplicación donde podemos ver de forma clara la evolución de los casos PDIA positivos y además una pequeña clasificación de los

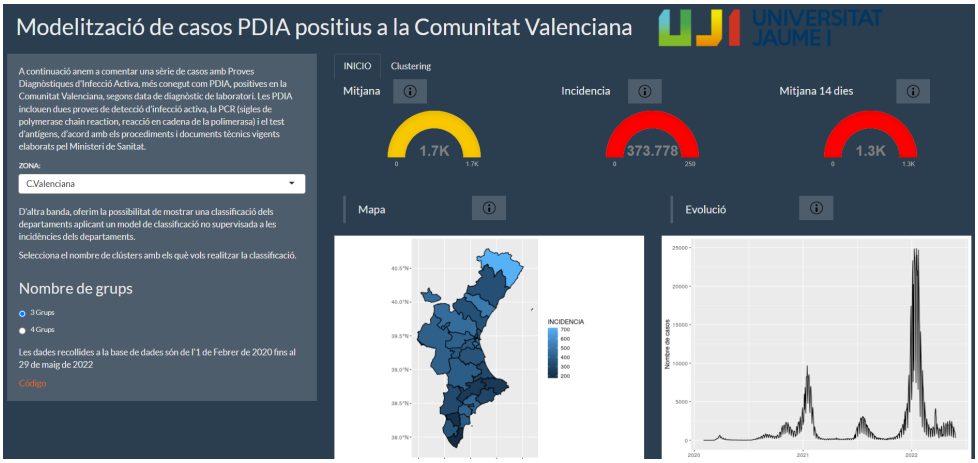


Figura 5.24: Captura de la nueva página web

departamentos según la incidencia. En el siguiente enlace, `https://github.com/juanponsg/Modelitzacio_casos_PDIA_Comunitat_Valenciana.git` podemos obtener el código de la aplicación web y además se puede acceder a ella gracias a que la hemos publicado en el servidor *shinyapps.io* (`https://juan-pg.shinyapps.io/Modelitzacio_casos_PDIA_Comunitat_Valenciana/`).

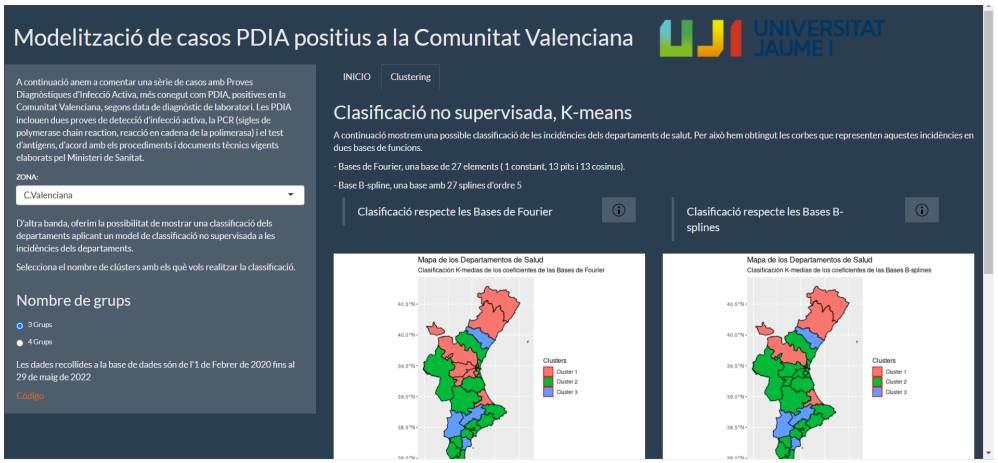


Figura 5.25: Captura de los mapas resultantes del K-medias en la nueva página web para el Departamento de Salud de Torrevieja

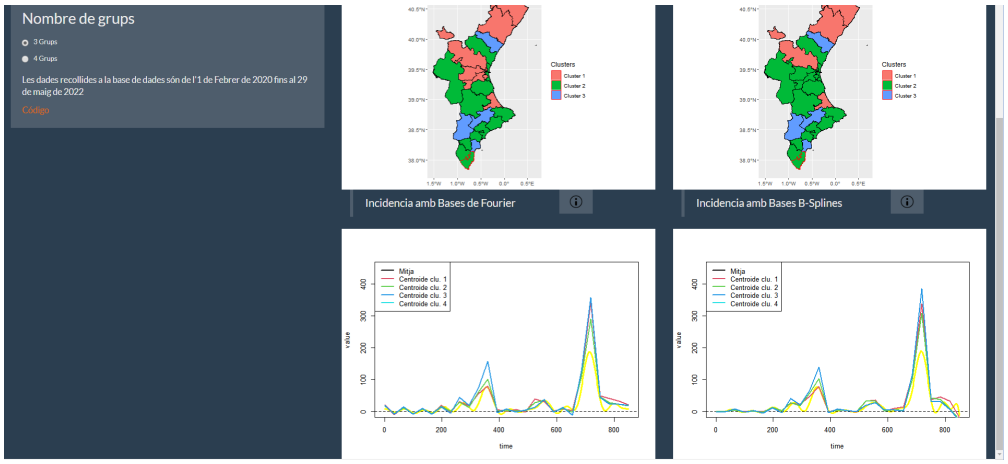


Figura 5.26: Captura de las comparaciones entre centroides y las incidencias resultantes del K-medias en la nueva página web para el Departamento de Salud de Torrevieja

Capítulo 6

Conclusiones

Empezamos este proyecto con la finalidad de introducirnos en el estudio de datos funcionales y entender cómo se trabaja con ellos. Por tanto hemos tenido que entender en qué consiste representar los datos en curvas a partir bases de funciones, para ello la principal fuente de información ha sido el libro "Functional Data Analysis"[6]. Además como hemos querido aplicar nuestra teoría de forma practica hemos recurrido al libro "Functional data analysis with R and MATLAB"[12].

Una vez hemos entendido como obtener nuestras curvas, pasamos estudiar como se implementaban los modelos clásicos de clasificación en el entorno de datos funcionales. Gracias a la parte práctica hemos podido visualizar cada uno de los modelos, además nos ha servido para enseñarnos que debemos saber como trabaja cada función para poder obtener el mejor resultado.

Por otra parte, con esta parte de práctica nos hemos podido dar cuenta lo importante que es la informática en el mundo de las matemáticas y viceversa, puesto que la integración de programas informáticos facilita el cálculo y la aplicación de algoritmos pero sin olvidar que todo esto es posible gracias a las matemáticas.

Bibliografía

[1] J Allaire. rsconnect: Deployment interface for r markdown documents and shiny applications. *R package version 0.8*, 16, 2019.

[2] D Attali and T Edwards. shinyalert: Easily create pretty popup messages (modals) in 'shiny'. *R package version*, 2(0), 2018.

[3] Manuel Febrero Bande, Manuel Oviedo de la Fuente, Pedro Galeano, Alicia Nieto, Eduardo Garcia-Portugues, and Maintainer Manuel Oviedo de la Fuente. Package 'fda. usc'. *CRAN Repository*, 2020.

[4] Winston Chang, Joe Cheng, JJ Allaire, Yihui Xie, and Jonathan McPherson. Package 'shiny'. 2015.

[5] Carl De Boor and Carl De Boor. *A practical guide to splines*, volume 27. springer-verlag New York, 1978.

[6] Frédéric Ferraty and Philippe Vieu. *Nonparametric functional data analysis: theory and practice*, volume 76. Springer, 2006.

[7] Evelyn Fix and Joseph Lawson Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57(3):238–247, 1989.

[8] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[9] Bryan FJ Manly. Stage-structured populations. In *Stage-Structured Populations*, pages 1–6. Springer, 1990.

[10] Daniel Peña. *Análisis de datos multivariantes*, volume 24. McGraw-hill Madrid, 2002.

[11] James O Ramsay and Bernard W Silverman. *Fitting differential equations to functional data: Principal differential analysis*. Springer, 2005.

[12] JO Ramsay, Giles Hooker, and Spencer Graves. Introduction to functional data analysis. In *Functional data analysis with R and MATLAB*, pages 1–19. Springer, 2009.

[13] Generalitat Valenciana. Covid-19 serie de casos con pdia positiva en la comunitat valenciana, según fecha en la que el laboratorio notifica el diagnóstico. https://dadesobertes.gva.es/dataset/ce195af2-39ec-4f44-bb77-b14235519b0d/resource/cb50e7d2-0c0e-46b8-a359-a0fa35998577/download/covid-19-serie-de-casos-con-pdia-positiva-en-la-comunitat-valenciana.csv.

[14] Hadley Wickham, Winston Chang, and Maintainer Hadley Wickham. Package 'ggplot2'. *Create elegant data visualisations using the grammar of graphics. Version*, 2(1):1–189, 2016.