Máster en Matemática Computacional

Proyecto final de máster

Outliers de datos funcionales para la detección de caudales anómalos en el sector hidráulico

---

Autora:
Laura Millán Roures

Tutores académicos:
Irene Epifanio López
Vicente Martínez García

Curso académico 2016/2017

Resumen

Este trabajo está basado en el proyecto llevado a cabo en una beca de investigación para la Cátedra Facsa de Innovación del Ciclo Integral del Agua de la UJI. El estudio realizado consiste, en líneas generales, en la detección de caudales anómalos nocturnos para el descubrimiento de posibles fugas en la red de distribución de agua.

Las técnicas utilizadas para llevar a cabo este proyecto se enmarcan dentro del campo emergente denominado Big Data. Es conocido que durante la última década ha crecido exponencialmente la cantidad de datos generados y almacenados por empresas e instituciones, algunas fuentes, ya en 2011 apuntaban a que el 90 % de los datos existentes se habían generado durante los dos últimos años [1]. Actualmente, las metodologías y arquitecturas basadas en bases de datos tradicionales no son suficientes para gestionar la complejidad que aporta el tratamiento de toda esa inmensa cantidad de datos. En este sentido, la metodología que proponemos no utiliza las técnicas clásicas de la estadística multivariante sino que hemos decidido enfocar el problema desde la perspectiva de datos funcionales, donde, por cada día, en lugar de tener varias variables se tiene una función que representa los valores de caudal registrados en un sector. Se ha decidido utilizar esta técnica, además de por la naturaleza de los datos, por sus numerosas ventajas, entre las que destacan, que puede haber valores faltantes, que no es necesario tener los datos medidos en los mismos instantes de tiempo y que se puede eliminar el ruido en los datos aplicando técnicas de suavizado. Esta última ventaja es sin duda una de las más importantes, ya que el ruido puede distorsionar fácilmente los resultados y, por tanto, las conclusiones obtenidas. Como el objetivo de este trabajo es la detección de caudales anómalos, de entre todos los problemas que se pueden plantear en el análisis de datos funcionales, se ha decidido utilizar técnicas de detección de *outliers* funcionales.

En este documento se recoge tanto la fundamentación teórica de las técnicas utilizadas como los resultados obtenidos para los caudales del consumo hídrico de tres sectores de la provincia de Castellón.

Palabras clave

Análisis de datos funcionales, Detección de *outliers* de datos funcionales, Técnicas de suavizado, Detección de caudales anómalos.

Keywords

Functional Data Analysis (FDA), *Outlier* detection for functional data, Smoothing techniques, Detecting anomalies in water distribution networks.

Índice general

Capítulo 1

Introdución

En este primer capítulo se explica el contexto en el que el proyecto ha sido desarrollado, así como la motivación que ha permitido llevarlo a cabo y los objetivos generales que se han alcanzado durante el desarrollo del proyecto.

En el resto de capítulos se describe la empresa donde se ha realizado el trabajo, las técnicas utilizadas y los resultados obtenidos. En concreto, en el segundo capítulo se detalla la empresa para la que se ha realizado la beca de investigación y el trabajo que se ha realizado para esta institución. En los capítulos tercero y cuarto se describen teóricamente las técnicas utilizadas, en el quinto capítulo se muestran los resultados conseguidos y, por último, se exponen las conclusiones obtenidas.

1.1. Contexto y motivación del proyecto

El trabajo que se describe en este documento es un proyecto realizado para la Cátedra Facsa de Innovación del Ciclo Integral del Agua de la UJI.

Uno de los objetivos principales que debe alcanzar cualquier empresa gestora del ciclo integral del agua, es el control y la disminución de los niveles de pérdidas de agua. El nivel de pérdidas de agua es un indicador de la eficiencia del sistema: elevadas pérdidas de agua reflejan un inadecuado funcionamiento de la red de abastecimiento y una gestión deficiente. Es por ello que uno de los propósitos a conseguir más importantes para cualquier empresa encargada del suministro de agua es el conocimiento, el control y la optimización del nivel de las pérdidas de agua.

Las pérdidas de agua pueden ser de dos tipos: reales o comerciales. Las pérdidas de agua reales se corresponden con las pérdidas físicas de agua que se ocasionan en las diferentes fases del ciclo de abastecimiento de agua, desde los depósitos donde se almacena el agua, hasta en cualquier parte de la red de distribución. Por otro lado, las pérdidas comerciales, no son

pérdidas físicas de agua, sino que es agua consumida por los usuarios pero no controlada, ya que no es registrada por los contadores y, por tanto, tampoco facturada a los usuarios.

Ambos problemas son importantes y necesitan soluciones. Cada año, más de 32.000 millones de metros cúbicos de agua se pierden físicamente en las redes de distribución de agua de todo el mundo. Además, alrededor de 16.000 millones de metros cúbicos son consumidos por los usuarios sin ser registrados ni facturados. Estas circunstancias conllevan grandes pérdidas económicas que superan incluso a las inversiones que se realizan para mejorar los sistemas de abastecimiento [2].

Desarrollar propuestas de mejora para paliar ambos problemas sería un objetivo a conseguir de gran utilidad para la sociedad. Con el agua que se pierde en las redes de distribución, se podría dotar de un servicio de suministro de agua a muchas localidades que a día de hoy todavía no disponen de suministro o tienen problemas para obtenerlo con fiabilidad.

Aunque lograr una reducción absoluta de las pérdidas de agua, tanto reales como comerciales, es imposible de alcanzar, las cifras anteriores muestran la dimensión del problema. Por ello, es necesario tratar de resolverlo porque, a pesar de la gravedad de la situación, muchos abastecimientos tienen como única política de mejora la improvisación y la adaptación continua. Sólo se llevan a cabo soluciones parciales pero no existe un programa donde consten los planes a realizar a largo plazo para lograr una disminución efectiva de las pérdidas de agua.

1.2. Objetivos generales

Debido a la situación descrita en la sección anterior, la Cátedra Facsa de la UJI, ha considerado necesaria la propuesta de una beca de investigación, con el objetivo de incorporar en su equipo a un estudiante del máster en Matemática Computacional.

Los objetivos generales que se pretenden alcanzar durante la realización de este proyecto están relacionados con el problema de la detección de pérdidas de agua reales, es decir, lo que comúnmente se conoce como fugas. Para ello, resulta de especial interés trabajar únicamente con los valores de consumo de agua registrados durante el horario nocturno. Esto es debido a que durante la noche apenas deben existir consumos de los usuarios. De este modo, desviaciones excesivas en los valores registrados pueden indicar que existe algún tipo de anomalía.

La metodología con la que se pretende abordar el problema es la detección de *outliers* de datos funcionales. La diferencia entre trabajar con datos funcionales, en lugar de con una o varias variables, es que se puede tener una función, o incluso más de una, para cada observación de la muestra. Por ello, para llevar a cabo este proyecto, el uso de esta metodología resulta idóneo, ya que el nivel de agua consumido por los usuarios viene dado por una función a lo largo del tiempo.

Además, se ha decidido tratar el problema con esta técnica debido a sus numerosas ventajas, entre las que se pueden destacar las que se describen a continuación:

- **No es necesario disponer de todos los datos.**
  Se permite que haya valores faltantes en los datos registrados. Esto supone una gran ventaja, ya que la presencia de valores faltantes suele ser muy común en todos los ámbitos. En este caso, Facsa cuenta con un sistema de telecontrol para la recogida de datos que envía la información obtenida desde la estación donde se ubica el contador hasta el centro de recogida de datos mediante Ethernet, GPRS, ondas de radio, etc., por lo que en ocasiones, es bastante frecuente que la red caiga o que el contador no registre correctamente algunos valores. Como trabajar con datos funcionales no requiere disponer de todos los datos, se evita de este modo tener que imputar valores, lo que podría conllevar una pérdida de precisión en los resultados obtenidos.
- **No es preciso tener los datos medidos en un mismo instante de cada día.**
  Esto también resulta útil porque Facsa dispone de datos de caudal con una frecuencia que puede ir desde cada pocos segundos a cada quince minutos.
- **Se puede eliminar el ruido de los datos aplicando técnicas de suavizado.**
  A menudo, el ruido puede distorsionar fácilmente los resultados obtenidos por lo que eliminarlo previamente, antes de utilizar los datos, suele ser una medida muy conveniente para obtener conclusiones adecuadas.

Utilizando técnicas para obtener *outliers* de datos funcionales, el objetivo principal consiste en detectar caudales anómalos nocturnos que se puedan corresponder con la presencia de fugas en la red de abastecimiento. Se analizan los valores registrados para tres sectores de la provincia de Castellón durante los últimos años.

Capítulo 2

La empresa Facsa

2.1. Introducción

En este capítulo se explica el proyecto realizado. En primer lugar, se indica para qué empresa se ha llevado a cabo y se describe cuál es su funcionalidad. A continuación, se detalla el software del que se ha hecho uso para elaborar este proyecto y, finalmente, el trabajo realizado en líneas generales. Los resultados obtenidos, se muestran con más detalle en el Capítulo 5.

2.2. Descripción de la empresa

La institución para la que se ha desarrollado este proyecto es la Cátedra Facsa de la UJI. Facsa se fundó en Castellón en el año 1873 con el objetivo de dotar a la capital de la provincia de una moderna red de distribución de agua potable. Desde entonces, ha ampliado sus actividades y consolidado su presencia en varias comunidades autónomas: Comunidad Valenciana, Aragón, Murcia, Castilla-La Mancha, Navarra, La Rioja y Baleares, convirtiéndose en una empresa de referencia en el sector del agua, llegando a suministrar agua potable a 1.000.000 personas en 70 poblaciones cada día.

Facsa ofrece todos los servicios propios del ciclo integral del agua: captación, potabilización, tratamiento, distribución y posterior recogida y depuración de las aguas residuales.

Como clave de su éxito, Facsa apuesta por la continua actualización de conocimientos de sus empleados, un equipo integrado por más de 800 profesionales multidisciplinares, para poder ofrecer un servicio de alta calidad enfocado a aportar respuestas y soluciones eficientes.

Facsa considera como uno de sus principales objetivos estratégicos, trabajar de forma profesional y constante para lograr la plena satisfacción del cliente. Asimismo, promueve, facilita e incentiva la colaboración y el trabajo en equipo de las personas que forman parte de la compañía. Además, otro de sus objetivos consiste en ofrecer un servicio de calidad sin perjuicio alguno del medio ambiente. Para ello, trabaja con los diversos grupos de interés en defensa de un desarrollo sostenible [3].

Los departamentos con los que cuenta la empresa se muestran en la Tabla 2.1.

DEPARTAMENTOS DE FACSA
Contratación
Compras
Verificación de contadores
Contabilidad
Facturación
Inspección de lectores
Informática
Expansión y desarrollo
Innovación y proyectos TIC
Tratamiento de aguas residuales
Calidad
Técnico: - Captaciones - Mantenimiento - Delineación - Abastecimiento
Comercial
RRHH
Dirección

Tabla 2.1: Departamentos de Facsa.

Facsa pertenece a un grupo de empresas denominado Grupo Gimeno que se divide en tres sectores: Gimeno Servicios, Gimeno Construcción y Gimeno Turismo y Ocio. Fruto de los esfuerzos y de la gran capacidad innovadora, Grupo Gimeno tiene más de 30 empresas que operan en todo el territorio nacional con más de 4.100 profesionales que forman parte de su equipo humano.

Las empresas de Grupo Gimeno además de operar en todo el territorio nacional también lo hacen a nivel internacional a través de sus actuaciones en áreas como Latinoamérica y Arabia Saudí. A nivel nacional, dispone de delegaciones en Castellón, Alicante, Valencia, Teruel, Murcia, Madrid, Barcelona, Mallorca, Zaragoza, Toledo, Sevilla y Pontevedra [4].

En la Figura 2.1 se indican las empresas que pertenecen a Grupo Gimeno.

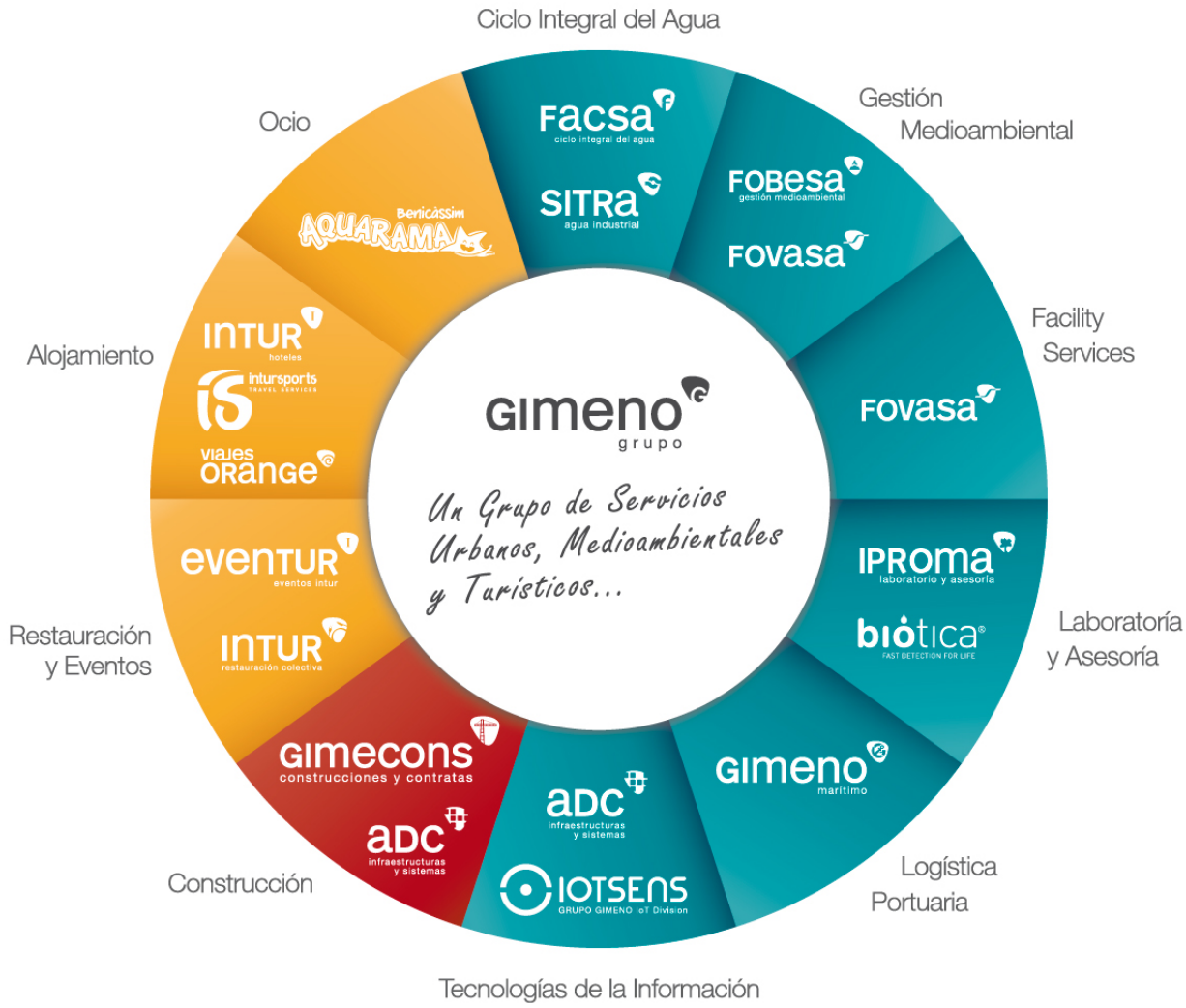


Figura 2.1: Empresas que constituyen el Grupo Gimeno.

Las empresas de la Figura 2.1 representadas de color azul se corresponden con el sector Gimeno Servicios, las de color rojo con el sector Gimeno Construcción y, finalmente, las de color naranja con el sector Gimeno Turismo y Ocio.

En la Figura 2.2 se muestra la localización de las diferentes delegaciones de las empresas que pertenecen al Grupo Gimeno.

Figura 2.2: Localización de las delegaciones de las empresas que pertenecen a Grupo Gimeno.

2.3. Descripción del software utilizado

La herramienta utilizada para llevar a cabo este proyecto ha sido el programa **R**. Es un software libre que permite realizar análisis estadísticos de datos. Se ha escogido este programa debido a la gran variedad de métodos estadísticos que contiene ya programados en diferentes bibliotecas y paquetes [5]. En concreto, se ha utilizado RStudio, una interfaz para R que contiene una serie de herramientas integradas y diseñadas para ayudar al usuario a hacer uso del programa R de forma más práctica y cómoda.

2.4. Descripción del trabajo realizado

La red de distribución de un municipio se divide en sectores donde, en cada sector, el caudal de entrada se registra mediante caudalímetros. En la Figura 2.3 se puede observar un ejemplo de la división de un municipio en distintos sectores.

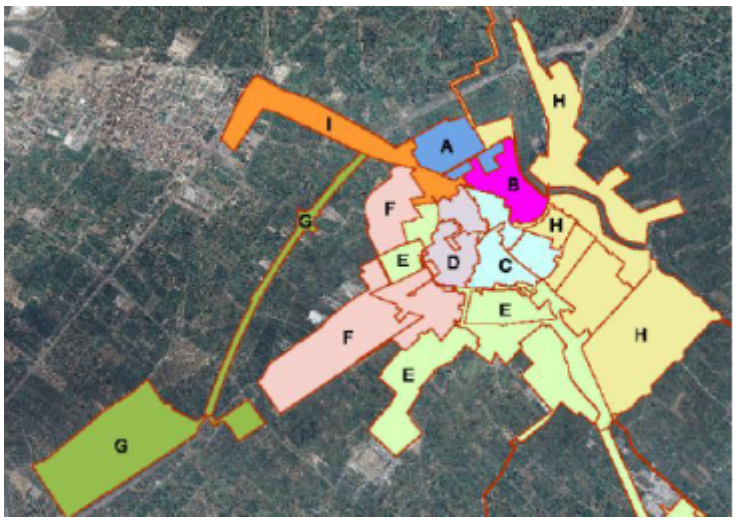


Figura 2.3: División de un municipio en sectores.

El trabajo realizado consiste en, dados tres sectores de la provincia de Castellón, emplear técnicas de detección de *outliers* funcionales para obtener caudales anómalos.

Por motivos de confidencialidad, se omite la localización de estos sectores y se denotan mediante las letras A, B y C. La información de la que se dispone, de cada uno de los sectores, es la relativa a los niveles de caudal registrados durante los años 2014, 2015 y 2016. El tiempo entre observaciones es de cada 5 minutos logrando un total de 288 observaciones cada día. Notemos que estos datos se enmarcan en problemas de alta dimensión y que no estamos tratando con la clásica estadística multivariante sino con datos funcionales donde se tiene una función para cada día. En la Tabla 2.2 se observa un ejemplo de los primeros datos registrados para el sector A.

Fecha	Caudal
01/01/2014 0:00	40.30
01/01/2014 0:05	39.84
01/01/2014 0:10	38.14
01/01/2014 0:15	40.82
01/01/2014 0:20	39.28
01/01/2014 0:25	38.78
01/01/2014 0:30	38.5
01/01/2014 0:35	38.24
01/01/2014 0:40	39.58
01/01/2014 0:45	38.14
01/01/2014 0:50	38.76
01/01/2014 0:55	37.62

Tabla 2.2: Primeros datos registrados del sector A.

Únicamente se van a considerar las observaciones en horario nocturno (de 00:00 a 06:00h) porque durante el día el consumo de agua suele variar. Sin embargo, es durante la noche cuando apenas existen consumos por parte de los usuarios, por lo que el análisis de estos valores nocturnos, facilita la detección de fugas en la red de distribución de agua.

En la Figura 2.4 se representan los valores de caudal registrados cada 5 minutos para el sector A, el día 01/01/2014, durante el horario nocturno.

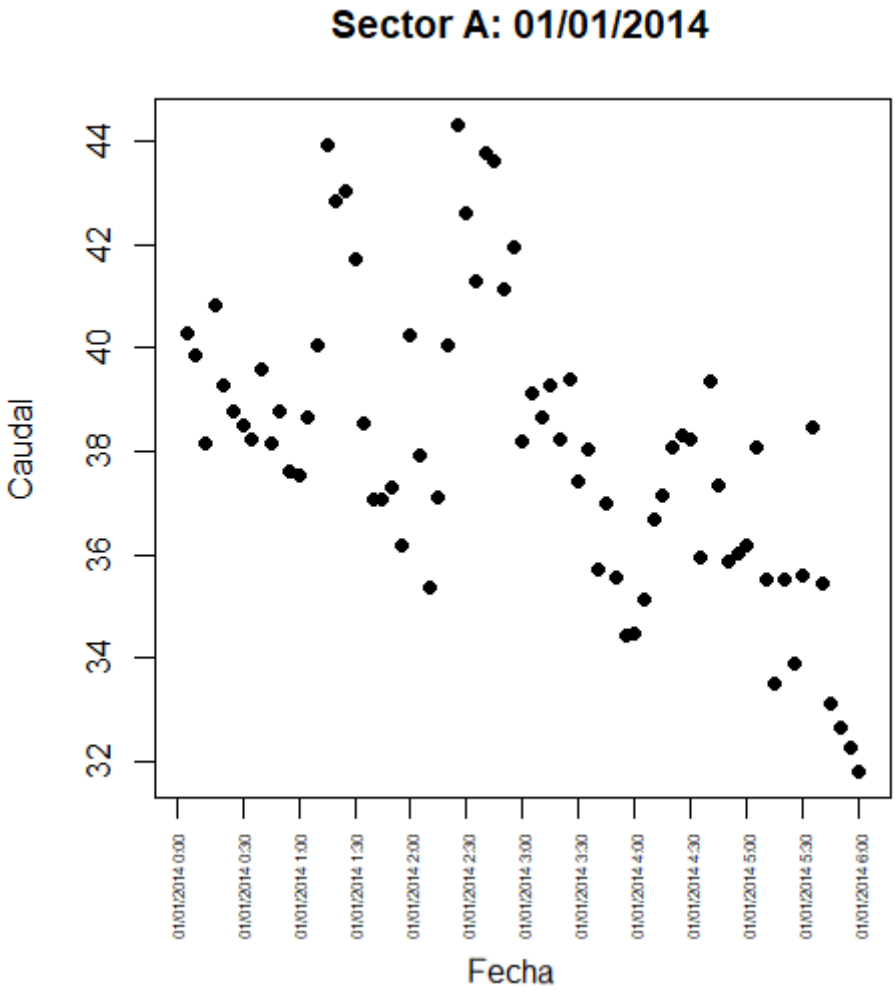


Figura 2.4: Representación de los valores de caudal registrados cada 5 minutos para el sector A, el día 01/01/2014, durante el horario nocturno.

Aunque las observaciones hayan sido recogidas de manera discreta (un dato cada 5 minutos), en este caso, el caudal diario viene dado por una función a lo largo del tiempo. Haciendo uso de una interpolación lineal, la función de caudal para el día 01/01/2014 del sector A se representa en la Figura 2.5.

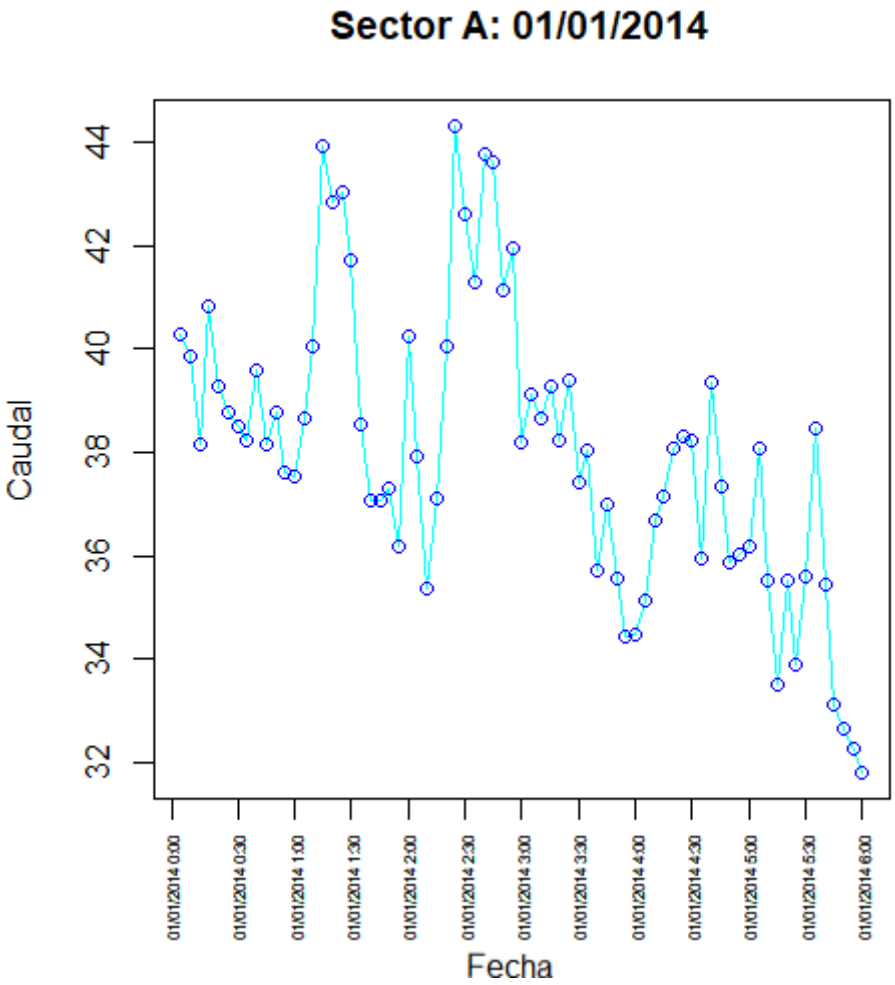


Figura 2.5: Interpolación lineal de los valores de caudal registrados cada 5 minutos para el sector A, el día 01/01/2014, durante el horario nocturno.

Como el caudal diario viene dado por una función, la búsqueda de caudales anómalos nocturnos se puede abordar como un problema de detección de *outliers* pero de funciones. Para lograr este objetivo, se ha decidido dividir la muestra separando los datos obtenidos por estaciones y también por días entre semana y días de fin de semana, pues, dependiendo de la estación del año y de si se trata de un día laboral o de un día de fin de semana, se ha detectado un patrón de consumo distinto. Por ejemplo, en una zona de playa en verano y en un día de fin de semana, el consumo será mayor que en esa misma zona durante un día del mes de invierno. De este modo, se han obtenido 8 grupos a analizar por cada uno de los sectores.

Para la detección de *outliers* de funciones, las instrucciones que se utilizan del software R son *fbplot* del paquete *fda* y *foutliers* del paquete *rainbow*. Además, también se hace uso del análisis de arquetipos con el objetivo de detectar *outliers*. Los resultados de su aplicación a tres sectores de la provincia de Castellón se muestran con detalle en el Capítulo 5.

Capítulo 3

Introducción al análisis de datos funcionales

3.1. Introducción

Los grandes avances llevados a cabo en la ciencia y en la tecnología durante los últimos años, han permitido que en muchos campos científicos como la meteorología, la medicina, etc., se almacenen grandes cantidades de información. De acuerdo con algunos de los estadísticos más reconocidos, es la era del Big Data (B. Efron y T.Hastie 2016 [6]), el Big Data marcará el futuro de la estadística (I.L Dryden y J.T. Kent 2015 [7]), no es una tendencia (Peña 2014 [8]). Generalmente, la información recopilada en el ámbito científico, es recogida de manera discreta, es decir, se recogen observaciones evaluadas cada cierto instante de tiempo. No obstante, el conjunto formado por estos datos se puede ver a menudo como una función continua de una variable a lo largo del tiempo, lo que permite conocer el valor de esa variable en cualquier instante, logrando de este modo, dotar al análisis posterior de mayor completitud y veracidad.

Este tipo de datos se conoce con el nombre de datos funcionales. Algunos ejemplos en los que es útil trabajar con datos funcionales pueden ser: el análisis del aumento de la estatura de una población a lo largo de los años, la variación de un indicador económico, el estudio de la temperatura anual, etc.

El análisis de datos funcionales (FDA) es un campo bastante reciente (el primer libro fue publicado en 1997) por Ramsay y Silverman. Un libro excelente para seguir este capítulo es la publicación realizada en el año 2005 por estos dos autores [9]. También se puede consultar Ferraty y Vieu (2006) [10], donde se proporciona una visión complementaria sobre métodos no paramétricos para datos funcionales.

3.2. Descripción de los datos funcionales

La idea básica del análisis de datos funcionales consiste en tratar los datos observados como tan sólo una observación, en lugar de como una sucesión de observaciones individuales. El término funcional hace referencia a la estructura interna de los datos en lugar de a su forma explícita, es decir, se asume la existencia de una función que da lugar a los datos observados.

En la práctica, cada dato funcional se recoge directamente como p pares (t_j, y_j) donde y_j es el valor observado en el tiempo t_j, posiblemente afectado por un error observacional que comúnmente se conoce como ruido. Como las muestras se componen de un conjunto de datos funcionales, cada individuo de la muestra es una función x_i constituida por p_i pares de la forma $(t_{ij}, y_{ij}), j = 1, \ldots, p_i$. Los valores t_{ij} pueden ser diferentes para cada observación al igual que el intervalo T en el que se recogen todos los datos.

En general, el tiempo suele ser la variable continua sobre la que se registran los datos funcionales; sin embargo, también se pueden registrar sobre otras variables como pueden ser la posición, la frecuencia, etc.

Además, también es necesario que las funciones sean suaves, de manera que un par de valores de datos adyacentes, y_j e y_{j+1} es poco probable que sean muy diferentes el uno del otro. Si este fuera el caso no se ganaría nada tratando los datos como funciones en lugar de simplemente como datos multivariantes. En general, por suavidad quiere decir que las funciones posean una o más derivadas. Sin embargo, los datos observados pueden no ser del todo suaves debido al error observacional. Cada observación y_{ij} viene dada por:

$$y_{ij} = x_i(t_{ij}) + \epsilon_{ij}$$

donde $x_i(t_{ij})$ es el valor real de la función x_i en el tiempo t_{ij} y ϵ_{ij} es el error de medición. Por lo tanto, una de las tareas que se deben realizar al representar los datos discretos como funcionales es filtrar ese ruido, que puede distorsionar los resultados fácilmente. Para ello, se aplican técnicas de suavizado, es decir, se ajustan los datos a una base.

3.3. Ejemplo de datos funcionales

Para entender el concepto de datos funcionales, se muestra un ejemplo con los datos registrados por la AEMET (Agencia Estatal de Meteorología) sobre la temperatura media mensual en Castellón de la Plana durante los años 2013, 2014, 2015 y 2016. La Tabla 3.1 contiene los datos recogidos por cada uno de los meses.

Año	Meses											
	1	2	3	4	5	6	7	8	9	10	11	12
2013	12.0	*	14.4	15.2	17.7	21.9	26.3	25.4	23.5	21.5	14.5	11.1
2014	12.5	12.7	14.1	17.9	19.5	23.4	25.2	23.1	24.9	21.2	15.9	11.6
2015	11.1	11.3	14.0	16.2	20.9	24.0	27.8	26.3	22.8	19.0	15.6	13.1
2016	10.9	13.5	13.6	16.5	18.8	23.4	26.1	26.0	24.1	20.4	14.7	12.7

Tabla 3.1: Temperaturas mensuales de Castellón de la Plana desde 2013 hasta 2016.

Sin embargo, aunque las observaciones se recojan de manera discreta (un dato por mes), resulta más interesante interpretar la temperatura media anual como una función a lo largo del tiempo en lugar de como un vector de observaciones recogidas en tiempo discreto. En la Figura 3.1 se muestra un ejemplo de las posibles funciones de temperatura media de cada año en base a las observaciones registradas.

Considerar las réplicas de la función de temperatura media durante varios años permite conocer cómo varía la temperatura a lo largo del año: la temperatura durante los primeros meses y los últimos del año es generalmente más baja que durante el resto de meses, alcanzando su máximo valor durante los meses de Julio y de Agosto. Si se consideraran periodos de tiempo más largos, también se podrían realizar estudios de cambio climático, comparando las temperaturas de los últimos años con el estándar de comportamiento histórico de los datos.

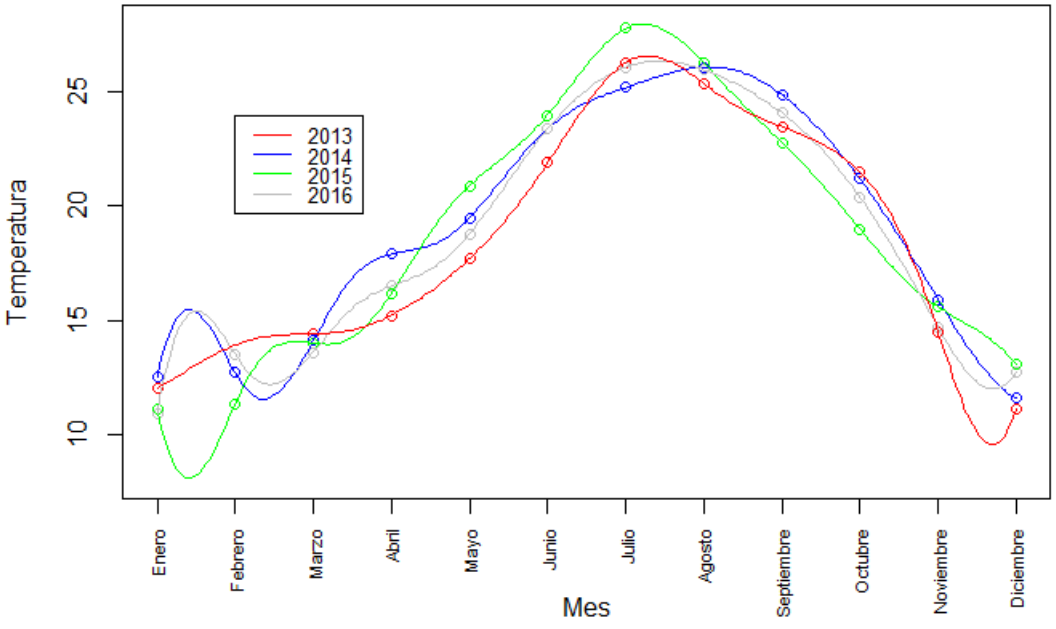


Figura 3.1: Temperaturas medias por meses en Castellón de la Plana desde el año 2013 hasta el año 2016.

Se ha realizado una interpolación utilizando bases B-spline (explicadas en la sección 3.6.2) de orden cuatro y con el número de bases igual a doce (número de elementos que contiene cada observación) para representar las funciones a partir de los datos. Se observan ligeras oscilaciones en la unión de los datos observados. Esto se debe al tipo interpolación realizada. En capítulos posteriores se analiza este tipo de comportamiento y la conveniencia de realizar ajustes en los datos observados y/o en el tipo de interpolación utilizada. El objetivo es obtener funciones que representen de la manera más adecuada posible la distribución de los datos.

3.4. Objetivos del análisis de datos funcionales

Los objetivos del análisis de datos funcionales son los mismos que en cualquier otra rama de la estadística:

- Representar los datos de manera que faciliten un análisis posterior.
- Mostrar los datos para resaltar varias características.
- Estudiar patrones en los datos.
- Explicar el valor de una variable dependiente en función de varias variables independientes.
- Crear grupos de observaciones similares (clasificación no supervisada).
- Clasificar una observación en un grupo donde los grupos están predefinidos a priori (clasificación supervisada).
- Detectar datos atípicos (*outliers*).

Este trabajo se centra fundamentalmente en la detección de datos atípicos. Un dato atípico es una observación que no parece corresponderse con el resto de observaciones del conjunto de datos objeto de estudio. El manejo de este tipo de datos es uno de los problemas interesantes que se presentan en el análisis de datos.

Las palabras 'no parece corresponderse' de la definición anterior ponen de relieve la relativa subjetividad del análisis de estos datos y plantea la duda de si esa observación, que no parece corresponderse con el resto de observaciones, es realmente un valor atípico. Por lo general, el tratamiento de la información se realiza utilizando los datos atípicos y se comparan las conclusiones con otros tratamientos que no los utilizan. Si éstas son diferentes, se debe realizar un análisis más detallado de los mismos. En secciones posteriores, se analizan las técnicas estadísticas disponibles para la detección y tratamiento de datos atípicos.

3.5. Estadísticos para datos funcionales

Los estadísticos para datos funcionales se obtienen generalizando las definiciones de los estadísticos para muestras multivariantes.

Dada una muestra de datos funcionales generada por las funciones $x_i(t), i = 1, \ldots, n$, los estadísticos vienen dados por las ecuaciones representadas en la Tabla 3.2.

Estadístico	**Definición**
Media	$$\overline{x}(t) = \frac{1}{n}\sum_{i=1}^{n} x_i(t)$$
Varianza	$$var_x(t) = \frac{1}{n-1}\sum_{i=1}^{n}[x_i(t) - \overline{x}(t)]^2$$
Desviación típica	$$sd_x(t) = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}[x_i(t) - \overline{x}(t)]^2}$$
Covarianza	$$cov_x(t_1, t_2) = \frac{1}{n-1}\sum_{i=1}^{n}[x_i(t_1) - \overline{x}(t_1)][x_i(t_2) - \overline{x}(t_2)]$$
Correlación	$$corr_x(t_1, t_2) = \frac{cov_x(t_1, t_2)}{\sqrt{var_x(t_1)var_x(t_2)}}$$

Tabla 3.2: Estadísticos para datos funcionales.

3.6. Representar funciones utilizando bases de funciones

El primer paso a realizar una vez recogidas las observaciones de manera discreta es determinar las funciones que representan a cada individuo. Es decir, a partir de los pares de valores

(t_{ij}, y_{ij}) correspondientes a cada individuo de la muestra x_i con $i = 1, \ldots, n$ donde n es el número de individuos de la muestra, se debe representar cada elemento con respecto a un sistema de funciones base.

Un sistema de funciones base es un conjunto de funciones conocidas ϕ_k independientes unas de otras. Permiten expresar una función $x(t)$ por una combinación lineal de funciones del siguiente modo:

$$x(t) = \sum_{k=1}^{K} c_k \phi_k(t)$$

donde K es el número de funciones base ϕ_k utilizadas. Por tanto, las funciones base tienen la propiedad de poder aproximar arbitrariamente bien cualquier función si se toma un número grande, K, de esas funciones y se escogen adecuadamente los pesos c_k de la combinación lineal. Expresándolo de manera matricial, si $\boldsymbol{c}$ es el vector de tamaño K formado por los coeficientes c_k y $\boldsymbol{\phi}$ es el vector formado por las funciones ϕ_k, podemos expresar la ecuación anterior como:

$$x = \boldsymbol{c}^{\boldsymbol{t}} \boldsymbol{\phi} = \boldsymbol{\phi}^{\boldsymbol{t}} \boldsymbol{c}$$

donde el superíndice t indica el vector transpuesto.

Cuando $K = p$ se lleva a cabo la interpolación lineal en el sentido que se escogen los coeficientes c_k para que si $x(t)$ es la función que representa un dato funcional, entonces, $x(t_j) = y_j$ para cada j. Por lo tanto, el grado en el que los datos y_j se suavizan en oposición a la interpolación viene dado por el número de funciones base, K, que se escogen de acuerdo a las características de los datos. Idealmente, las funciones base deberían tener características que coincidan con las propiedades de las funciones que se quieren estimar. Esto hace más fácil alcanzar una aproximación satisfactoria con un número pequeño K de funciones base que, entre otras ventajas, conlleva un menor coste computacional.

Los sistemas de funciones bases más conocidos son las bases de Fourier y las bases B-spline. El primero se utiliza para describir datos periódicos, mientras que el segundo se puede utilizar aunque los datos no sean periódicos. A pesar de que estas dos bases sean las más utilizadas, también hay otras bases que se deben tener en consideración como son las bases exponenciales, las bases de potencias, las bases polinómicas, las ondículas, etc.

3.6.1. Base de Fourier

Uno de los sistemas de bases de funciones más conocidos viene dado por las series de Fourier. En ese caso, las funciones base son

$$1, \sin \omega t, \cos \omega t, \sin 2\omega t, \cos 2\omega t, \ldots \tag{3.1}$$

Las funciones base quedan definidas por

- $\phi_0(t) = 1$,

- $\phi_{2r-1}(t) = \sin r\omega t$,
- $\phi_{2r}(t) = \cos r\omega t$.

Luego, una función $x(t)$ se puede aproximar por el valor de $\widehat{x}(t)$ donde

$$\widehat{x}(t) = c_0 + c_1 \sin \omega t + c_2 \cos \omega t + c_3 \sin 2\omega t + c_4 \cos 2\omega t + \dots$$

Se trata de una base periódica donde el parámetro ω se conoce como la frecuencia de la señal. Indica el número de veces que la señal pasa por un punto en la unidad de la variable independiente de la señal. Se puede medir en hertzios (Hz = vueltas/segundo) o en radianes por segundo. Determina el periodo $\frac{1}{r\omega}$ si se mide en hertzios o $\frac{2\pi}{r\omega}$ si se mide en radianes por segundo.

A continuación, se muestra que las funciones que componen la serie de Fourier, que se han representado en radianes/segundo en la ecuación (3.1), son funciones periódicas de periodo $\frac{2\pi}{\omega}$:

- $\sin(r\omega(t + \frac{2\pi}{r\omega})) = \sin(r\omega t + \cancel{r\omega}\frac{2\pi}{\cancel{r\omega}}) = \sin(r\omega t + 2\pi) = \sin r\omega t$,
- $\cos(r\omega(t + \frac{2\pi}{r\omega})) = \cos(r\omega t + \cancel{r\omega}\frac{2\pi}{\cancel{r\omega}}) = \cos(r\omega t + 2\pi) = \cos r\omega t$.

Cuando los instantes de tiempo, t_j, en los que han sido registrados los valores de la muestra están equiespaciados en el intervalo de tiempo con el que se trabaja, T, y el periodo es igual a la longitud del intervalo T, la base es ortonormal (dividiendo por las constantes adecuadas: $\sqrt{p}$ para $j = 0$ y $\sqrt{p/2}$ para el resto de valores de j). Es decir, si se define Φ la matriz que contiene los valores de las K funciones base, $\Phi^t\Phi = \mathbf{I}$ donde el superíndice t denota la transpuesta y $\mathbf{I}$ es la matriz identidad.

Para calcular los coeficientes c_k de la serie de Fourier se puede utilizar el algoritmo de la Transformada Rápida de Fourier (FFT) cuando p es una potencia de 2 y los instantes en los que se obtienen las observaciones de la muestra están igualmente espaciados. En ese caso, el número de operaciones necesarias para calcular todos los coeficientes c_k y los p datos suavizados de cada observación $x(t)$ evaluada en los valores t_j es de $O(p \log p)$ en lugar de $O(p^2)$ que se requeriría por evaluación directa de la fórmula de la transformada discreta de Fourier. Esto supone una disminución importante en el número de operaciones pues si consideramos $p = 1024$ esta simplificación reduce el número de operaciones de algo más de un millón a tan sólo 10.240.

Como:

- $D[\sin r\omega t] = r\omega \cos r\omega t$,
- $D[\cos r\omega t] = -r\omega \sin r\omega t$,

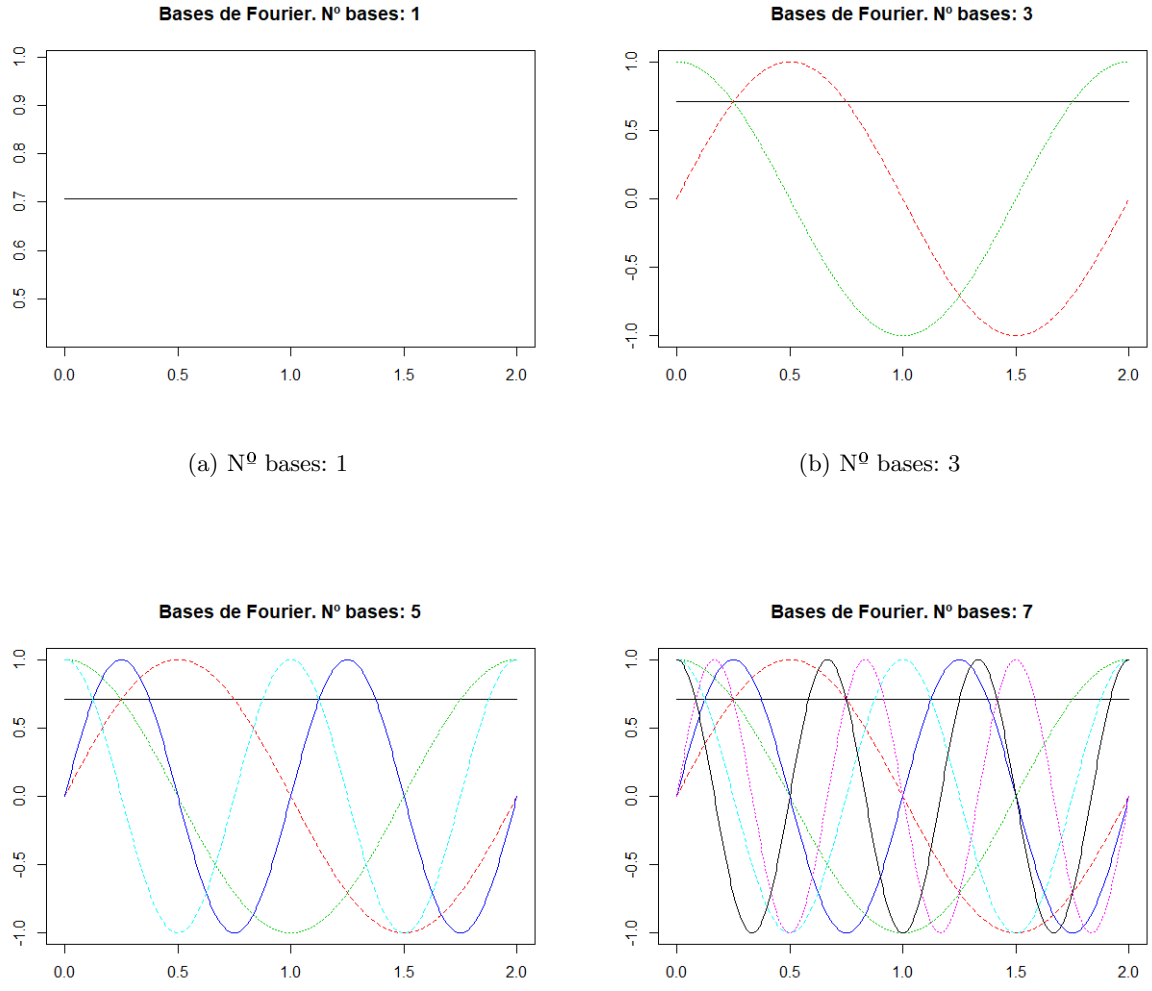
donde el símbolo D representa la derivada de la función introducida entre corchetes, los coeficientes de la derivada de la base de Fourier vienen dados por

$$(0, \omega c_1, -\omega c_2, 2\omega c_3, -2\omega c_4, \dots).$$

La serie de Fourier es una de las bases de funciones más utilizadas por los estadísticos y por los ingenieros. Resulta verdaderamente útil para trabajar con funciones estables, es decir, funciones donde no hay características locales fuertes y donde la curvatura tiende a ser idéntica en todas las partes de la curva. Sin embargo, no son adecuadas para datos que presentan discontinuidades bien en la función o en sus derivadas de orden bajo.

La serie de Fourier es computacionalmente eficiente y se puede aplicar prácticamente a cualquier campo. No obstante, los resultados obtenidos suelen ser más adecuados utilizando otras bases de funciones que se explican en secciones posteriores.

En la Figura 3.2 se muestra la base de Fourier para $1, 3, 5$ y 7 funciones base.



(a) Nº bases: 1

(b) Nº bases: 3

(c) Nº bases: 5

(d) Nº bases: 7

Figura 3.2: Bases de Fourier para $K = 1, 3, 5$ y 7.

3.6.2. Base de B-Splines

Las funciones spline son las más utilizadas para aproximar funciones no periódicas. Han conseguido sustituir prácticamente a las bases de polinomios porque, además de proporcionar la rapidez computacional que se obtiene con las bases de polinomios, ofrecen una mayor flexibilidad que se alcanza normalmente con tan sólo un número de funciones base pequeño.

Además, se han desarrollado sistemas de bases para las funciones spline cuyo coste computacional es $O(p)$, lo que resulta esencial porque muchas aplicaciones trabajan con grandes cantidades de observaciones.

En esta sección se explica en primer lugar la estructura de una función spline para posteriormente poder explicar el sistema de bases que se utiliza habitualmente para construir splines: el sistema B-spline.

El primer paso para definir un spline consiste en dividir el intervalo sobre el que se quieren aproximar las funciones en L subintervalos, separados por los valores τ_l donde $l = 1, \ldots, L-1$. El conjunto formado por estos puntos de corte recibe el nombre de vector nudo y, a cada elemento del vector nudo, se le denomina nodo o nudo.

En cada intervalo, un spline es un polinomio de un grado especifico m. El orden de un polinomio es el número de constantes necesarias para definirlo y es uno más que su grado.

Los polinomios adyacentes se unen suavemente en los puntos de unión proporcionando continuidad de tipo C^{m-2} en cada uno de los nodos. Se puede reducir la continuidad en un nodo repitiendo el valor de ese nodo. Si se tiene un spline de orden m con un nodo de multiplicidad s (el valor del nodo se repite s veces), la continuidad en el nodo se reduce de C^{m-2} a C^{m-s-1}.

En la Figura 3.3 se observa cómo se aproximan las funciones spline a la función seno en el intervalo $[0, 2\pi]$ (en las imágenes situadas a la izquierda) y cómo se aproximan las funciones spline a la derivada de la función seno, es decir, a la función coseno (en las imágenes situadas a la derecha). Los órdenes de los splines por los que se aproximan ambas funciones son de arriba a abajo 2, 3 y 4. Observamos que se han escogido tres puntos de corte dividiendo, de este modo, el intervalo en cuatro subintervalos. Si se incluyen también 0 y 2π como puntos de corte, entonces se numeran como $\tau_0, \ldots, \tau_L$ donde $L = 4$.

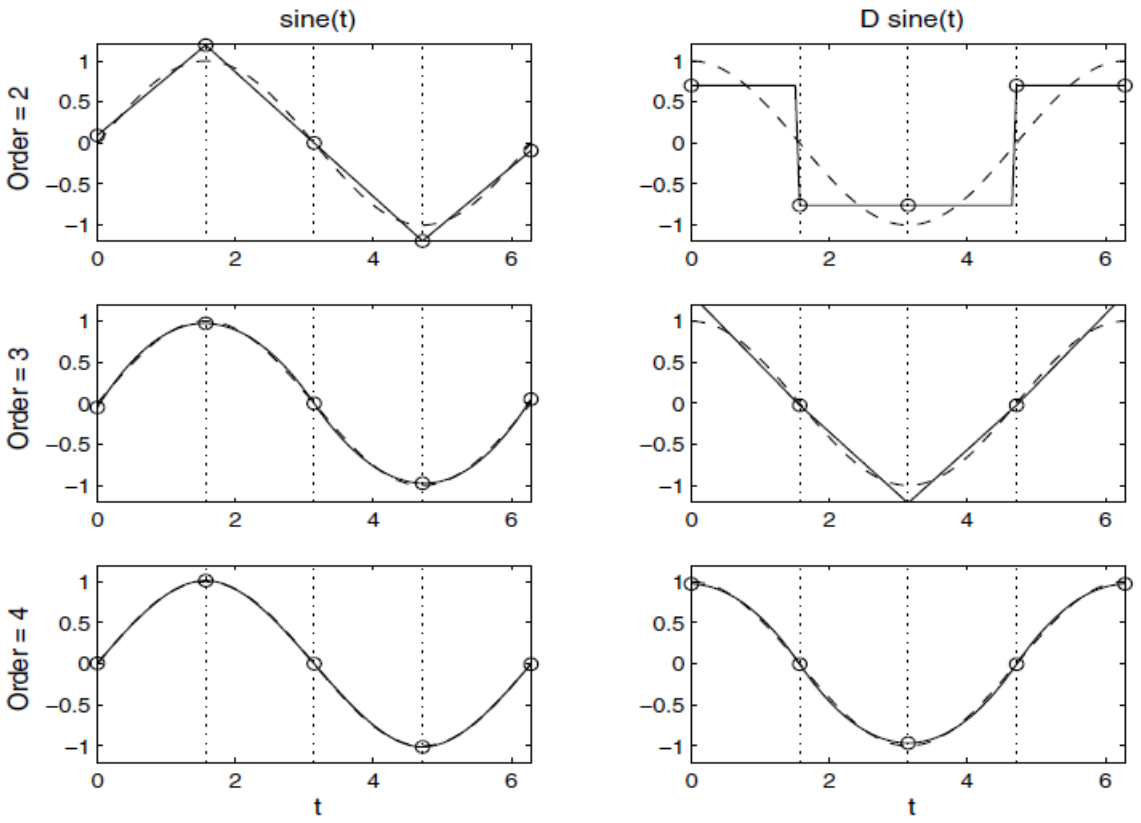


Figura 3.3: Aproximación mediante funciones splines a la función seno (imágenes de la izquierda) y a su derivada, la función coseno (imágenes de la derecha) con ordenes 2, 3 y 4 respectivamente, sobre el intervalo $[0, 2\pi]$.

Las líneas punteadas representan las funciones que se deben aproximar y las líneas negras las aproximaciones realizadas mediante las funciones spline.

Para calcular el número de grados de libertad en el ajuste, si observamos la aproximación de la función seno por un spline de orden dos, como tan sólo hay dos grados de libertad en una recta y se tienen cuatro rectas, el número de coeficientes para definirlas es 2×4, pero debemos restar un grado de libertad por cada uno de los tres puntos de unión, ya que los valores de la función coinciden en los puntos de unión. Por tanto, el número de grados de libertad es: $2 \times 4 - 3 \times 1 = 5$.

En el caso de la aproximación de orden tres, los polinomios son cuadráticos y por tanto dan lugar a 3×4 coeficientes pero, en este caso, la función y la primera derivada se unen suavemente luego debemos restar 3×2 grados de libertad. En total se tienen $3 \times 4 - 3 \times 2 = 6$ grados de libertad.

Finalmente, para la aproximación de orden cuatro, los polinomios son cúbicos, donde los valores de la la función y de las dos primeras derivadas coinciden, dando así lugar a $4 \times 4 - 3 \times 3 =$

7 grados de libertad.

Se verifica que el número de grados de libertad del ajuste es igual a la suma del orden de los polinomios más el número de nodos interiores.

Aumentando el orden de los polinomios se obtiene una mejor aproximación. Sin embargo, la mejor manera de ganar flexibilidad en un spline es aumentando el número de nodos. Aunque, en general, se suelen utilizar nodos equiespaciados, es más adecuado situar más nodos en las zonas donde la función tenga mayor variación. Además, siempre debe haber un dato dentro de cada subintervalo. No es lógico que entre dos nodos no haya ningún valor.

Por lo tanto, en general, una función spline está determinada por dos elementos:

- El orden de los segmentos polinomiales.
- El vector nudo dado por la sucesión $\tau_0, \ldots, \tau_l$ con $l = 1, \ldots, L-1$.

El número de parámetros para definir una función spline, en el caso en que cada nodo tenga tan sólo multiplicidad uno, viene dado por el orden más el número de nodos interiores, $m + L - 1$.

Bases B-spline para las funciones spline

Hasta ahora se ha definido qué es una función spline pero no se han dado las claves necesarias para construirla. Para ello, es necesario un sistema de funciones base $\phi_k(t)$ que deben verificar las siguientes propiedades:

- Cada función base $\phi_k(t)$ debe ser una función spline definida con un orden m y un vector nudo τ.
- Cualquier combinación de las funciones base debe ser una función spline, ya que la suma y el producto de funciones spline es también una función spline.
- Cada función spline definida con un orden m y un vector nudo τ se debe poder expresar como una combinación lineal de esas funciones base.

La manera más conocida de construir funciones spline es utilizando el sistema de bases B-splines que fue desarrollado por de Boor (2001) [11]. Una función spline $S(t)$ con nodos discretos interiores se define de la siguiente manera:

$$S(t) = \sum_{k=1}^{m+L-1} c_k B_k(t, \tau),$$

donde $B_k(t, \tau)$ denota el valor del $k-$ésimo B-spline en el punto t definido por el vector nudo τ.

En la Figura 3.4 se representan B-splines de grado 1, 2, 3 y 4 para un vector nudo con nodos equiespaciados.

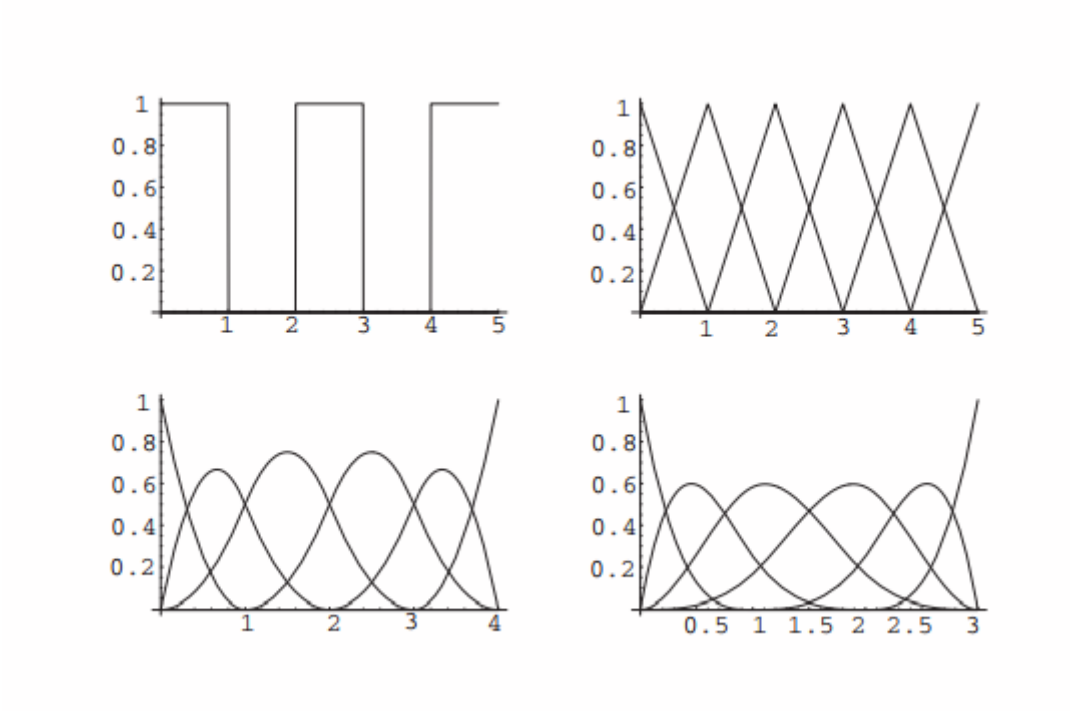


Figura 3.4: B-splines de grado 1, 2, 3 y 4 para un vector nudo uniforme (con nodos equiespaciados).

Una característica sorprendente de las bases spline es que el aumento del número de B-splines no siempre produce un mejor ajuste. Es decir, a veces, dependiendo de donde se coloquen los nodos es posible que un sistema B-spline de menor dimensión produzca mejores resultados que un sistema de dimensiones superiores. Esto es debido a que el espacio de funciones definido por K B-splines no siempre está contenido en el definido por $K+1$ B-splines. Sin embargo, si K se incrementa introduciendo un nuevo nodo a los anteriores o aumentando el orden y dejando el vector nudo intacto, entonces el $K-$espacio está contenido en el $K+1-$espacio.

3.6.3. Bases de potencias y bases exponenciales

Las familias más conocidas de funciones base son las utilizadas para construir series de potencias:

$$1, t, t^2, t^3, \ldots, t^k, \ldots$$

Los sistemas de base exponencial consisten en una serie de funciones exponenciales,

$$1, e^t, e^{2t}, \ldots, e^{kt}, \ldots$$

Este tipo de funciones es usual encontrarlo en las soluciones de las ecuaciones diferenciales lineales con coeficientes constantes.

3.6.4. Bases polinomiales

Una base polinomial es de la forma $\phi_k(t) = (t-\omega)^k, k = 0, \ldots, K$ donde ω es un parámetro que normalmente se elige en el centro del intervalo de aproximación. Este parámetro se puede tomar cuidadosamente para evitar el error de redondeo en los cálculos, ya que los valores presentan una mayor correlación cuando el grado aumenta.

Uno de los inconvenientes que presenta esta base es que no puede describir algunas características locales sin utilizar un valor de K grande. Además, suele aproximar bien los datos centrales pero no se ajusta correctamente en las colas.

3.6.5. Ondículas

Las ondículas se utilizan como funciones básicas para representar otras funciones tal y como se hace con las funciones seno y coseno en las bases de Fourier. A diferencia de con las bases de Fourier, en el análisis con ondículas no se asume que los datos sean periódicos, por lo tanto, es posible estudiar datos no periódicos utilizando muchas menos funciones que las que se necesitarían si se utilizaran funciones seno y coseno para alcanzar una aproximación adecuada de la forma funcional.

El análisis con ondículas permite definir una función prototipo (también conocida con el nombre de ondícula madre) que no siempre es la misma, es decir, las funciones base no siempre son iguales a diferencia de lo que ocurre con las bases de Fourier, donde las funciones base son siempre el seno y el coseno.

Se considera una función ondícula madre, ψ, considerándose las dilataciones y las traslaciones de la forma

$$\psi_{jk} = 2^{j/2}\psi(2^j t - k),$$

con j y k enteros. Esta función se construye para asegurar que la base es ortogonal, en el sentido de que la integral del producto de dos funciones base cualesquiera distintas es cero.

Las ondículas son las bases de funciones más recientes. Entre sus ventajas podemos destacar la rapidez computacional y que se adaptan bien con discontinuidades o cambios rápidos.

3.7. Métodos de suavizado

Como ya se ha comentado en secciones anteriores, cada dato funcional se recoge como p pares de observaciones de la forma (t_j, y_j) donde y_j es el valor observado en el tiempo t_j. Para convertir los datos recogidos en forma discreta a una función, se utiliza una técnica que se conoce como suavizado y que consiste en ajustar los datos a una base de funciones, permitiendo también eliminar el ruido registrado al obtener las observaciones.

Existen varios métodos de suavizado entre los que destacan el suavizado por mínimos cuadrados, el suavizado mediante kernels o el suavizado por regularización.

3.7.1. Suavizar datos funcionales por mínimos cuadrados

El objetivo es ajustar una curva $x(t)$ a las observaciones discretas $y_j, j = 1, \ldots, p$ (donde $y_j = x(t_j) + \epsilon_j$ siendo $x(t_j)$ el valor de la función $x(t)$ en el instante t_j y siendo ϵ_j el error observacional correspondiente a la observación y_j) usando una combinación lineal de funciones base para $x(t)$ de la forma

$$x(t) = \sum_{k=1}^{K} c_k \phi_k(t) = \boldsymbol{c^t \phi}.$$

Los vectores $\boldsymbol{c}$ y $\boldsymbol{\phi}$ son de longitud K y contienen los coeficientes c_k y las funciones base ϕ_k respectivamente.

Definimos Φ como la matriz que contiene los valores de las K funciones base para los p puntos de la muestra, es decir, es una matriz $p \times K$ que contiene los valores $\phi_k(t_j)$.

Ajuste por mínimos cuadrados no ponderados

Los coeficientes c_k se determinan por el criterio de mínimos cuadrados

$$\text{SMSSE}(\mathbf{y}|\boldsymbol{c}) = \sum_{j=1}^{p} [y_j - \sum_{k=1}^{K} c_k \phi_k(t_j)]^2.$$

Expresado en forma matricial se obtiene

$$\text{SMSSE}(\mathbf{y}|\boldsymbol{c}) = (\mathbf{y} - \Phi \boldsymbol{c})^t (\mathbf{y} - \Phi \boldsymbol{c}).$$

Nota 1 *Si $\boldsymbol{A}^t$ y $\boldsymbol{B}^t$ son las matrices transpuestas de $\boldsymbol{A}$ y $\boldsymbol{B}$, respectivamente, entonces*

- $(\boldsymbol{A} + \boldsymbol{B})^t = \boldsymbol{A}^t + \boldsymbol{B}^t$,
- $(\boldsymbol{A} \times \boldsymbol{B})^t = \boldsymbol{B}^t \times \boldsymbol{A}^t$.

Aplicando la Nota 1, la expresión anterior se puede escribir como

$$\text{SMSSE}(\mathbf{y}|\boldsymbol{c}) = (\mathbf{y}^t - (\Phi\boldsymbol{c})^t)(\mathbf{y} - \Phi\boldsymbol{c}) = (\mathbf{y}^t - \boldsymbol{c}^t\Phi^t)(\mathbf{y} - \Phi\boldsymbol{c}) =$$
$$\mathbf{y}^t\mathbf{y} - \mathbf{y}^t\Phi\boldsymbol{c} - \boldsymbol{c}^t\Phi^t\mathbf{y} + \boldsymbol{c}^t\Phi^t\Phi\boldsymbol{c}.$$

Nota 2 *Dados los vectores* $\boldsymbol{a}$ *y* $\boldsymbol{x}$*; y la matriz cuadrada simétrica* $\boldsymbol{A}$*, se verifica:*

- *Si* $f = \boldsymbol{a}^t\boldsymbol{x}$*, entonces* $\dfrac{\partial f}{\partial \boldsymbol{x}} = \boldsymbol{a}$ *y si* $f = \boldsymbol{x}^t\boldsymbol{a}$*, entonces* $\dfrac{\partial f}{\partial \boldsymbol{x}^t} = \boldsymbol{a}$,
- *Si* $f = \boldsymbol{x}^t\boldsymbol{A}\boldsymbol{x}$*, entonces* $\dfrac{\partial f}{\partial \boldsymbol{x}} = 2\boldsymbol{A}\boldsymbol{x}$.

Tomando la derivada de $\text{SMSSE}(\mathbf{y}|\boldsymbol{c})$ con respecto de $\boldsymbol{c}$ aplicando la Nota 2 se tiene la ecuación

$$2\Phi^t\Phi\boldsymbol{c} - 2\Phi^t\mathbf{y} = 0,$$

y despejando $\boldsymbol{c}$ se obtiene el estimador $\widehat{\boldsymbol{c}}$ que minimiza $\text{SMSSE}(\mathbf{y}|\boldsymbol{c})$,

$$\cancel{2}\Phi^t\Phi\boldsymbol{c} = \cancel{2}\Phi^t\mathbf{y}$$
$$\widehat{\boldsymbol{c}} = (\Phi^t\Phi)^{-1}\Phi^t\mathbf{y}.$$

La curva ajustada es

$$\widehat{\mathbf{y}} = \Phi\widehat{\boldsymbol{c}} = \Phi(\Phi^t\Phi)^{-1}\Phi^t\mathbf{y}.$$

La aproximación por mínimos cuadrados es adecuada cuando los residuos ϵ_j son independientes e idénticamente distribuidos con media cero y varianza constante.

Ajuste por mínimos cuadrados ponderados

Cuando las varianzas de los errores no son constantes o los errores no están idénticamente distribuidos, se debe aportar un peso diferente a los distintos residuos. Para ello, se extiende el criterio de mínimos cuadrados de la forma

$$\text{SMSSE}(\mathbf{y}|\boldsymbol{c}) = (\mathbf{y} - \Phi\boldsymbol{c})^t\mathbf{W}(\mathbf{y} - \Phi\boldsymbol{c}),$$

donde $\mathbf{W}$ es una matriz simétrica y definida positiva (para todos los vectores $\mathbf{v} \in \mathbb{C}^p, \mathbf{v}^t\mathbf{W}\mathbf{v} > 0$). Si la matriz de varianzas y covarianzas $\sum_e$ para los residuos ϵ_j es conocida entonces

$$\mathbf{W} = \sum\nolimits_e{}^{-1},$$

pues toda matriz definida positiva es invertible (su determinante es positivo), y su inversa es definida positiva.

En aplicaciones donde no es factible estimar $\sum_e$, se asume que las covarianzas entre los errores son cero y en ese caso $\mathbf{W}$ es diagonal con la varianza del error asociado a los y_j's en la diagonal. De este modo, derivando la expresión $\text{SMSSE}(\mathbf{y}|\boldsymbol{c})$ y despejando $\boldsymbol{c}$, el estimador de mínimos cuadrados ponderados para los coeficientes c_k es

$$\widehat{\boldsymbol{c}} = (\Phi^t\mathbf{W}\Phi)^{-1}\Phi^t\mathbf{W}\mathbf{y}.$$

3.7.2. Elección del número de funciones base

Para elegir el número de funciones base K, se debe tener en cuenta que cuanto mayor sea K mejor será el ajuste pero, en ese caso, se corre el riesgo de ajustar el ruido que puede distorsionar los resultados fácilmente. Por el otro lado, si se escoge K demasiado pequeño es posible que la función que se pretende estimar sea demasiado suave y que se pierdan características importantes.

Para elegir el número de funciones base, existen varios métodos basados en los datos, aunque ninguno es válido en todos los casos. A día de hoy, determinar el número de bases a elegir, todavía sigue siendo un campo de investigación activo. En general se suele probar distinto número de bases y, en base a las gráficas (de la función y de sus derivadas), decidir.

Cuando se suaviza una curva, la cantidad total de información sólo permite estimar una varianza constante, o a lo sumo, una función varianza $\sigma^2(t)$ con leves variaciones. Si se asume un modelo estándar con error, el estimador más adecuado para calcular la varianza de los residuos de cada función de la muestra es

$$s^2 = \frac{1}{p-K}\sum_{j=1}^{p}(y_j - \widehat{y}_j)^2.$$

Una estrategia razonable para elegir K es calcular la media de la varianza de los residuos de todas las funciones de la muestra mediante la ecuación

$$\overline{s^2} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{p_i-K}\sum_{j=1}^{p_i}(y_{ij} - \widehat{y}_{ij})^2\right),$$

e ir añadiendo bases mientras $\overline{s^2}$ decrezca rápidamente.

Existen otras estrategias pero ninguna de ellas es idónea pues el mayor problema es que son computacionalmente muy costosas. Dos de estas estrategias son el criterio de la incorrelación de los errores y aplicar validación cruzada.

Criterio de la incorrelación de los errores

El criterio de la incorrelación de los errores se basa en los siguientes principios:

- Si se usan pocas funciones base entonces la función es suave, existen bandas de datos a cada lado de la curva ajustada y por tanto, la correlación de los errores es muy alta.
- Aumentando el número de bases, el ajuste se acerca cada vez más a la interpolación e irá disminuyendo la autocorrelación.

El algoritmo que se utiliza es un algoritmo iterativo que ajusta y calcula la autocorrelación, haciendo la media, cada vez con un número mayor de bases, hasta que la correlación es cero (pasa de positiva a negativa).

Validación cruzada

La idea básica de estimar el número de bases mediante validación cruzada consiste en separar la muestra en dos partes. Una parte recibe el nombre de muestra de entrenamiento y se utiliza para ajustar el modelo a los datos, y la otra parte recibe el nombre de muestra de validación y se utiliza para testear el modelo con esos datos. De esta forma, se puede comprobar cómo de bien se ajusta el modelo a los datos que no se utilizaron para estimar la función. Además del coste computacional, otro de los problemas de este método es que tiende a no suavizar demasiado ajustando ruido que en general se prefiere ignorar.

3.7.3. Mínimos cuadrados localizados. Suavizado mediante kernels

Recordemos que el objetivo consiste en ajustar una curva $x(t)$ a las observaciones discretas $y_j, j = 1, \ldots, p$ (donde $y_j = x(t_j) + \epsilon_j$).

Para un método de suavizado, el valor de la función estimada en un punto t puede estar influenciada por las observaciones próximas a t. Este rasgo es una propiedad implícita de los estimadores que se han considerado hasta ahora. En este apartado, se consideran estimadores donde la dependencia local se hace más explícita mediante funciones de peso local.

De acuerdo con el dominio de suavizado lineal, el estimador de la función $x(t)$ es de la forma

$$x(t) = \sum_{j=1}^{p} w_j y_j,$$

de modo que los pesos w_j serán relativamente grandes para valores muestrales t_j próximos al valor fijado t.

Ahora buscamos métodos de suavizado que hagan explícito este principio de ponderación localizada. Las ponderaciones w_j se construyen mediante un cambio de origen y escala de una función núcleo con valores $\text{Kern}(u)$.

Un *kernel* es una función de variable real $\text{Kern}: \mathbb{R} \to \mathbb{R}$ con las siguientes propiedades:

- $\text{Kern}(u) \in [0, \infty)$ si $u \in [-1, 1]$,
- $\text{Kern}(u) \approx 0$ si $u \notin [-1, 1]$,
- $\text{Kern}(u) = \text{Kern}(-u)$,
- $\int_{-1}^{1} \text{Kern}(u)\, du = 1$,
- $\int_{-1}^{1} u\, \text{Kern}(u)\, du = 0$,
- $\int_{-1}^{1} u^2\, \text{Kern}(u)\, du \in \mathbb{R}^+$.

Es decir, tiene la mayoría de su masa concentrada próxima a cero, y decaen o desaparecen por completo para $|u| \geq 1$.

Algunos ejemplos de kernel son:

- **Uniforme:**
$$\text{Kern}(u) = \begin{cases} 0.5, \text{ si } |u| \leq 1, \\ 0, \text{ en otro caso.} \end{cases}$$
- **Cuadrático:**
$$\text{Kern}(u) = \begin{cases} 0.75(1-\mu^2), \text{ si } |u| \leq 1, \\ 0, \text{ en otro caso.} \end{cases}$$
- **Gaussiano:**
$$\text{Kern}(u) = (2\pi)^{-1/2} e^{(-u^2/2)}.$$

Si definimos los pesos como
$$w_j(t) = \text{Kern}\left(\frac{t_j - t}{h}\right),$$
entonces los valores grandes de w_j se concentran ahora en los t_j próximos a t. El grado de concentración es controlado por el tamaño de h. El parámetro de concentración h se denomina generalmente *ancho de banda*. Valores pequeños de h implican que sólo las observaciones cercanas a t (punto en el que se estima) reciben peso.

El caso más simple y clásico de un estimador que hace uso de pesos locales es el estimador kernel. La estimación en un punto dado es una combinación lineal de las observaciones locales,
$$\widehat{x}(t) = \sum_{j=1}^{p} S_j(t) y_j,$$
para algunas funciones de peso S_j definidas adecuadamente.

El estimador kernel más popular es el estimador Nadaraya-Watson. Se construye usando los pesos
$$S_j(t) = \frac{\text{Kern}[(t_j - t)/h]}{\sum_r \text{Kern}[(t_r - t)/h]}.$$
Aunque los valores de peso para el método de Nadaraya-Watson son normalizados, esto no es esencial.

Los pesos desarrollados por Gasser y Müller se construyen de la siguiente manera
$$S_j(t) = \frac{1}{h} \int_{\bar{t}_{j-1}}^{\bar{t}_j} \text{Kern}\left(\frac{u - t}{h}\right) \, du, \quad \bar{t}_j = \frac{t_{j+1} + t_j}{2}.$$
$1 < j < p$, $\bar{t}_0 = t_1$ y $\bar{t}_n = t_n$.

La principal ventaja de los kernels es la computación rápida. Su principal debilidad es que no aproxima bien cerca de los extremos, sobre todo si h es grande en relación a la tasa de muestreo. No es conveniente estimar derivadas derivando las funciones kernel. Se pueden diseñar kernels específicos para estimar derivadas.

Las ideas de los estimadores núcleo y de los estimadores de funciones base pueden, en cierto sentido, combinarse para obtener estimadores de funciones base localizadas. La idea básica es extender el criterio de mínimos cuadrados para dar una medida del error local como

$$SMEE_t(\mathbf{y}|\boldsymbol{c}) = \sum_{j=1}^{p} w_j(t)[y_j - \sum_{k=1}^{K} c_k \phi_k(t_j)]^2,$$

donde los pesos w_j se construyen a partir de las funciones kernel utilizando

$$w_j(t) = \text{Kern}\left(\frac{t_j - t}{h}\right).$$

En términos matriciales,

$$SMEE_t(\mathbf{y}|\boldsymbol{c}) = (\mathbf{y} - \Phi\boldsymbol{c})^t \mathbf{W}(t)(\mathbf{y} - \Phi\boldsymbol{c}),$$

donde $\mathbf{W}(t)$ es una matriz diagonal que contiene los pesos $w_j(t)$ en la diagonal. Los coeficientes $\boldsymbol{c}(t)$ que minimizan $SMEE_t(\mathbf{y}|\boldsymbol{c})$ son

$$\widehat{\boldsymbol{c}}(t) = (\Phi\mathbf{W}(t)\Phi)^{-1}\Phi^t\mathbf{W}(t)\mathbf{y},$$

y sustituyendo en

$$\widehat{x}(t) = \sum_{k=1}^{K} \widehat{c}_k \phi_k(t),$$

se obtiene un estimador de la forma

$$\widehat{x}(t) = \sum_{j=1}^{p} S_j(t_j) y_j$$

siendo los valores S_j los elementos del vector

$$S(t) = \mathbf{W}(t)\Phi[\Phi^t\mathbf{W}(t)\Phi]^{-1}\phi(t),$$

donde $\phi(t)$ es el vector con elementos $\phi_k(t)$.

Aproxima adecuadamente características locales con un número pequeño K de funciones base. Computacionalmente depende del número de t_j's con $w_j(t)$ distinto de cero y del valor de K que, en general, suelen ser pequeños. Sin embargo, el precio a pagar por esta flexibilidad es que se debe realizar la expansión para cada punto de evaluación t.

3.7.4. Suavizar datos funcionales por regularización

Uno de los inconvenientes que tiene el suavizado de datos funcionales aplicando el ajuste por mínimos cuadrados es la dificultad de controlar el grado de suavidad de la curva ajustada. Por ello, una de las alternativas al suavizado por mínimos cuadrados es el suavizado por regularización, donde el suavizado spline es uno de sus casos particulares más utilizados.

La regularización es uno de los métodos más adecuados para resolver los problemas que ocurren en el análisis de datos funcionales. Produce mejores resultados que mediante mínimos cuadrados, especialmente en la estimación de derivadas.

En el método de regularización se exige que la curva sea suave y se impone en la expresión que se debe minimizar.

Para estimar la curva $x(t)$ que genera las observaciones $y_j, j = 1, \ldots, p$ donde $y_j = x(t_j) + \epsilon_j$ tiene en cuenta dos objetivos:

- Garantizar que la curva estimada $x(t)$ es una aproximación adecuada a los datos. Una forma de hacerlo es minimizando la suma de cuadrados de los residuos:
$$\sum_{j=1}^{p} [y_j - x(t_j)]^2.$$
- Asegurar que la curva sea suave.

Por tanto, la regularización busca un equilibrio entre el ajuste a los datos y la suavidad. En este método se pueden utilizar más funciones base que valores observados haya en la muestra y mantener aún así la suavidad.

La medida más utilizada para cuantificar la irregularidad de una función es el cuadrado de la segunda derivada $[D^2x(t)]^2$ de una función, conocido como curvatura de x en t, ya que una línea recta no tiene curvatura y su segunda derivada es nula. Por ello, una forma natural de medir la irregularidad de una función es en términos de la integral del cuadrado de la segunda derivada
$$PEN_2(x) = \int [D^2x(s)]^2 ds.$$
Se puede esperar que las funciones con elevada variación produzcan valores altos de $PEN_2(x)$ porque sus segundas derivadas son grandes sobre al menos parte del rango de interés.

Se define la penalización de cuadrados del error como:
$$PENSSE_\lambda(x|\mathbf{y}) = \sum_{j=1}^{p} [y_j - x(t_j)]^2 + \lambda PEN_2(x).$$
La función estimada se obtiene encontrando la función $x(t)$ que minimiza $PENSSE_\lambda(x)$ sobre el espacio de funciones para el que $PEN_2(x)$ está definida. El parámetro λ es un parámetro de

suavizado que mide la tasa de cambio entre el ajuste a los datos y la variabilidad de la función, cuantificada por $PEN_2(x)$.

A medida que λ aumenta, cada vez se penaliza más la irregularidad a través del término $PEN_2(x)$. El criterio $PENSSE_\lambda(x|\mathbf{y})$ da más énfasis a la suavidad y menos a la adaptación de los datos. Por esta razón, cuando $\lambda \longrightarrow \infty$, la curva ajustada debe aproximarse a la regresión lineal de los datos observados, donde $PEN_2(x) = 0$. Por otro lado, para valores de λ pequeños, la curva tiende a ser cada vez más variable, ya que cada vez hay menos penalizaciones en la irregularidad. Por esta razón cuando $\lambda \longrightarrow 0$, la curva ajustada debe aproximarse a una interpolación de los datos, satisfaciendo $x(t_j) = y_j$ para todo j.

Si no se hacen suposiciones sobre la función a estimar, excepto que tiene una segunda derivada y se asume que los puntos de muestreo $t_j, j = 1, \ldots, p$ son distintos, la función que minimiza $PENSSE_\lambda(x|\mathbf{y})$, por un teorema encontrado en de Boor (2002) [11] y otros textos más avanzados sobre suavizado, es una función spline cúbica con nodos en los puntos t_j. Situar los nodos en los datos elimina uno de los problemas en el uso de splines. Además, se adapta de manera natural al espaciamiento desigual de los puntos de muestreo y, por lo tanto, de las regiones donde la densidad de datos es alta, y al mismo tiempo, especialmente suave en las regiones donde hay pocas observaciones. Cuando el número de nodos es muy elevado, por razones computacionales se aconseja disminuir su número.

Suavizado spline

El objetivo del suavizado consiste en ajustar las observaciones $y_j, j = 1, \ldots, p$ usando el modelo $y_j = x(t_j) + \epsilon_j$. Para ello, se hace uso de una combinación lineal de funciones base para $x(t)$ de la forma

$$x(t) = \sum_{k=1}^{K} c_k \phi_k(t) = \boldsymbol{c^t \phi}.$$

Los vectores $\boldsymbol{c}$ y $\boldsymbol{\phi}$ son de longitud K y contienen los coeficientes c_k y las funciones base ϕ_k respectivamente.

Recordemos que definíamos Φ como la matriz que contiene los valores de las K funciones base para las p observaciones y, que en el caso de no penalizar la irregularidad, es decir, aplicando un ajuste de mínimos cuadrados ponderados, la estimación del vector $\boldsymbol{c}$ es

$$\widehat{\boldsymbol{c}} = (\Phi^t \mathbf{W} \Phi)^{-1} \Phi^t \mathbf{W} \mathbf{y},$$

donde $\mathbf{W}$ es una matriz de pesos definida positiva e $\mathbf{y}$ es el vector de datos que se quiere suavizar.

Como $\widehat{\mathbf{y}} = \Phi \widehat{\boldsymbol{c}}$, la curva ajustada en ese caso es

$$\widehat{\mathbf{y}} = \Phi(\Phi^t \mathbf{W} \Phi)^{-1} \Phi^t \mathbf{W} \mathbf{y},$$

que se puede escribir como

$$\widehat{\mathbf{y}} = \mathbf{S}_\phi \mathbf{y},$$

donde $\mathbf{S}_\phi$ es el *operador de proyección*

$$\mathbf{S}_\phi = \Phi(\Phi^t \mathbf{W} \Phi)^{-1} \Phi^t \mathbf{W},$$

correspondiente al sistema de bases ϕ.

Renombrando la penalización de la irregularidad $PEN_m(x)$ en términos matriciales, se sigue

$$\begin{aligned} PEN_m(x) &= \int [D^m x(s)]^2 ds \\ &= \int [D^m \boldsymbol{c}^t \boldsymbol{\phi}(s)]^2 \; ds \\ &= \int [D^m \boldsymbol{c}^t \boldsymbol{\phi}(s)][D^m \boldsymbol{c}^t \boldsymbol{\phi}(s)] \; ds \\ &= \int [D^m \boldsymbol{c}^t \boldsymbol{\phi}(s)][D^m \boldsymbol{\phi}^t(s) \boldsymbol{c}] \; ds \\ &= \boldsymbol{c}^t [\int D^m \boldsymbol{\phi}(s) D^m \boldsymbol{\phi}^t(s) \; ds] \; \boldsymbol{c} \\ &= \boldsymbol{c}^t \mathbf{R} \boldsymbol{c}, \end{aligned}$$

donde

$$\mathbf{R} = \int D^m \boldsymbol{\phi}(s) D^m \boldsymbol{\phi}^t(s) \; ds.$$

Por lo tanto, la penalización de cuadrados del error viene dada por

$$\begin{aligned} PENSSE_m(\mathbf{y}|\boldsymbol{c}) &= (\mathbf{y} - \Phi \boldsymbol{c})^t \mathbf{W} (\mathbf{y} - \Phi \boldsymbol{c}) + \lambda \boldsymbol{c}^t \mathbf{R} \boldsymbol{c} \\ &= (\mathbf{y}^t - (\Phi \boldsymbol{c})^t) \mathbf{W} (\mathbf{y} - \Phi \boldsymbol{c}) + \lambda \boldsymbol{c}^t \mathbf{R} \boldsymbol{c} \\ &= (\mathbf{y}^t - \boldsymbol{c}^t \Phi^t) \mathbf{W} (\mathbf{y} - \Phi \boldsymbol{c}) + \lambda \boldsymbol{c}^t \mathbf{R} \boldsymbol{c} \\ &= \mathbf{y}^t \mathbf{W} \mathbf{y} - \mathbf{y}^t \mathbf{W} \Phi \boldsymbol{c} - \boldsymbol{c}^t \Phi^t \mathbf{W} \mathbf{y} + \boldsymbol{c}^t \Phi^t \mathbf{W} \Phi \boldsymbol{c} + \lambda \boldsymbol{c}^t \mathbf{R} \boldsymbol{c}. \end{aligned}$$

Tomando ahora la derivada con respecto de $\boldsymbol{c}$, se obtiene

$$-2\Phi^t \mathbf{W} \mathbf{y} + 2\Phi^t \mathbf{W} \Phi \boldsymbol{c} + 2\lambda \mathbf{R} \boldsymbol{c} = 0.$$

Despejando $\boldsymbol{c}$ se tiene el estimador $\widehat{\boldsymbol{c}}$ que minimiza $PENSSE_m(\mathbf{y}|\boldsymbol{c})$,

$$\begin{aligned} 2\Phi^t \mathbf{W} \Phi \boldsymbol{c} + 2\lambda \mathbf{R} \boldsymbol{c} &= 2\Phi^t \mathbf{W} \mathbf{y} \\ \cancel{2}(\Phi^t \mathbf{W} \Phi + \lambda \mathbf{R}) \boldsymbol{c} &= \cancel{2} \Phi^t \mathbf{W} \mathbf{y} \\ \widehat{\boldsymbol{c}} &= (\Phi^t \mathbf{W} \Phi + \lambda \mathbf{R})^{-1} \Phi^t \mathbf{W} \mathbf{y}. \end{aligned}$$

Los valores suavizados quedan expresados mediante

$$\widehat{\mathbf{y}} = \Phi \widehat{\boldsymbol{c}} = \Phi (\Phi^t \mathbf{W} \Phi + \lambda \mathbf{R})^{-1} \Phi^t \mathbf{W} \mathbf{y},$$

donde $\widehat{\mathbf{y}}$ se puede expresar como

$$\widehat{\mathbf{y}} = \mathbf{S}_{\phi,\lambda} \mathbf{y},$$

con

$$\mathbf{S}_{\phi,\lambda} = \Phi(\Phi^t\mathbf{W}\Phi + \lambda\mathbf{R})^{-1}\Phi^t\mathbf{W}.$$

Si comparamos la expresión de $\mathbf{S}_{\phi,\lambda}$ con la del operador proyección $\mathbf{S}_\phi$, se observa que la única diferencia es el término $\lambda\mathbf{R}$, luego los dos operadores son iguales cuando el parámetro de suavizado λ toma el valor cero. Al operador $\mathbf{S}_{\phi,\lambda}$ se le denota por *operador sub-proyección*, porque a diferencia del operador proyección, el operador sub-proyección no satisface la propiedad de idempotencia

$$\mathbf{S}_{\phi,\lambda}\mathbf{S}_{\phi,\lambda} \neq \mathbf{S}_{\phi,\lambda}.$$

También es útil el filtro lineal definido por el proceso de suavizado para estimar la aceleración. Sea $\Phi^{(2)}$ la matriz que contiene los valores de las segundas derivadas de las funciones base evaluadas en los puntos de muestreo, es decir, $D^2\phi_k(t_j)$, y sea $\widehat{\mathbf{y}}^{(2)}$ el vector con las estimaciones de la aceleración en los puntos de muestreo. Entonces

$$\widehat{\mathbf{y}}^{(2)} = \Phi^{(2)}(\Phi^t\mathbf{W}\Phi + \lambda\mathbf{R})^{-1}\Phi^t\mathbf{W}\mathbf{y} = \mathbf{S}^{(2)}_{\phi,\lambda}\mathbf{y},$$

donde

$$\mathbf{S}^{(2)}_{\phi,\lambda} = \Phi^{(2)}(\Phi^t\mathbf{W}\Phi + \lambda\mathbf{R})^{-1}\Phi^t\mathbf{W}.$$

$\mathbf{S}^{(2)}_{\phi,\lambda}$ es la matriz que transforma el vector de datos $\mathbf{y}$ en el vector con las estimaciones de la aceleración $\widehat{\mathbf{y}}^{(2)}$.

Se pueden utilizar otras medidas de ajuste más generales en lugar de usar la suma de cuadrados de los residuos. Por ejemplo, si se tiene un modelo para los valores observados y_j y se puede obtener la log-verosimilitud de x, se podría considerar maximizar la expresión

$$\text{log-verosimilitud } x - \lambda PEN_2(x).$$

Además, también se pueden utilizar penalizaciones de la irregularidad más generales. Por ejemplo, si se desean estimar derivadas de orden m, se deben penalizar dos derivadas más altas $(m+2)$. Es decir, si se quieren estimar derivadas de orden 2, se debe utilizar como medida de penalización

$$PEN_4(x) = \int [D^4x(s)]^2 ds.$$

Elección del parámetro de suavizado λ

Cuando se ajustan los datos utilizando una penalización de la irregularidad en lugar de por mínimos cuadrados, se pasa de definir la suavidad en términos del número de funciones base K, a definir la suavidad en términos del parámetro de suavizado λ.

Los métodos más utilizados para elegir λ son el método de validación cruzada y el método de validación cruzada generalizada.

- **Método de validación cruzada.**

La idea básica de utilizar validación cruzada consiste en seleccionar una parte de la muestra para entrenar el modelo y otra para validar el modelo obtenido, comprobando que se ajusta a datos con los que no ha sido creado el modelo.

Una de las técnicas más utilizadas para elegir el parámetro λ es la conocida como *leave-one-out*. Esta técnica consiste en dejar tan sólo una única observación en la muestra de validación y ajustar los datos al resto de observaciones. Una vez los datos han sido ajustados, se estima el valor del ajuste para la observación que se ha dejado en la muestra de validación. Este procedimiento debe repetirse dejando cada vez una observación de la muestra en el conjunto de validación y, en todos los pasos, se debe sumar el error cometido en el ajuste. Se repite para distintos valores de λ y se escoge el que minimice la suma de cuadrados de todos los errores.

Este método tiene dos problemas:

- Presenta un elevado coste computacional.
- Suele favorecer el ajuste de variaciones de alta frecuencia, que se prefieren ignorar.

- **Método de validación cruzada generalizada.**

  Este método fue desarrollado por Craven y Wahba (1979) [12]. Originalmente se propuso como una versión más simple de la validación cruzada evitando la necesidad de resuavizar p veces. Además, también es más adecuado que el método de validación cruzada pues se ajusta menos al ruido de los datos. El criterio viene expresado como

$$GCV(\lambda) = \frac{n^{-1}SSE(\mathbf{y}|\boldsymbol{c})}{[n^{-1}\text{traza}(\mathbf{I} - \mathbf{S}_{\phi,\lambda})]^2},$$

  donde

$$\mathbf{S}_{\phi,\lambda} = \Phi(\Phi^t\mathbf{W}\Phi + \lambda\mathbf{R})^{-1}\Phi^t\mathbf{W},$$

  y $SSE(\mathbf{y}|\boldsymbol{c})$ es la suma de cuadrados del error,

$$SSE(\mathbf{y}|\boldsymbol{c}) = (\mathbf{y} - \Phi\boldsymbol{c})^t\mathbf{W}(\mathbf{y} - \Phi\boldsymbol{c}).$$

Análisis biresolución

El análisis biresolución presenta un enfoque más general donde el suavizado spline es uno de sus casos particulares. Permite utilizar un número elevado de funciones base, pero usando una penalización por irregularidad al ajustar la función a los datos observados.

Se parte de dos conjuntos de funciones base, $\phi_j, j = 1, \ldots, J$ y $\psi_k, k = 1, \ldots, K$ que se complementan entre sí. Las funciones ϕ_j son pequeñas en número y elegidas para describir las características de los datos a gran escala. Por contra, las funciones ψ_k son mayores en número y se utilizan para representar otras características locales no representadas por ϕ_j.

Se asume que cualquier función x se puede expresar en términos de las dos bases como

$$x(s) = \sum_{j=1}^{J} d_j\phi_j(s) + \sum_{k=1}^{K} c_k\psi_k(s) = x_S + x_R.$$

En general, las bases ϕ_j y ψ_k se escogen linealmente independientes.

La penalización de la irregularidad se hace depender únicamente de los coeficientes de ψ_k. Una posibilidad es tomar la norma L_2 de x_R,

$$PEN_0(x_R) = \int x_R(s)^2 ds = \int [\sum_{k=1}^{K} c_k \psi_k(s)]^2 ds.$$

Otra posibilidad es tomar un cierto orden de la derivada de x_R antes de realizar el cuadrado e integrar. Por ejemplo, se podría utilizar,

$$PEN_2(x_R) = \int (D^2 x_R)^2 = \int [\sum_{k=1}^{K} c_k D^2 \psi_k(s)]^2 ds,$$

para evaluar la importancia de x_R en términos de su curvatura total, medida por su segunda derivada al cuadrado.

Más generalmente, se podría usar cualquier operador diferencial L, definiendo

$$PEN_L(x_R) = \int (L x_R)^2 = \int [\sum_{k=1}^{K} c_k L\psi_k(s)]^2 ds.$$

Se puede expresar en forma matricial como

$$PEN_L(x_R) = \boldsymbol{c}^t \mathbf{R} \boldsymbol{c},$$

donde la matriz $\mathbf{R}$ contiene los elementos

$$\mathbf{R}_{kl} = \int L\psi_k(s) L\psi_l(s) ds.$$

Alternativamente, se puede especificar $\mathbf{R}$ directamente como una matriz simétrica no definida negativa, sin referencia a la irregularidad de x_R.

Consideramos ahora una función x de la forma

$$x(s) = \sum_{j=1}^{J} d_j \phi_j(s) + \sum_{k=1}^{K} c_k \psi_k(s) = x_S + x_R,$$

y definimos la penalización de la irregularidad como

$$PEN(x) = \boldsymbol{c}^t \mathbf{R} \boldsymbol{c}.$$

Para expresar la penalización de suma de cuadrados, es necesario expresar la suma de los cuadrados de los residuos en términos de los vectores de coeficientes $\mathbf{d}$ y $\boldsymbol{c}$,

$$\sum_{j=1}^{p} [y_j - x(t_j)]^2 = (\mathbf{y} - \Phi\mathbf{d} - \Psi\boldsymbol{c})^t (\mathbf{y} - \Phi\mathbf{d} - \Psi\boldsymbol{c}),$$

donde Ψ es una matriz $p \times K$ que contiene los elementos $\psi_k(t_j)$.
Se define la penalización de cuadrados del error como

$$PENSSE_\lambda(x|\mathbf{y}) = (\mathbf{y} - \Phi\mathbf{d} - \Psi\boldsymbol{c})^t(\mathbf{y} - \Phi\mathbf{d} - \Psi\boldsymbol{c}) + \lambda\boldsymbol{c}^t\mathbf{R}\boldsymbol{c} =$$

$$\mathbf{y}^t\mathbf{y} - \mathbf{y}^t\Phi\mathbf{d} - \mathbf{y}^t\Psi\boldsymbol{c} - \mathbf{d}^t\Phi^t\mathbf{y} + \mathbf{d}^t\Phi^t\Phi\mathbf{d} + \mathbf{d}^t\Phi^t\Psi\boldsymbol{c} - \boldsymbol{c}^t\Psi^t\mathbf{y} + \boldsymbol{c}^t\Psi^t\Phi\mathbf{d} + \boldsymbol{c}^t\Psi^t\Psi\boldsymbol{c} + \lambda\boldsymbol{c}^t\mathbf{R}\boldsymbol{c}.$$

Se puede minimizar esta forma cuadrática en términos de $\mathbf{d}$ y $\boldsymbol{c}$ para encontrar la función x.
La solución para $\mathbf{d}$ para cualquier valor fijo de $\boldsymbol{c}$ se obtiene derivando $PENSSE_\lambda(x|\mathbf{y})$ con respecto de $\mathbf{d}$ e igualando a cero. Derivando se obtiene la ecuación

$$-2\mathbf{y}\Phi^t + 2\Phi^t\Phi\mathbf{d} + 2\Phi^t\Psi\boldsymbol{c} = 0.$$

Despejando $\mathbf{d}$,

$$2\Phi^t\Phi\mathbf{d} = 2\mathbf{y}\Phi^t - 2\Phi^t\Psi\boldsymbol{c}$$
$$\widehat{\mathbf{d}} = (\Phi^t\Phi)^{-1}\Phi^t(\mathbf{y} - \Psi\boldsymbol{c}),$$

y consecuentemente,

$$\Phi\widehat{\mathbf{d}} = \mathbf{S}_\phi(\mathbf{y} - \Psi\boldsymbol{c}),$$

donde la matriz de proyección $\mathbf{S}_\phi$ es

$$\mathbf{S}_\phi = \Phi(\Phi^t\Phi)^{-1}\Phi^t.$$

Si sustituimos esta solución para $\mathbf{d}$ en la expresión $PENSSE_\lambda(x|\mathbf{y})$, se obtiene

$$PENSSE_\lambda(x|\mathbf{y}) = \mathbf{y}^t\mathbf{y} - \mathbf{y}^t\Phi[(\Phi^t\Phi)^{-1}\Phi^t(\mathbf{y} - \Psi\boldsymbol{c})] - \mathbf{y}^t\Psi\boldsymbol{c} - [(\Phi^t\Phi)^{-1}\Phi^t(\mathbf{y} - \Psi\boldsymbol{c})]^t\Phi^t\mathbf{y}$$
$$+[(\Phi^t\Phi)^{-1}\Phi^t(\mathbf{y} - \Psi\boldsymbol{c})]^t\Phi^t\Phi[(\Phi^t\Phi)^{-1}\Phi^t(\mathbf{y} - \Psi\boldsymbol{c})] + [(\Phi^t\Phi)^{-1}\Phi^t(\mathbf{y} - \Psi\boldsymbol{c})]^t\Phi^t\Psi\boldsymbol{c} -$$
$$\boldsymbol{c}^t\Psi^t\mathbf{y} + \boldsymbol{c}^t\Psi^t\Phi[(\Phi^t\Phi)^{-1}\Phi^t(\mathbf{y} - \Psi\boldsymbol{c})] + \boldsymbol{c}^t\Psi^t\Psi\boldsymbol{c} + \lambda\boldsymbol{c}^t\mathbf{R}\boldsymbol{c}.$$

Derivando ahora con respecto de $\boldsymbol{c}$ se tiene la ecuación

$$2\Psi^t\Phi(\Phi^t\Phi)^{-1}\Phi^t\mathbf{y} - 2\Psi^t\mathbf{y} + 2\Psi^t\Psi\boldsymbol{c} - 2\Psi^t\Phi(\Phi^t\Phi)^{-1}\Phi^t\Psi\boldsymbol{c} + 2\lambda\mathbf{R}\boldsymbol{c} = 0,$$

y como $\mathbf{S}_\phi = \Phi(\Phi^t\Phi)^{-1}\Phi^t$,

$$2\Psi^t\mathbf{S}_\phi\mathbf{y} - 2\Psi^t\mathbf{y} + 2\Psi^t\Psi\boldsymbol{c} - 2\Psi^t\mathbf{S}_\phi\Psi\boldsymbol{c} + 2\lambda\mathbf{R}\boldsymbol{c} = 0.$$

Definiendo $\mathbf{Q}_\phi$ como

$$\mathbf{Q}_\phi = \mathbf{I} - \mathbf{S}_\phi,$$

la ecuación anterior se puede escribir

$$-\Psi^t\mathbf{Q}_\phi\mathbf{y} + \Psi^t\mathbf{Q}_\phi\Psi\boldsymbol{c} + \lambda\mathbf{R}\boldsymbol{c} = 0.$$

Despejando $\mathbf{c}$,

$$\Psi^t\mathbf{Q}_\phi\Psi\boldsymbol{c} + \lambda\mathbf{R}\boldsymbol{c} = \Psi^t\mathbf{Q}_\phi\mathbf{y}$$
$$\boldsymbol{c}(\Psi^t\mathbf{Q}_\phi\Psi + \lambda\mathbf{R}) = \Psi^t\mathbf{Q}_\phi\mathbf{y}$$
$$\widehat{\boldsymbol{c}} = (\Psi^t\mathbf{Q}_\phi\Psi + \lambda\mathbf{R})^{-1}\Psi^t\mathbf{Q}_\phi\mathbf{y}$$

La penalización de la irregularidad se controla mediante el parámetro λ. Si $\lambda \approx 0$, no se aplica ninguna penalización y las estimaciones obtenidas minimizando el criterio $PENSSE_\lambda(x|\mathbf{y})$ son las obtenidas por una expansión ordinaria con $\{\phi_j\}$ y $\{\psi_k\}$. Por otro lado, cuando $\lambda \longrightarrow \infty$ la penalización de la irregularidad es máxima y la contribución $\{\psi_k\}$ es prácticamente cero. Si $\mathbf{R}$ es definida positiva, se obtiene la estimación correspondiente sólo a las bases $\{\phi_j\}$. Si $\mathbf{R}$ no es definida positiva, aparece una contribución $\{\psi_k\}$ si $Lx_R(s) = 0 \ \forall s$.

El suavizado spline es un caso particular del análisis biresolución. En el suavizado spline no se especifica $\{\phi_j\}$, sólo $\{\psi_k\}$, siendo ésta la base B-spline. Se escogen los nodos en los datos y $L = D^2$.

Ajuste de funciones restringidas: positivas y monótonas

- **Ajuste de funciones positivas.**

  Una función x suave y positiva siempre se puede definir como una función exponencial de la forma

  $$x(t) = e^{W(t)},$$

  de modo que $W(t)$ es el logaritmo de $x(t)$. Como $W(t)$ no presenta ninguna restricción, se puede expandir en términos de un conjunto de funciones base,

  $$W(t) = \sum_k c_k \phi_k(t).$$

  La irregularidad de una función positiva se define como la irregularidad de su logaritmo W, luego el criterio a minimizar, tomando en este caso la segunda derivada es

  $$PENSSE_\lambda(W|\mathbf{y}) = (\mathbf{y} - e^{W(t)})^t \mathbf{W} (\mathbf{y} - e^{W(t)}) + \lambda \int [D^2 W(t)]^2 dt.$$

  Para minimizar este criterio con respecto a los coeficientes c_k de la expansión, se deben utilizar métodos numéricos. Estos métodos disminuyen iterativamente una estimación inicial de $W(t)$ hasta que se alcanza la convergencia. Sin embargo, como la función exponencial es prácticamente lineal, los métodos convergen rápidamente. De hecho, comenzar con $W = 0$ suele funcionar bastante bien en la mayoría de los casos.

- **Ajuste de funciones monótonas.**

  En este caso la solución está ligada al problema de estimar una función positiva pues, en este caso, Dx se asume una función positiva. Por lo tanto,

  $$Dx(t) = e^{W(t)}.$$

  Integrando ambos lados de la ecuación,

  $$x(t) = C + \int_{t_0}^{t} e^{W(u)} du,$$

  donde C es una constante que se debe estimar a partir de los datos.

3.8. Registro

Una vez que las observaciones se han transformado a la forma funcional, el siguiente paso consiste en considerar métodos para realizar su análisis. Sin embargo, todavía se deben solucionar algunos problemas importantes antes de proceder a analizar los datos.

A menudo, la variación en las observaciones funcionales implican tanto variaciones en fase como variaciones en amplitud y confundir ambos términos puede conllevar muchos problemas.

Los valores de $x_i(t_j)$ pueden diferir en:

- **Amplitud**: Para un tiempo t los valores de las distintas funciones son diferentes.
- **Fase**: Las funciones x_1 y x_2 no son comparadas en los mismos instantes temporales.

Para resolver este problema, se debe hacer enfásis en el registro de los datos, transformando los argumentos t en lugar de los valores $x(t)$.

Existen diferentes tipos de registro entre los que podemos destacar el registro mediante puntos característicos, el registro global asumiendo un modelo paramétrico o el registro de ajuste continuo.

Se considera $T = [T_1, T_2]$ el intervalo sobre el que registrar las funciones y, además, también se asume que las funciones están definidas en un intervalo mayor que contiene a T.

- **Registro mediante puntos característicos.**

  Un punto característico de una curva es una singularidad que se puede asociar a un valor específico de t. Éstos son generalmente, máximos, mínimos, curvas que pasan por el cero, y pueden ser identificados a nivel de algunas derivadas.

  Consideramos ahora el problema más general de estimar una posible transformación no lineal h_i y veamos cómo se pueden utilizar los puntos característicos para estimar esta transformación.

  Este proceso de registro requiere para cada curva x_i, la identificación de los valores $t_{if}, f = 1, \ldots, F$ asociados a cada una de las F características. El objetivo es construir una transformación h_i, a la que llamaremos función *warping*, para cada curva de modo que

  $$x^*(t) = x_i(h_i(t)),$$

  tenga valores muy parecidos en los puntos característicos para cualquier i.

- **Registro global asumiendo un modelo paramétrico.**

Una de las transformaciones más simples es el llamado registro *Shift*. Este registro está basado en una traslación, de modo que $h_i(t) = t + \delta_i$. Luego,

$$x_i^* = x_i(t + \delta_i),$$

donde δ_i es el retraso o adelanto de la i-ésima prueba que se realiza (que se elige para alinear la curvas de forma adecuada).

Otra transformación que también se suele realizar, es la transformación lineal donde $h_i = (t + \delta_i)\beta_i$ con $\beta_i > 0$. Generalmente se asume $h_i(t) = h_i(t|\gamma_i)$ siendo γ_i un vector de parámetros.

Para estimar la transformación se utiliza el método *Procrustes* iterativamente según el siguiente procedimiento:

1. Se calcula la media de las funciones

$$\widehat{\mu} = \frac{\sum_{i=1}^{n} x_i(t)}{n}.$$

2. Se determinan los γ_i que minimizan la siguiente función utilizando Newton-Raphson.

$$REGSSE = \sum_{i=1}^{n} \int_T [x_i(h_i(t|\gamma_i)) - \widehat{\mu}(s)]^2 ds.$$

Los pasos se iteran hasta alcanzar la convergencia.

- **Criterio de ajuste continuo.**

  En lugar del criterio *REGSSE*, otro criterio más eficaz se basa en la minimización del segundo mayor valor propio de una cierta matriz de productos cruzados. Es el criterio utilizado por defecto en la librería *fda* de R.

Capítulo 4

Outliers de datos funcionales

4.1. Introducción

Los datos funcionales aparecen cada vez más frecuentemente en la práctica. Por ello, se han desarrollado varias técnicas estadísticas, generalizando las técnicas conocidas en estadística univariante y multivariante, para poder analizarlos.

Una de las ramas de estudio más importantes de la estadística es el análisis de la calidad de los datos, ya que datos con problemas (heterogeneidad, faltantes, etc.) pueden conducir a decisiones erróneas con graves consecuencias. Entre los posibles problemas que pueden presentar los datos, se encuentran los conocidos como valores atípicos (*outliers*). Según Hawkins, un *outlier* es 'una observación que se desvía mucho de otras observaciones y despierta sospechas de ser generada por un mecanismo diferente' [13].

Hay una extensa literatura sobre la búsqueda de valores atípicos y el desarrollo de métodos para localizarlos. Se puede consultar, por ejemplo, P. Rousseeuw y A. Leroy (1987) [14] y R. Maronna, D. Martin y V. Yohai (2006) [15]. Pero, para el análisis de datos funcionales, la detección de *outliers* es un campo todavía muy reciente, limitado prácticamente a curvas univariantes, donde para cada curva y para cada instante de tiempo, se tiene tan sólo una única observación. Se considera que una curva es un *outlier* si ha sido generada por un proceso estocástico con una distribución diferente que el resto de curvas, que se asume que están idénticamente distribuidas. Por tanto, se asume que todas las curvas vienen dadas por el mismo proceso estocástico, y las curvas que no son compatibles con esta suposición, son las consideradas *outliers*. M. Febrero, P. Galeano y W. González-Manteiga (2008) [16] identificaron dos razones por las que pueden aparecer *outliers* en los datos funcionales:

- **Pueden ser causados por errores de medición.** Estos errores deben ser identificados y corregidos si es posible, para evitar su uso en análisis posteriores.
- **Pueden ser observaciones correctas, que son sospechosas de ser erróneas, en el sentido que no siguen el mismo patrón que la mayoría de las curvas.** Este

tipo de *outliers* ayuda a detectar anomalías en el sistema.

En este capítulo se explican las tipologías de *outliers* de datos funcionales y las metodologías que existen para su detección, en los paquetes del software R.

4.2. Tipologías de *outliers* para datos funcionales

En esta sección se establece una taxonomía de *outliers* de datos funcionales para poder clasificar los diferentes tipos de comportamiento que pueden presentar, según M. Hubert, P. Rousseeuw y P. Segaert (2015) [17].

Las observaciones funcionales pueden desviarse de la mayoría de las curvas en diferentes formas:

- ***Outliers* aislados.**

  Presentan un comportamiento extraño durante un intervalo breve de tiempo. En la Figura 4.1 se muestra un ejemplo de este tipo de *outliers*. La imagen representa el espectro de resonancia magnética nuclear (NMR) de los protones de 40 muestras de vino (F. Larsen, F. van den Berg y S. Engelsen (2006)) [18]. La curva de color rojo, presenta valores más elevados alrededor de 5.4, por lo que se considera un *outlier* aislado.

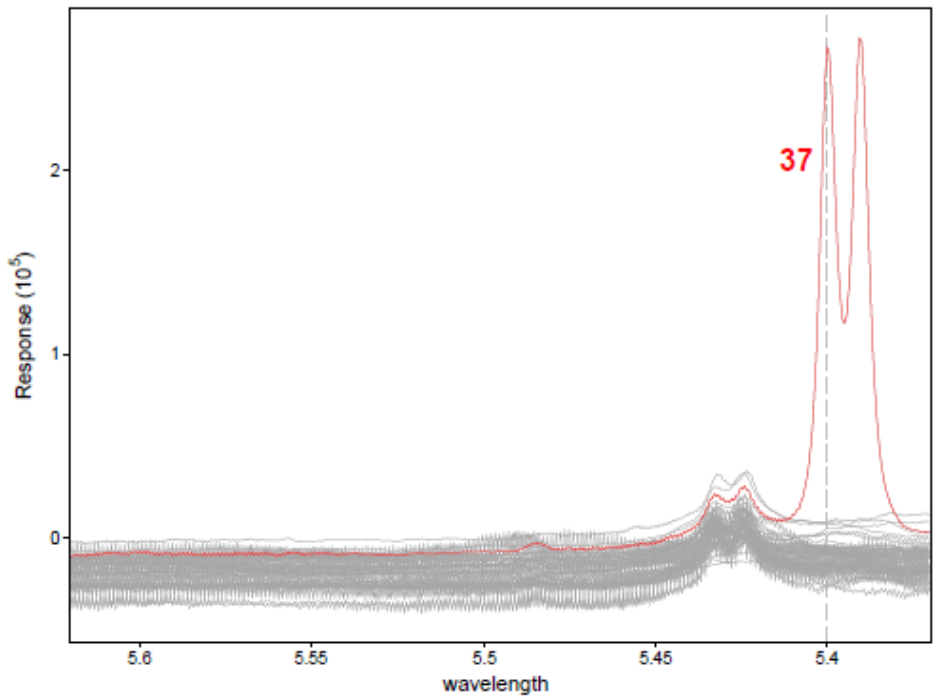


Figura 4.1: Espectro de resonancia magnética nuclear para 40 muestras de vino, con un *outlier* aislado.

- ***Outliers* persistentes**

  Es el caso opuesto a los *outliers* aislados. Se definen como observaciones funcionales que **presentan un comportamiento extraño durante la mayor parte de su dominio**. Dentro de este grupo, se distinguen tres tipos de *outliers*: desplazados, en la forma o en amplitud.

  - ***Outliers* desplazados**

    Presentan la misma forma que la mayor parte de las curvas pero de manera desplazada. Un ejemplo de este tipo de *outliers* se muestra en la Figura 4.2.

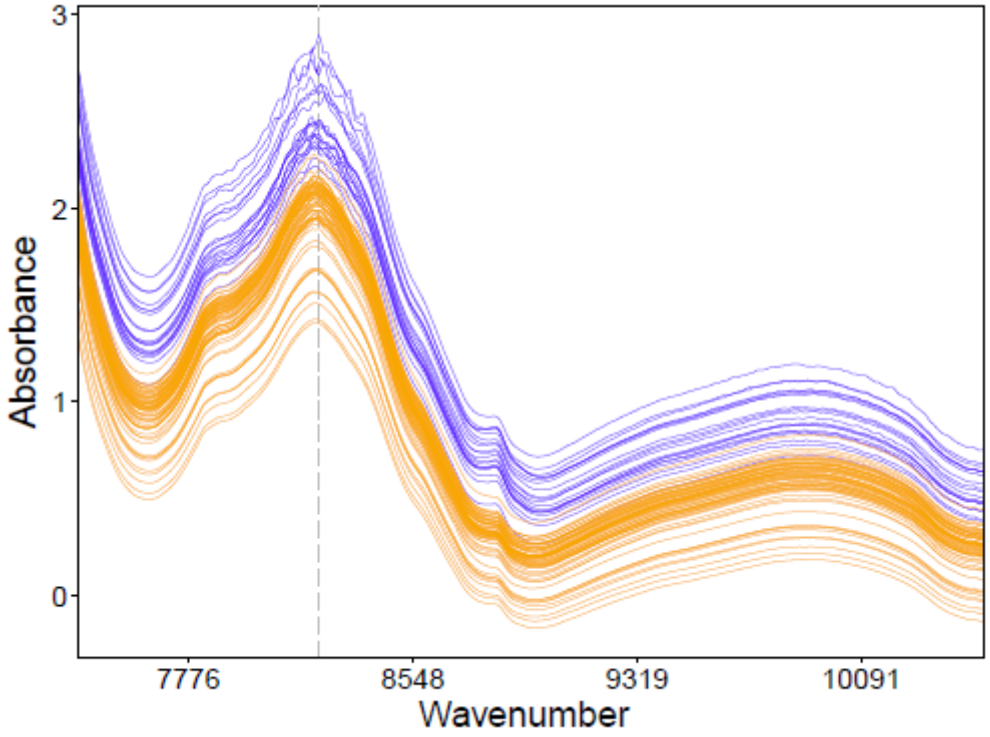


Figura 4.2: Respuestas de espectroscopia infrarroja para un lote de pastillas. Las pastillas de 90 mg se representan de color naranja y las de 250 mg de color azul.

La imagen representa las respuestas de espectroscopia infrarroja para un lote de pastillas (Dyrby et al (2002)) [19]. El conjunto de datos completo consiste en pastillas de diferentes tamaños y con diferentes cantidades de sus componentes. Se han seleccionado las 70 observaciones correspondientes a comprimidos de 90 mg (mostrados en naranja) y se han añadido 20 comprimidos de 250 mg (mostrados en azul). Estos dos grupos de pastillas también difieren en la cantidad de componentes que contienen. En el grupo de los comprimidos de 90 mg, se observa que las curvas con valores más bajos son muy similares al resto pero con valores inferiores, por lo que se pueden clasificar como *outliers* desplazados.

- ***Outliers* en la forma**

  Presentan una forma distinta al resto de curvas pero sin destacar necesariamente en ningún instante de tiempo. Un ejemplo de *outliers* en la forma se muestra en la Figura 4.3, donde se representan las primeras derivadas de las curvas de la Figura 4.2. Ahora, las curvas azules tienen una forma diferente alrededor de la línea gris punteada. En comparación con el grupo de los comprimidos de 90 mg, podemos considerar que los comprimidos de 250 mg constituyen valores atípicos en la forma.

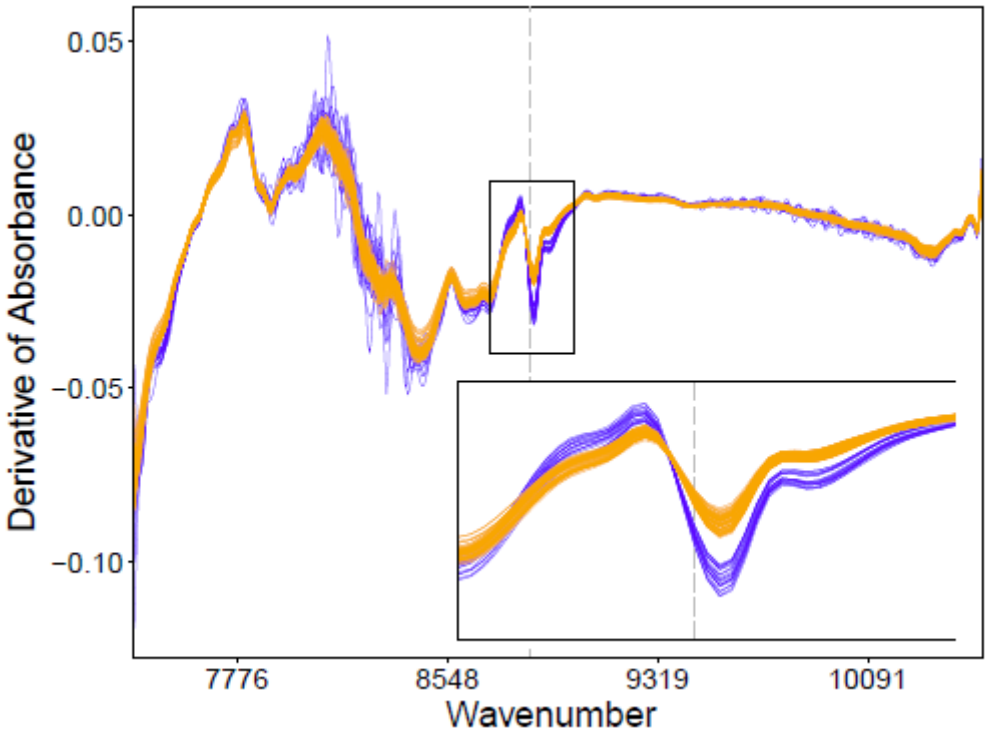


Figura 4.3: Derivadas de las curvas de la Figura 4.2

.

- ***Outliers* en amplitud**

  Presentan la misma forma que el resto de curvas pero difieren en su escala (rango, amplitud, etc.). Un ejemplo de *outliers* en la magnitud de los datos se muestra en la Figura 4.4. En este gráfico se representan las coordenadas horizontales y verticales del trazado realizado por diferentes personas para representar la letra '*i*' (sin el punto). En la imagen se han marcado tres curvas que presentan diferentes comportamientos. La curva roja, tiene la misma forma que el resto de curvas, pero es más pequeña; por tanto, puede ser considerada como un *outlier* en amplitud. Por otro lado, la observación azul se desvía del resto de trayectorias durante un largo periodo de tiempo, especialmente en la componente vertical, por tanto, es un ejemplo de *outlier* en la forma. La curva negra, sería también otro ejemplo de *oulier* en la forma.

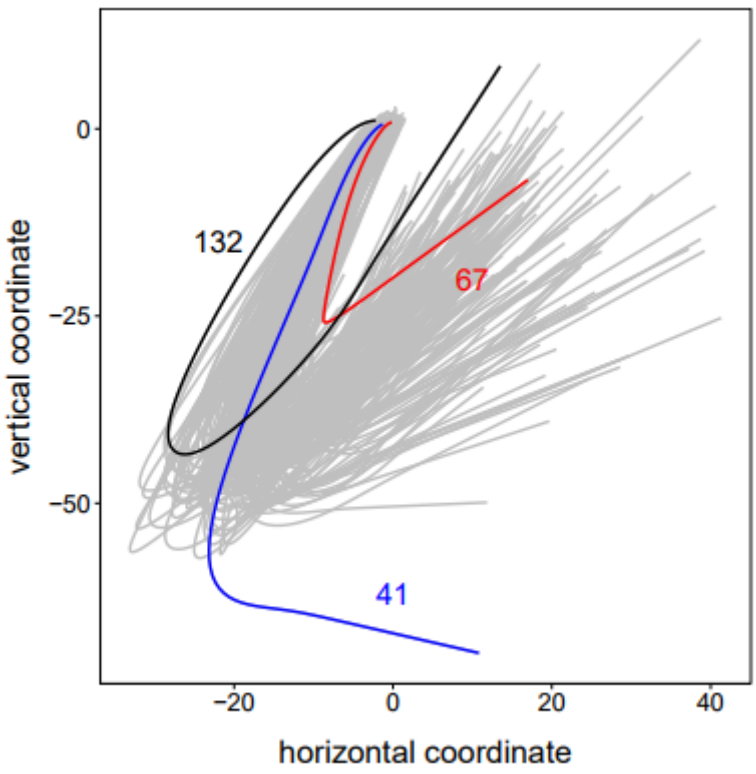


Figura 4.4: Trayectorias realizadas al escribir la letra '*i*' (sin el punto). La curva roja es un ejemplo de *outlier* en la magnitud de los datos y las curvas azul y negro representan *outliers* en la forma de los datos.

4.3. Metodologías de R para detectar *outliers* de datos funcionales

El estudio de atípicos es importante en cualquier análisis estadístico de datos. Para la detección de datos atípicos funcionales mediante el software R, se puede utilizar la instrucción *foutliers* incluida en el paquete *rainbow* y la instrucción *fbplot* contenida en el paquete *fda*. En esta sección se explican las distintas metodologías contenidas en ambos paquetes para la detección de *outliers* funcionales. Además, también se describe en qué consiste el análisis basado en arquetipos, ya que también puede servir para detectar *outliers*.

4.3.1. Metodologías *foutliers*

Existen varios métodos para obtener *outliers* funcionales utilizando la instrucción *foutliers*. Los métodos que se describen en esta sección son: 'lrt', 'depth.trim', 'depth.prod' y 'HUoutliers'.

Los tres primeros métodos están basados en el uso de la profundidad funcional. La profundidad funcional es una generalización de la profundidad multivariante, de tal modo que permite ordenar una muestra de forma análoga a como lo hace la mediana para muestras univariantes. La profundidad multivariante mide cómo de profundo está un punto teniendo en cuenta una distribución de probabilidad o respecto a una nube de datos. Esto lleva a construir un ranking de los puntos desde el centro hacia el exterior: los puntos más profundos son aquellos más cercanos al centro de la nube de puntos y los más alejados son los que tienen menos profundidad. Así pues, la profundidad funcional permite ordenar una muestra de curvas desde el centro hacia fuera, donde la curva con mayor profundidad se corresponde con la mediana, mientras que los atípicos se definen como las curvas de menor profundidad. El último de estos métodos, 'HUoutliers', utiliza una versión robusta de componentes principales para detectar *outliers*.

Método 'lrt':

El método 'lrt', es un método propuesto por M. Febrero, P. Galeano y W. González-Manteiga (2007) [20].

Recordemos que si $z_1, \ldots, z_n$ es una muestra que sigue una distribución normal, el estadístico utilizado para probar si la observación z_i, sigue la distribución normal, es decir, es o no un *outlier* es:

$$LRT(z_i) = \frac{z_i - \overline{z}}{\widehat{\sigma}},\ i = 1, \ldots, n,$$

donde $\overline{z}$ y $\widehat{\sigma}$ son la media y la desviación típica muestral respectivamente. En la práctica, el número de *outliers* y su localización son desconocidos a priori, por tanto, se debe probar para cada observación y emplear el estadístico:

$$\lambda = \max_{1 \leq i \leq n} |LRT(z_i)|.$$

Comparando este estadístico con algún punto de corte, e iterando el proceso, se puede determinar la presencia de *outliers*. Si las observaciones de la muestra no han sido extraídas de una distribución normal, el test de razón de verosimilitud puede verse como un test de quasi-verosimilitud y todavía funciona bien.

Ni la media ni la desviación típica muestral son estadísticos robustos a la presencia de *outliers* y, esto produce un efecto denominado 'enmascarado': *outliers* grandes aumentan la desviación típica enmascarando la presencia de otros *outliers*. Para evitar este efecto, la media y la desviación típica se deben reemplazar por otros estimadores más robustos. La media se puede sustituir por la mediana o la media truncada. Para calcular la media truncada, en general, se descartan porciones de la muestra en el extremo inferior y superior, típicamente entre el 5 y el 25 % de los elementos, contando ambos extremos. Si se ordenan las curvas en orden decreciente conforme a su profundidad, se obtienen las curvas ordenadas $x_{(1)}, \ldots, x_{(n)}$, de modo que $x_{(1)}$ es la curva con mayor profundidad y $x_{(n)}$, la curva con menor profundidad. Entonces, la media truncada se define como:

$$\widehat{\mu}_{TM,\alpha} = \frac{1}{n - [\alpha n]} \sum_{i=1}^{n-[\alpha n]} x_{(i)},$$

donde α se elige dependiendo del número de elementos de la muestra que se desean descartar para realizar la media.

La desviación típica se puede reemplazar por la desviación absoluta media (se calcula como la mediana del valor absoluto de la diferencia entre cada valor menos la mediana del grupo) o por la desviación típica truncada. La idea de la desviación típica truncada es similar a la de la media truncada: se obtiene la desviación típica de los puntos más profundos. Se define como:

$$\widehat{\sigma}_{TSD,\alpha} = \left(\frac{1}{n - [\alpha n]} \sum_{i=1}^{n-[\alpha n]} (x_{(i)} - \widehat{\mu}_{TM,\alpha})^2 \right)^{\frac{1}{2}}.$$

Como se está considerando la variación de las curvas con respecto a una media truncada, se espera que $\widehat{\sigma}_{TSD,\alpha}$ sea menos afectada que la desviación típica usual, por las curvas extremas, porque las curvas con menor profundidad no se tienen en cuenta en el cálculo.

Siguiendo el razonamiento para muestras univariantes, se procede del siguiente modo. Sea

$$O_\alpha(x_i) = \left\| \frac{x_i - \widehat{\mu}_{TM,\alpha}}{\widehat{\sigma}_{TSD,\alpha}} \right\|,$$

donde $\| \ \|$ es la norma en el espacio de funciones ($\| \ \|_1$, $\| \ \|_2$ ó $\| \ \|_\infty$), $\widehat{\mu}_{TM,\alpha}$ es la media truncada y $\widehat{\sigma}_{TSD,\alpha}$ es la desviación típica truncada. Por tanto, $O_\alpha(x_i)$ es la distancia entre x_i y $\widehat{\mu}_{TM,\alpha}$ relativa a $\widehat{\sigma}_{TSD,\alpha}$.

El proceso para la detección de *outliers* es el siguiente:

Procedimiento de detección de *outliers* funcionales.

1. Dadas las observaciones funcionales $x_1, \ldots, x_n$, se obtiene el estadístico:

$$\Lambda = \max_{1 \leq i \leq n} O_\alpha(x_i).$$

2. Sea x_I la curva que alcanza el máximo para el estadístico Λ. Si $\Lambda = O_\alpha(x_I) > C$, se asume que x_I es un *outlier* y se elimina de la muestra.

Los pasos se repiten hasta que no se encuentran más *outliers*.

Para encontrar el valor de corte, C, se utiliza un procedimiento bootstrap. En estadística, los métodos bootstrap se corresponden con cualquier prueba que se basa en el muestreo aleatorio con reemplazo. Permite asignar medidas de precisión (definidas en términos de sesgo, varianza, intervalos de confianza, error de predicción o alguna otra medida de este tipo) a las estimaciones de la muestra. El procedimiento bootstrap para obtener C consiste en los siguientes pasos:

Procedimiento bootstrap para obtener el valor de corte C

1. Sea $y_i^b, i = 1, \ldots, n$ y $b = 1, \ldots B$, las B curvas suavizadas bootstrap. Para cada $b = 1, \ldots, B$, se obtiene:

$$\Lambda^b = \max_{1 \leq i \leq n - [\alpha n]} O_\alpha \left(y_{(i)}^b \right),$$

donde $y_{(i)}^b, i = 1, \ldots, n$ son las curvas bootstrap suavizadas ordenadas en base a sus profundidades y α es el valor escogido para obtener $\widehat{\mu}_{TM,\alpha}$ y $\widehat{\sigma}_{TSD,\alpha}$ utilizado en el proceso de detección de *outliers.*

2. El máximo valor de $\Lambda^1, \ldots, \Lambda^B$ es el elegido como el valor de corte utilizado en el proceso de detección de *outliers.*

Es importante observar que se calculan los valores de $\Lambda^1, \ldots, \Lambda^B$ utilizando sólo las $n - [\alpha n]$ curvas más profundas de las curvas bootstrap suavizadas. Esto se hace para evitar la presencia de valores atípicos en las curvas bootstrap. Si la muestra no tiene *outliers*, esta elección puede no ser apropiada porque el valor de C estará sesgado hacia abajo. Por tanto, se evita la detección de falsos *outliers* tomando el valor de corte como el máximo de $\Lambda^1, \ldots, \Lambda^B$, que se espera que sea bastante grande. Una elección adecuada para α es la proporción de curvas sospechosas de ser *outliers* en la muestra.

Métodos 'depth.trim' y 'depth.prod':

Los métodos 'depth.trim' y 'depth.prod' fueron propuestos por M. Febrero, P. Galeano y W. González-Manteiga (2008) [16].

Están basados en la profundidad de los datos funcionales, asumiendo la independencia de los mismos y obteniendo un contraste de hipótesis para saber si cada una de las curvas de un conjunto de datos es atípica o no. Se entiende que los valores atípicos se encuentran entre las curvas con poca profundidad. De esta forma, el método tiene como objetivo calcular un percentil, punto de corte, que determina que los datos funcionales con profundidad asociada menor que dicho percentil son atípicos.

Partiendo de un conjunto de curvas formado por n datos funcionales: $x_1, \ldots, x_n$, se procede como sigue:

1. Calcular la profundidad (D) asociada a cada curva del conjunto de datos, $D(x_1), \ldots, D(x_n)$.

2. Sean $x_1, \ldots, x_k$ las k curvas cuya profundidad es menor o igual que el punto de corte C, es decir, $D(x_k) \leq C$. Las k curvas son los valores atípicos detectados y se eliminan del conjunto de datos.

3. Se vuelve al paso 1 y se repite el proceso hasta que no se detecten más valores atípicos. Este paso es necesario, ya que al detectar las curvas que son atípicas y eliminarlas de la muestra, el conjunto de datos se reduce. Se tiene entonces un nuevo conjunto de datos cuyas profundidades deben ser recalculadas y, en función de esas nuevas profundidades, se pueden detectar o no nuevos valores atípicos que hasta el momento estaban 'enmascarados' por los demás.

El punto C se calcula eligiendo el error de tipo I en el contraste que fija como hipótesis nula que un dato no es atípico. Se fija dicho error en el 1 %, de forma que, en ausencia de atípicos, el porcentaje de curvas mal clasificadas como atípicas sea del 1 %. Por lo tanto el valor C seleccionado sería aquel que verifique:

$$P(D(x) \leq C | x \text{ no atípico}) = 0,01.$$

Sin embargo, la distribución de las profundidades funcionales es desconocida, por lo que no se puede calcular C como el primer percentil de dicha distribución. El punto C se encuentra estimando este percentil haciendo uso de las curvas de la muestra. Como la muestra puede tener *outliers*, hay que obtener un estimador robusto del percentil basado en la muestra. Para ello, se utilizan dos alternativas de procedimientos bootstrap.

El proceso mediante el que se calcula el punto de corte C es el siguiente:

- Calcular la profundidad asociada a cada curva del conjunto de datos, $D(x_1), \ldots, D(x_n)$.
- Limpiar la muestra de posibles atípicos. Este paso es necesario para obtener una estimación. Para ello:
  - **Método 'depth.trim'**: Se obtienen B muestras bootstrap de tamaño n del conjunto de datos después de eliminar el α % de las curvas. El nivel α utilizado puede ser elegido como la proporción de valores atípicos sospechosos en la muestra. Las curvas bootstrap se denotan por x_i^b, con $i = 1, \ldots, n$ y $b = 1, \ldots, B$. A continuación, se obtienen las muestras bootstrap suavizadas, $y_i^b = x_i^b + z_i^b$ donde z_i^b está normalmente distribuida con media cero y matriz de covarianzas $\gamma \sum_x$, siendo γ el parámetro de suavizado y $\sum_x$ la matriz de covarianzas del conjunto de datos del que previamente se han eliminado los valores atípicos. Se considera el parámetro de suavizado $\gamma =$ 0,05.
  - **Método 'depth.prod'**: Se basa en muestrear las curvas con probabilidad proporcional a su profundidad. Así, se muestrean con más frecuencia los puntos más profundos tratando de evitar la presencia de valores atípicos en las muestras de bootstrap. Las curvas bootstrap se denotan por x_i^b, con $i = 1, \ldots, n$ y $b = 1, \ldots, B$ y se obtienen las muestras bootstrap suavizadas al igual que en el procedimiento anterior.
- Para cada conjunto bootstrap $b = 1, \ldots, B$, obtener C^b como el percentil 1 % de la distribución de las profundidades $D(y_i^b)$.
- Obtener C como la mediana de los B valores de C^b.

El cálculo de C se realiza solamente una vez, mantiendo el mismo valor en las distintas iteraciones del proceso. El cálculo de dicho punto de corte podría actualizarse en cada iteración, pero esta opción se rechaza debido al incremento del coste computacional requerido para ello y al planteamiento de C como un estimador robusto del percentil señalado.

Cálculo de la profundidad funcional

Los métodos que se han visto hasta ahora hacen uso de la profundidad funcional para la detección de *outliers*. A continuación, se definen algunas de las medidas de profundidad que se pueden utilizar.

- **Profundidad Fraiman-Muniz (FM):** Está fue la primera medida de profundidad propuesta para datos funcionales y fue introducida por Fraiman y Muniz (2001) [21].

  Sea $F_{n,t}(x_i(t))$ la distribución empírica acumulada de los valores de las curvas $x_1(t), \ldots, x_n(t)$ en un instante de tiempo $t \in [a, b]$. Entonces, la profundidad funcional de Fraiman y Muniz de una curva x_i con respecto al conjunto $x_1, \ldots, x_n$ viene dada por

  $$FM_n(x_i) = \int_a^b D_n(x_i(t))dt,$$

  donde $D_n(x_i(t))$ es la profundidad univariada de la curva $x_i(t)$ dada por

  $$D_n(x_i(t)) = 1 - |\frac{1}{2} - F_{n,t}(x_i(t))|,$$

  para cada $1 \leq i \leq n$.

- **Profundidad modal (MD):** Esta profundidad funcional fue introducida por A. Cuevas, M. Febrero, R. Fraiman (2006) [22].

  Los autores definen la moda funcional como la curva más densamente rodeada por el resto de curvas. La profundidad modal de una curva x_i con respecto al conjunto de curvas $x_1, \ldots, x_n$ viene dada por

  $$MD_n(x_i, h) = \sum_{k=1}^{n} K\left(\frac{||x_i - x_k||}{h}\right),$$

  donde $\| \ \|$ es una norma en el espacio funcional, $K : R^+ \to R^+$ es una función kernel y h es un ancho de banda. Entonces, la moda funcional se define como la curva que alcanza el máximo valor de $MD_n(x_i, h)$.

  En la práctica, se debe escoger una norma, una función kernel y un ancho de banda h. A. Cuevas, M. Febrero, R. Fraiman (2006) recomiendan utilizar las normas L^2 y L^∞ dadas por

  $$||x_i - x_k||_2 = \left(\int_a^b (x_i(t) - x_k(t))^2 dt\right)^{\frac{1}{2}},$$
  $$||x_i - x_k||_\infty = \sup_{t \in (a,b)} |x_i(t) - x_k(t)|,$$

  y el kernel Gaussiano truncado:

  $$K(t) = \frac{2}{\sqrt{2\pi}} \exp\left(\frac{-t^2}{2}\right), \ t > 0.$$

El ancho de banda escogido es el percentil 15 de la distribución empírica de $\{||x_i - x_k||,\ i,k = 1,\ldots,n\}$ aunque una amplia gama de valores es aceptada siempre que no sean muy pequeños.

- **Profundidad de proyección aleatoria (RPD):** A. Cuevas, M. Febrero, R. Fraiman (2007) [23] consideraron la profundidad de proyección aleatoria basada en medir la profundidad de los datos funcionales bajo proyecciones y tomando información adicional sobre sus derivadas. La idea básica consiste en proyectar cada curva funcional y su primera derivada sobre una dirección aleatoria, definiendo un punto en $\mathbb{R}^2$. Una profundidad de datos en $\mathbb{R}^2$ proporciona un orden de los puntos proyectados. Si se utiliza un gran número de proyecciones aleatorias, el valor medio de las profundidades de las proyecciones define una profundidad para los datos funcionales. Dado el conjunto de curvas $x_1,\ldots,x_n$ y una dirección v, $T_{i,v} =< v, x_i >$ es la proyección de x_i sobre la dirección v, es decir,

$$T_{i,v} =< v, x_i >= \int_a^b v(t)x_i(t)dt.$$

De modo similar, $T'_{i,v} =< v, x'_i >$ es la proyección de la derivada de x_i, x'_i, en la dirección v. Por tanto, el par $(T_{i,v}, T'_{i,v})$ es un punto en $\mathbb{R}^2$. Ahora, si $v_1,\ldots,v_p$ son P direcciones aleatorias independientes, la profundidad de proyección aleatoria de una curva x_i se define como

$$RPD_n(x_i) = \frac{1}{P}\sum_{p=1}^{P} D_n(< v_p, x_i >, < v_p, x'_i >),$$

donde D_n es cualquier profundidad definida en $\mathbb{R}^2$ del punto $(< v_p, x_i >, < v_p, x'_i >) \in \mathbb{R}^2$.

Método 'HUOutliers':

El método 'HUOutliers', es un método propuesto por R. J. Hyndman y M.S. Ullah (2007) [24]. Se basa en una versión robusta de componentes principales.

El análisis de componentes principales (PCA) es una de las técnicas utilizadas para reducir la dimensión en muestras multivariantes. Esta técnica puede extenderse a datos funcionales dando lugar al análisis de componentes principales funcionales FPCA.

Consideramos la variable aleatoria $\mathbf{X} \in \mathbb{R}^d$. Sin pérdida de generalidad asumimos que la variable considerada tiene media cero. PCA especifica las d direcciones $\{\mathbf{v}_k\}_{k=1}^d \in \mathbb{R}^d$ que maximizan la varianza de cada componente, sujeto a la condición de ortonormalidad. El objetivo es encontrar los vectores $\mathbf{v}_k$ tal que la varianza de

$$\alpha_k = \mathbf{v}_k^t\mathbf{X},$$

se maximice sujeto a

- $\mathbf{v}_k^t\mathbf{v}_k = 1\ (k = 1,\ldots,d)$,

- $\mathbf{v}_k^t \mathbf{v}_j = 0$ $(k \neq j)$.

Los vectores $\mathbf{v}_k$ se pueden obtener resolviendo

$$\mathbf{B}\mathbf{v} = \lambda \mathbf{v},$$

donde $\mathbf{B}$ denota la matriz de covarianzas de $\mathbf{X}$ y $\mathbf{v} \in \mathbb{R}^d$. El vector propio $\mathbf{v}_k$ es la k-ésima componente principal, asumiendo que los correspondientes valores propios satisfacen $\lambda_k \geq \lambda_{k+1}$ $(k = 1, \ldots, d-1)$.

Reducir la dimensión es especialmente importante cuando la dimensión de los datos es infinita, como es el caso de los datos funcionales. Nos centraremos sólo en curvas observadas en $[a, b]$ $(-\infty < a < b < \infty)$ y de cuadrado integrable. Entonces, si χ denota una variable aleatoria funcional, PCA se puede generalizar a FPCA. El objetivo es encontrar las funciones $\phi_k : [a, b] \to \mathbb{R}$ de modo que la varianza de

$$\beta_k = \int_a^b \phi_k(t)\chi(t)dt,$$

se maximice, sujeto a las condiciones:

- $\int_a^b \phi_k^2(t)dt = 1$,
- $\int_a^b \phi_k(t)\phi_j(t)dt = 0$ $(k \neq j)$.

Como en el caso de dimensión finita, las componentes principales ϕ_k se pueden definir como las funciones ortonormales verificando

$$\int_a^b \mathbf{C}(t, s)\phi_k(s)ds = \lambda_k \phi_k(t), \ (t \in [a, b], k = 1, 2, \ldots),$$

donde $\mathbf{C}(t, s)$ denota la covarianza entre $\chi(t)$ y $\chi(s)$. Finalmente, la dimensión de la reducción se calcula considerando la aproximación

$$\chi \approx \sum_{k=1}^{K} \beta_k \phi_k,$$

donde $K < \infty$ y $\sum_{k=1}^{K} \lambda_k$ es próximo a $\sum_{k=1}^{\infty} \lambda_k$ (se ha asumido que $\lambda_k \geq \lambda_{k+1}, k = 1, 2, \ldots$).

Las componentes principales, ϕ_k, dependen del operador de covarianza desconocido $\mathbf{C}$. Asumiendo que se tienen las observaciones $\{\chi_i\}_{i=1}^n$ idénticamente distribuidas de una variable aleatoria funcional χ, las estimaciones para ϕ_k se pueden obtener usando

$$\widehat{\mathbf{C}}(t, s) = \frac{1}{n} \sum_{i=1}^{n} (\chi_i(t) - \overline{\chi}(t))(\chi_i(s) - \overline{\chi}(s)),$$

donde

$$\overline{\chi}(t) = \frac{1}{n}\sum_{i=1}^{n}\chi_i(t).$$

Además de reducir la dimensión, FPCA se puede utilizar para detectar *outliers.* La estimación de las componentes principales funcionales está basada en el operador de covarianza $\widehat{\mathbf{C}}$ que es sensible a *outliers.*

Hyndman y Ullah proponen estimar las componentes principales por medio de las funciones $\widehat{\phi}_k$ que maximizan la varianza de

$$z_{ik} = w_i \int_a^b \phi_k(t)\chi_i(t)dt,$$

sujeto a las mismas restricciones anteriores de ortogonalidad. Los pesos w_i se calculan como

$$w_i = \begin{cases} 1 \text{ si } v_i < S + \lambda\sqrt{S}, \\ 0 \text{ en otro caso.} \end{cases}$$

donde

$$v_i = \int_a^b (\chi_i(t) - \sum_{k=1}^{K} \tilde{\beta_{ik}}\tilde{\phi}_k)(t))^2 dt,$$

con $\tilde{\phi}_k$ estimadores de $\phi_k(t)$ obtenidos mediante el algoritmo RAPCA (ver M. Hubert, P.J, Rousseeuw y S. Verboven (2002) [25]) siendo S la mediana de $v_1, \ldots, v_n$ y $\lambda > 0$ un parámetro de control del grado de robustez. Una vez obtenidos los estimadores $\widehat{\phi}_k(t)$, los coeficientes correspondientes a la curva $\chi_i(t)$ se construyen como

$$\widehat{\beta}_{ik} = \int_a^b \widehat{\phi}_k(t)\chi_i(t)dt.$$

El método propuesto por Hyndman y Ullah para detectar *outliers* se conoce como método ISE y consiste en comprobar para cada χ_i si $w_i = 0$. En ese caso, la curva χ_i es considerada un *outlier.*

4.3.2. Médologías *fbplot*

Y. Sun y M. Genton (2011) [26] extendieron la construcción de los gráficos boxplot al caso funcional, a través de la generalización de estimadores robustos. Para ordenar la muestra, se basaron en una medida de profundidad propuesta por S. López-Pintado y J. Romo (2009) [27].

Profundidad de banda para datos funcionales

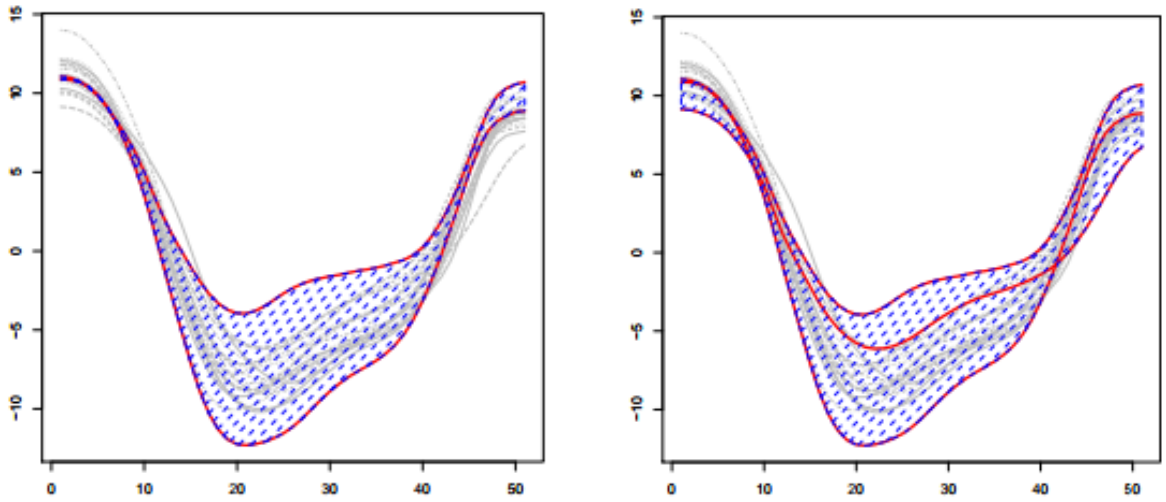
La noción de profundidad introducida en S. López-Pintado y J. Romo (2009), se basa en la representación gráfica de los datos y de las bandas que determinan el plano. El gráfico de una

función x es el subconjunto del plano $G(x) = \{(t, x(t)) : t \in I\}$. Dadas las curvas $x_1(t), \ldots, x_n(t)$, la banda en $\mathbb{R}^2$ determinada por k de tales curvas $x_{i1}, \ldots, x_{ik}$ es

$$V(x_{i1}, x_{i2}, \ldots, x_{ik}) = \{(t, y) : t \in I, \min_{r=1,\ldots,k} x_{ir}(t) \leq y \leq \max_{r=1,\ldots,k} x_{ir}(t)\} =$$

$$= \{(t, y) : t \in I, y = \alpha_t \min_{r=1,\ldots,k} x_{ir}(t) + (1 - \alpha_t) \max_{r=1,\ldots,k} x_{ir}(t), \alpha_t \in [0, 1]\}.$$

La Figura 4.5a muestra en rojo con sombreado azul la banda $V(x_1, x_2)$ delimitada por dos curvas; el gráfico de las funciones x_i se incluye representado en color gris. La Figura 4.5b presenta la banda dada por tres curvas $V(x_1, x_2, x_3)$.



(a) Banda formada por dos curvas

(b) Banda formada por tres curvas

La profundidad de una curva en la muestra se mide en términos de cuántas bandas la contienen. Para cualquier función $x \in \{x_1, \ldots, x_n\}$, calculamos para $j \geq 2$

$$BD_n^{(j)} = \binom{n}{j}^{-1} \sum_{1 \leq i_1 \leq i_2 \leq \cdots \leq i_j \leq n} \mathbb{I}\{G(x) \subset V(x_{i1}, x_{i2}, \ldots, x_{ij})\},$$

donde $\mathbb{I}(A)$ es el indicador de la condición A, es decir, $\mathbb{I}(A)$ vale 1 si se verifica A y 0 si no.

El numerador de esta expresión cuenta la cantidad de bandas $V(x_{i1}, \ldots, x_{ij})$ determinadas por j curvas diferentes $x_{i1}, \ldots, x_{ij}$ que contienen el gráfico de x. El denominador representa la cantidad de bandas que pueden formarse con j curvas elegidas de las n que conforman la muestra. Luego $BD_n^{(j)}(x)$ es la proporción de bandas formadas por j curvas que contienen el gráfico de la curva x.

Dadas las funciones $x_1, \ldots, x_n$, se define la profundidad de banda de una función x como

$$BD_{n,J}(x) = \sum_{j=2}^{J} BD_n^{(j)}(x), \quad J \geq 2.$$

Con esta noción de profundidad, dada una muestra, definimos su mediana muestral como la observación $\widehat{m}_{n,J}$ con mayor valor de profundidad

$$\widehat{m}_{n,J} = \underset{x \in \{x_1, \ldots, x_n\}}{\arg\max} \; BD_{n,J}(x).$$

En principio, la definición propuesta para la profundidad de banda depende del parámetro J, es decir de la cantidad máxima curvas que generan una banda.

Profundidad de banda generalizada

Cuando las curvas son muy irregulares, pocas bandas contendrán por completo una curva. Varias curvas de la muestra tendrán el mismo valor de profundidad, lo cual resulta en un ordenamiento pobre de la muestra con muchos empates. Por ejemplo, la banda definida por dos curvas ($J = 2$) que se corten en un punto, no contendrá ninguna otra curva y no contribuirá al valor de la profundidad.

En este sentido, la definición anterior de profundidad de banda es restrictiva. Esto proviene de usar la función $\mathbb{I}$ que toma sólo los valores 0 ó 1. Una definición alternativa, más flexible, es medir el conjunto donde la función queda contenida en la correspondiente banda. Para cualquier función x en $x_1, \ldots, x_n$, sea para $j \geq 2$

$$A_{i1,\ldots,ij}(x) = \{t \in I : \min_{r=i_1,\ldots,i_j} x_r(t) \leq x(t) \leq \max_{r=i_1,\ldots,i_j} x_r(t)\},$$

el conjunto de puntos del intervalo I donde la función x está dentro de la banda determinada por las observaciones $x_{i1}, \ldots, x_{ij}$. Si λ es la medida de Lebesque en I,

$$\lambda_r(A_j(x)) = \frac{\lambda(A_j(x))}{\lambda(I)}$$

es la 'proporción de tiempo' que x está en la banda. La medida de Lebesgue es la forma estándar de asignar una longitud, área o volumen a los subconjuntos de un espacio euclídeo. Los conjuntos a los que se les puede asignar un tamaño se denominan Lebesgue-medibles. El volumen o medida de un conjunto Lebesgue-medible A se denota por $\lambda(A)$. Si A es un intervalo cerrado $[a, b]$, su medida de Lebesgue es la longitud $b - a$.

Se define entonces,

$$MDB_n^{(j)}(x) = \binom{n}{j}^{-1} \sum_{1 \leq i_1 \leq i_2 \leq \cdots \leq i_j \leq n} \lambda_r(A(x; x_{i1}, x_{i2}, \ldots, x_{ij})), j \geq 2.$$

Es una generalización de $BD_n^{(j)}(x)$ ya que si x está siempre dentro de la banda $\lambda_r(A_j(x)) = 1$ y extiende la definición anterior.

Dado $J \geq 2$, para funciones $x_1, \ldots, x_n$, la profundidad de banda generalizada de una curva x es

$$MBD_{n,J}(x) = \sum_{j=2}^{J} MBD_n^{(j)}(x).$$

Como la profundidad de banda generalizada tiene en cuenta la proporción de tiempo que una curva permanece dentro de una banda, es más conveniente para obtener la curva más representativa en términos de magnitud, a diferencia de la profundidad de banda que depende más de la forma de las curvas, luego puede usarse para obtener la curva más representativa en términos de forma.

La Figura 4.5 muestra un ejemplo sencillo con $n = 4$ curvas sobre cómo se calculan BD y MBD. Para $J = 2$, hay 6 posibles bandas limitadas por 2 curvas. Por ejemplo, el área pintada en la Figura 4.5 es la banda limitada por $y_1(t)$ y $y_3(t)$. Se ve que la curva $y_2(t)$ pertenece totalmente a la banda, mientras que la curva $y_4(t)$ lo hace parcialmente. Se considera que si una curva toca el borde de la banda, pertenece a ella. Así pues, $BD(y_2) = 5/6 = 0.83$ porque sólo la banda delimitada por $y_3(t)$ y $y_4(t)$ no contiene completamente a la curva $y_2(t)$ y $BD(y_4) = 3/6 = 0.5$ dado que es la única completamente contenida en las bandas limitadas por sí misma y otra curva. De modo similar, se puede calcular $BD(y_1) = 0.5$ y $BD(y_3) = 0.5$. Para calcular MBD, la curva $y_2(t)$ está siempre contenida en las cinco bandas, entonces $MBD(y_2) = 0.83$ (el mismo valor que BD). En contraste, la curva y_4 sólo pertenece a la banda pintada el 40 % del tiempo, entonces, $MBD(y_4) = (3 + 0.4 + 0.4)/6 = 0.63$ por definición. Para las otras dos curvas $MBD(y_1) = 0.5$ y $MBD(y_3) = 0.7$.

Una diferencia entre ambas profundidades de banda es su comportamiento ante curvas que abandonan el centro de la muestra en un intervalo corto, es decir, permanecen en el interior de la muestra casi todo el tiempo pero toman valores extremos en subintervalos pequeños. En ese caso, el valor de MBD sigue siendo grande mientras que BD es sensible y pasa a tomar valores pequeños. Es decir que BD es resistente a los *outliers* de forma mientras que la profundidad MBD es robusta cuando los *outliers* son en magnitud.

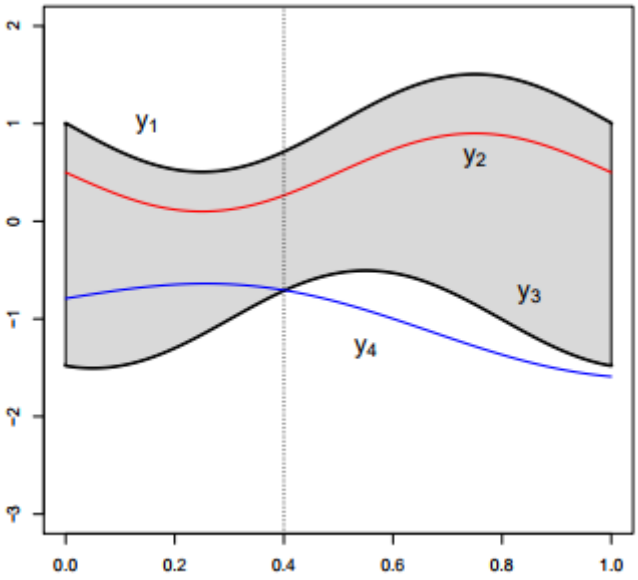


Figura 4.5: Ejemplo del cálculo de BD y MBD: la región pintada es la banda delimitada por las curvas $y_1(t)$ y $y_3(t)$. La curva $y_2(t)$ pertenece completamente a la banda mientras que $y_4(t)$ lo hace parcialmente.

La profundidad de banda y su versión numérica permiten ordenar muestras y extender los métodos univariados, basados en los estadísticos de orden, al caso funcional. A continuación, se extiende la construcción del boxplot.

Boxplot funcional

Suponemos que cada observación es una función real $x_i(t), i = 1, \ldots, n, t \in I$, donde I es un intervalo en $\mathbb{R}$. Sea $x_{(i)}(t)$ la observación asociada al i-ésimo mayor valor de profundidad. Luego $x_{(1)}(t), \ldots, x_{(n)}(t)$ son los estadísticos de orden siendo $x_{(1)}(t)$ la curva más profunda (más central) y $x_{(n)}(t)$ la más exterior. Los estadísticos de orden inducidos por la medida de profundidad empiezan en la curva muestral más central y se alejan hacia afuera en todas las direcciones.

En el boxplot clásico, la caja representa el 50 % de los datos. En el boxplot funcional se busca el 50 % de las observaciones más profundas

$$C_{0,5} = \{(t, x(t)) : \min_{r=1,\ldots,\lceil \frac{n}{2} \rceil} x_{[r]}(t) \leq x(t) \leq \max_{r=1,\ldots,\lceil \frac{n}{2} \rceil} x_{[r]}(t)\},$$

donde $\lceil \frac{n}{2} \rceil$ es el menor entero mayor a $\frac{n}{2}$.

Esta región central da una medida robusta de dispersión del 50 % de las curvas más centrales. La curva $x_{(1)}(t)$ es la que indica la mediana, o sea, la curva más central, la de mayor profundidad. La mediana funcional también es un estadístico robusto para medir centralidad. La idea de regiones centrales puede ampliarse para definir las regiones del 25 y 75 %.

Los bigotes del gráfico boxplot son las líneas que se extienden desde la caja hasta la última observación que no es atípica. Se necesita entonces una regla de detección de *outliers*. De nuevo, se extiende la regla del boxplot clásico, inflar la región central 1.5 veces. Por tanto, se define la *región exterior* inflando la región central 1.5 veces su tamaño. Cualquier curva fuera de estos límites se clasifica como un potencial *outlier*.

Es inmediato ver que si las funciones fueran constantes el boxplot funcional se reduce al boxplot univariado.

En la Figura 4.6 se ilustra la construcción de un boxplot funcional sobre una muestra que representa las alturas de una población de mujeres. En negro se dibuja la mediana, la zona pintada de color rosa es la región central que contiene al 50 % de las curvas y el dato atípico se representa en color rojo.

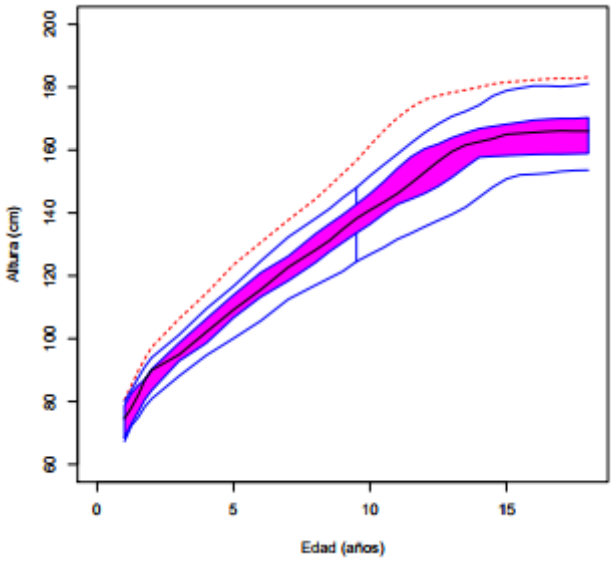


Figura 4.6: Boxplot funcional de las alturas para una población de mujeres, con un *outlier* en color rojo.

4.3.3. Análisis de arquetipos

El análisis de arquetipos, en inglés, *archetype analysis* (AA) es una técnica estadística que busca aproximar datos por una combinación convexa de elementos extremos denominados arquetipos. Los arquetipos se construyen como una combinación convexa de las observaciones. AA fue introducido por A. Cutler y L. Breiman (1994) [28]. Recientemente, G. Vinué, I. Epifanio, S. Alemany (2015) [29] han introducido el *archetypoid analysis* (ADA), de modo que se diferencia del análisis AA ya conocido en que los elementos extremos no son una combinación convexa de las observaciones sino que son observaciones reales de la muestra. En esta sección se definen ambos análisis para muestras multivariantes y se generaliza posteriormente esta definición para

datos funcionales, siguiendo I. Epifanio (2016) [30].

Definición de AA y ADA para muestras multivariantes

Sea $\mathbf{X}$ una matriz $n \times p$ que contiene los valores observados de p variables para n individuos. El objetivo de AA es encontrar una matriz $\mathbf{Z}$ de dimensión $k \times p$ cuyas filas son los k arquetipos de esos datos, de forma que los elementos de la muestra se puedan aproximar como una combinación convexa, de los arquetipos.

Para la obtención de los arquetipos, AA calcula dos matrices α y β que minimizan la suma de cuadrados de los residuos (RSS) que se obtiene de la combinación de la ecuación donde cada elemento $\mathbf{x}_i, i = 1, \ldots, n$ se aproxima como una mixtura de los arquetipos $\mathbf{z}_j, j = 1, \ldots, k$:

$$\sum_{i=1}^{n} ||\mathbf{x}_i - \sum_{j=1}^{k} \alpha_{ij} \mathbf{z}_j||^2,$$

y la ecuación donde los arquetipos $\mathbf{z}_j, j = 1, \ldots, k$ se expresan como una mixtura de los datos:

$$\mathbf{z}_j = \sum_{l=1}^{n} \beta_{jl} \mathbf{x}_l.$$

Por tanto,

$$RSS = \sum_{i=1}^{n} ||\mathbf{x}_i - \sum_{j=1}^{k} \alpha_{ij} \mathbf{z}_j||^2 = \sum_{i=1}^{n} ||\mathbf{x}_i - \sum_{j=1}^{k} \alpha_{ij} \sum_{l=1}^{n} \beta_{jl} \mathbf{x}_l||^2$$

bajo las restricciones:

1) $\sum_{j=1}^{k} \alpha_{ij} = 1$ con $\alpha_{ij} \geq 0$ para $i = 1, \ldots, n$ y

2) $\sum_{l=1}^{n} \beta_{jl} = 1$ con $\beta_{jl} \geq 0$ para $j = 1, \ldots, k$.

La restricción 1), quiere decir que las aproximaciones de los elementos $\mathbf{x}_i$ son una combinación convexa de los arquetipos,

$$\widehat{\mathbf{x}}_i = \sum_{j=1}^{k} \alpha_{ij} \mathbf{z}_j.$$

Cada α_{ij} es el peso del arquetipo j para la observación i, es decir, los coeficientes de la matriz α indican cuánto contribuye cada arquetipo a la estimación de cada observación. La restricción 2), indica que los arquetipos $\mathbf{z}_j$ son una mixtura de las observaciones,

$$\mathbf{z}_j = \sum_{l=1}^{n} \beta_{jl} \mathbf{x}_l.$$

Notemos que los arquetipos no son necesariamente observaciones de la muestra. Esto sólo ocurre si sólo un β_{jl} es igual a 1 en la restricción 2) para cada j. Esto implica que β_{jl} sólo puede tomar los valores 0 ó 1, ya que $\beta_{jl} \geq 0$ y la suma de la restricción 2) es 1. En ADA, los arquetipos son observaciones de la muestra, luego, la segunda restricción se transforma en

$$2) \ \sum_{j=1}^{n} \beta_{jl} = 1 \text{ con } \beta_{jl} \in \{0, 1\} \text{ para } j = 1, \ldots, k.$$

Notemos que esta restricción implica que $\beta_{jl} = 1$ para un único valor de l y $\beta_{jl} = 0$ para el resto.

Generalización para datos funcionales

En el contexto de datos funcionales, el análisis de arquetipos se puede generalizar fácilmente. Ahora, los valores de las p variables en el caso multivariante se reemplazan por los valores de una función y los sumatorios se reemplazan por integrales. Además, las normas vectoriales se sustituyen por normas funcionales y los vectores $\mathbf{x}_i$ y $\mathbf{z}_j$ se corresponden con funciones, x_i y z_j. El objetivo es encontrar k funciones arquetipo, de modo que los datos funcionales de la muestra se puedan aproximar por combinaciones convexas de esos arquetipos.

El análisis de arquetipos puede ayudar a detectar *outliers* de datos funcionales, pues cuánto mayor sea el número de arquetipos, k, se puede llegar a obtener algún *outlier* como arquetipo, de modo que tenga valores de α elevados en muy pocos individuos, incluso puede que sólo él mismo.

Capítulo 5

Resultados obtenidos

En este capítulo se describe cómo se lleva a cabo el proceso de detección de *outliers* funcionales y cuáles son los resultados obtenidos respecto de los caudales del consumo hídrico en tres sectores de la provincia de Castellón.

En primer lugar, se debe realizar el suavizado, ajustando los datos a una base, con el objetivo de obtener la función que da lugar a las observaciones discretas de la muestra y de eliminar el ruido producido por los sistemas de medición. En segundo lugar, una vez se han ajustado los datos a una base, se aplican distintas metodologías de detección de *outliers* explicadas en el Capítulo 4 y se comparan los resultados obtenidos con el objetivo de determinar qué método es más adecuado y proporciona mejores resultados.

Para cada sector se organiza la muestra de modo que cada función se corresponde con los valores de caudal registrados durante una noche (de 00:00 a 06:00h), es decir, que cada fila de la muestra contiene los datos de una noche.

5.1. Suavizado

El primer paso cuando se trabaja con datos funcionales, consiste en transformar los datos discretos a una función eliminando también el ruido producido por los sistemas de medición. El sistema de bases de funciones elegido para realizar el ajuste son las bases B-spline. Se ha decidido utilizar este tipo de bases debido a que las bases más utilizadas suelen ser las bases de Fourier y las bases B-spline pero, en este caso, las bases de Fourier no son adecuadas porque el tipo de datos no es periódico, ya que se trabaja únicamente con el horario nocturno. Además, las bases B-spline presentan un coste computacional muy bajo que las hace adecuadas para trabajar con grandes cantidades de datos. El orden escogido para las bases B-spline es orden 4 (B-splines cúbicos) porque, en este caso, no se van a utilizar derivadas y se reduce de este modo el coste computacional, evitando utilizar órdenes más altos. La posición de los nodos elegida son nodos equiespaciados. Para la elección del número de funciones base, K, se ha seguido la

estrategia de probar con distinto número de bases (de 4 a 22) y calcular para cada una de ellas la media de la varianza de los residuos de todas las funciones de la muestra mediante la ecuación

$$\overline{s^2} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{p_i - K}\sum_{j=1}^{p_i}(y_{ij} - \widehat{y}_{ij})^2\right),$$

variando K de 4 a 22. Las Figuras 5.1, 5.2 y 5.3 representan la media de la varianza de los residuos de todas las funciones de la muestra dependiendo del número de bases elegido para los sectores A, B y C respectivamente.

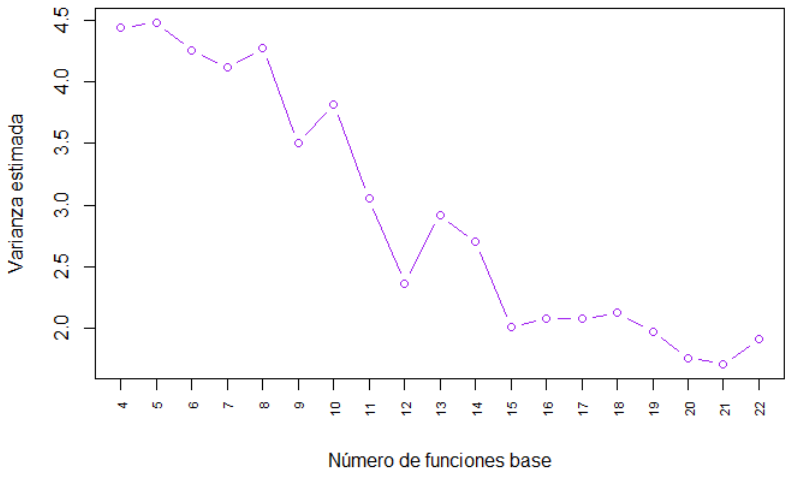


Figura 5.1: Varianza media de los residuos del sector A, en función del número de bases B-spline.

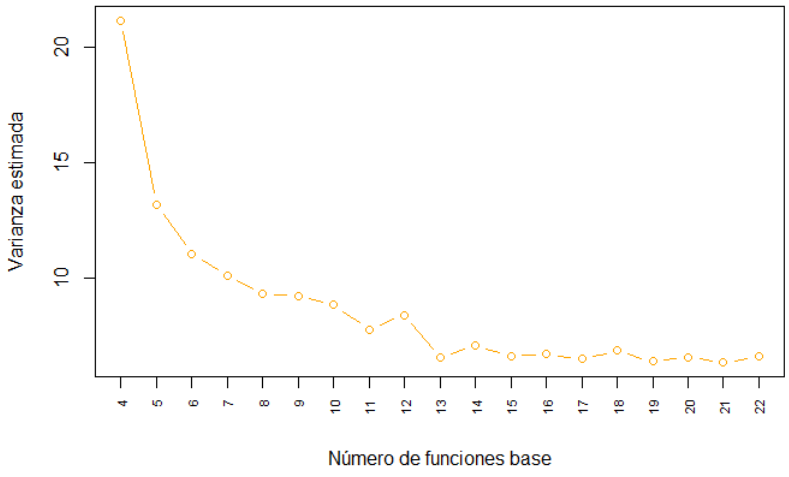


Figura 5.2: Varianza media de los residuos del sector B, en función del número de bases B-spline.

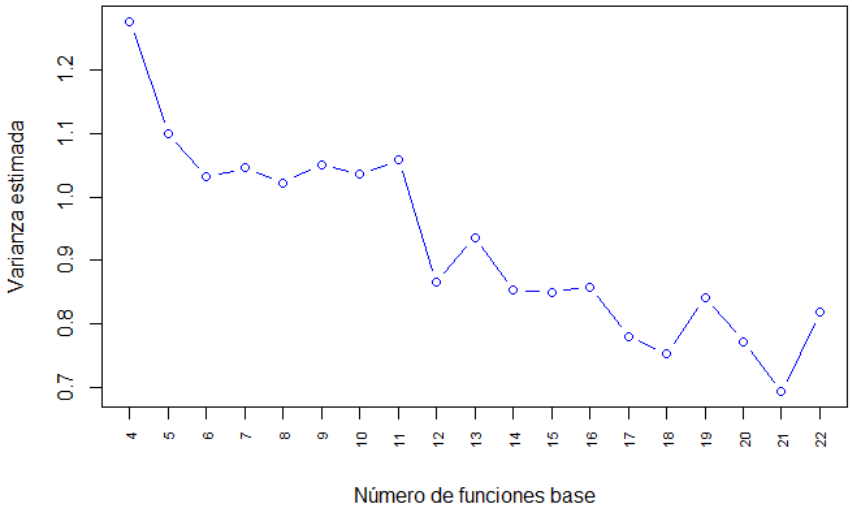


Figura 5.3: Varianza media de los residuos del sector C, en función del número de bases B-spline.

Observando los gráficos realizados para los tres sectores, para elegir el número de funciones base, se debe mirar hasta que número de funciones base la varianza media de los residuos va descendiendo rápidamente. Para el sector A se escogen 12 funciones base, para el sector B, 13 funciones base y para el sector C, 12 funciones base.

Para realizar el suavizado con R se utiliza el paquete *fda*. En primer lugar, se debe crear el sistema de bases seleccionado con el número de funciones base elegido. En este caso, como se ha decidido utilizar bases B-spline, la instrucción a utilizar es *create.bspline.basis* y, para ajustar los datos a esta base creada, se debe utilizar la instrucción *Data2fd*. Finalmente, hay que usar la sentencia *eval.fd* para obtener los valores de la función ajustada en los instantes de tiempo deseados.

En la Figura 5.4, se representa el ajuste de los datos del día 01/01/2014 del sector A al sistema de bases B-spline de orden 4 con un número de funciones base igual a 12. La nube de puntos se corresponde con los valores de caudal registrados cada 5 minutos y la línea de color rojo con la función suavizada estimada.

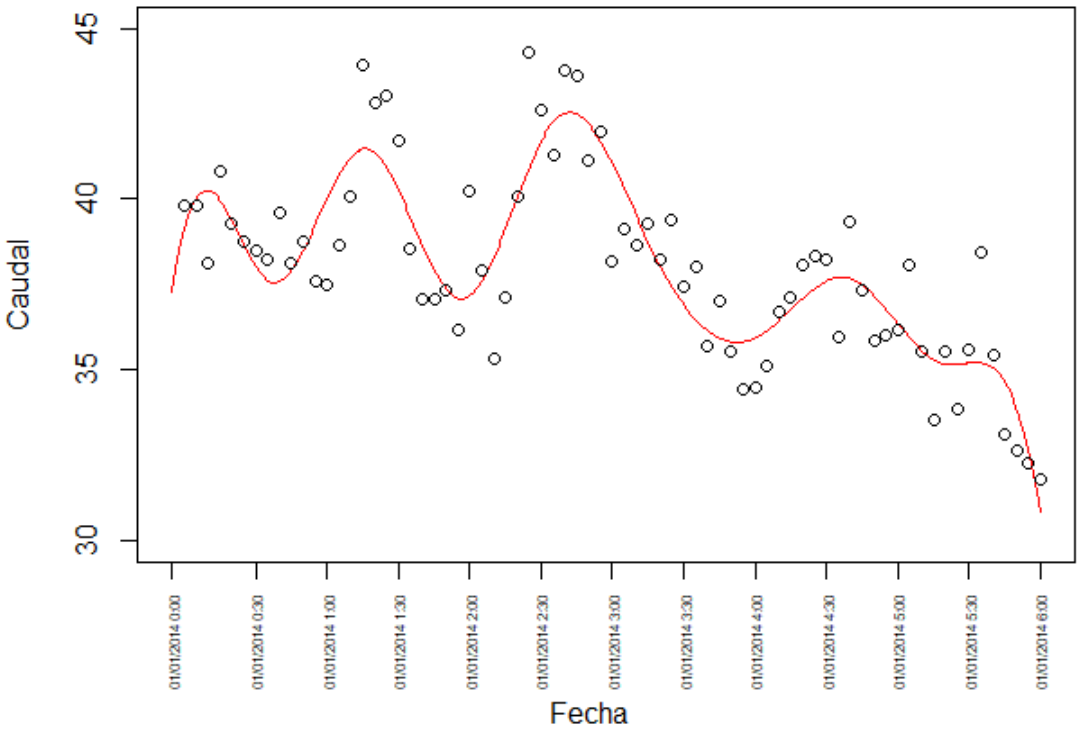


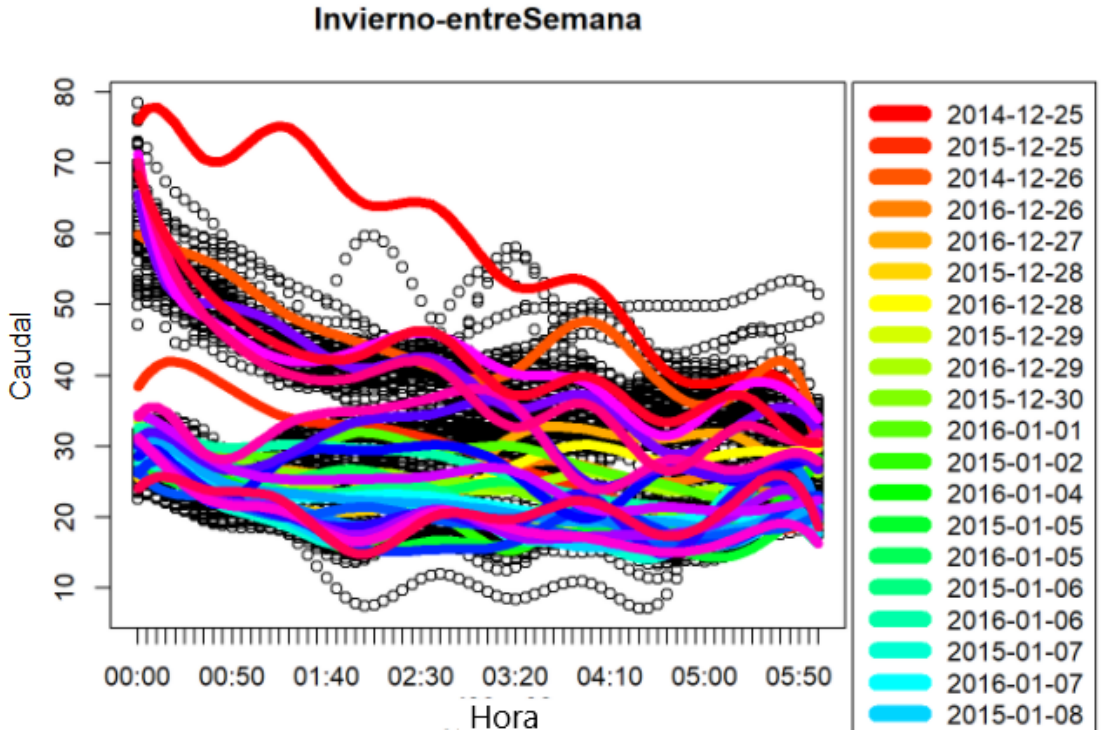
Figura 5.4: Suavizado de los datos del día 01/01/2014 para el sector A.

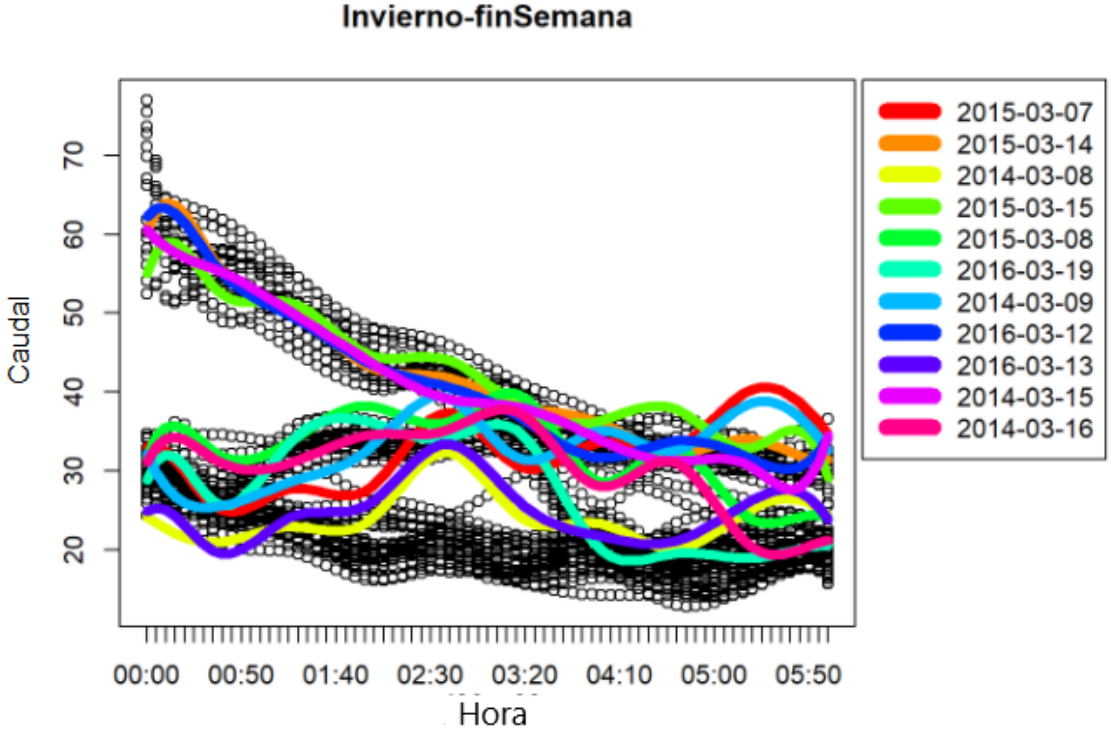
5.2. Búsqueda de *outliers* funcionales

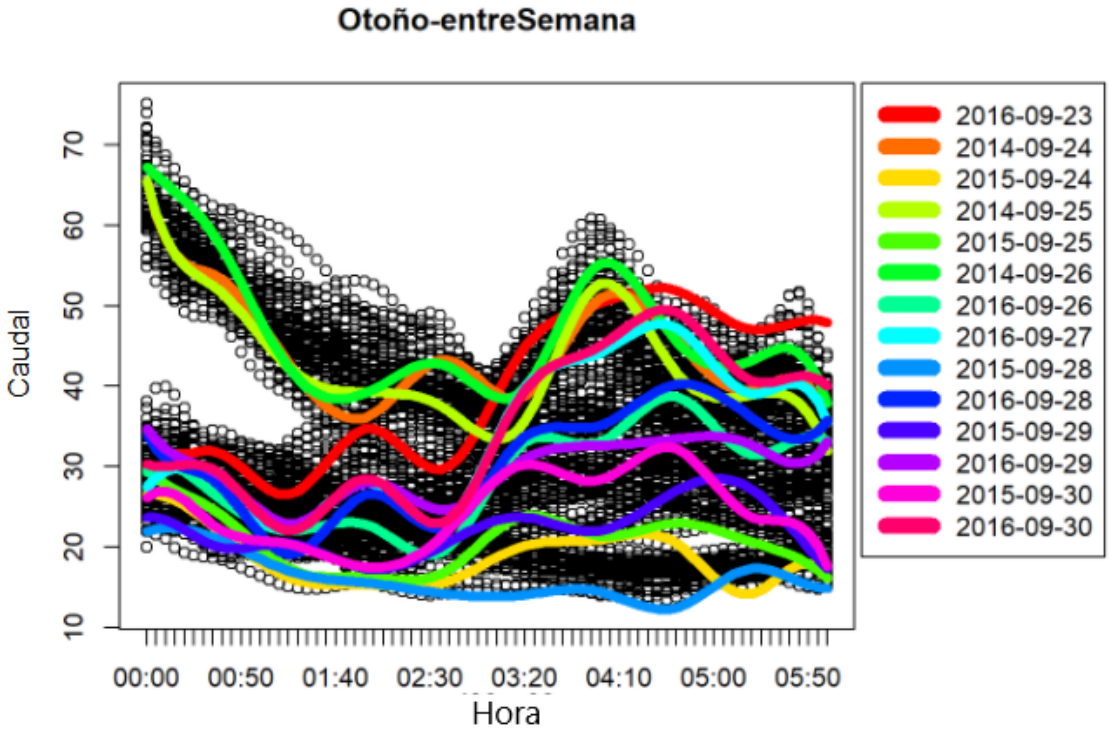
5.2.1. Sector A

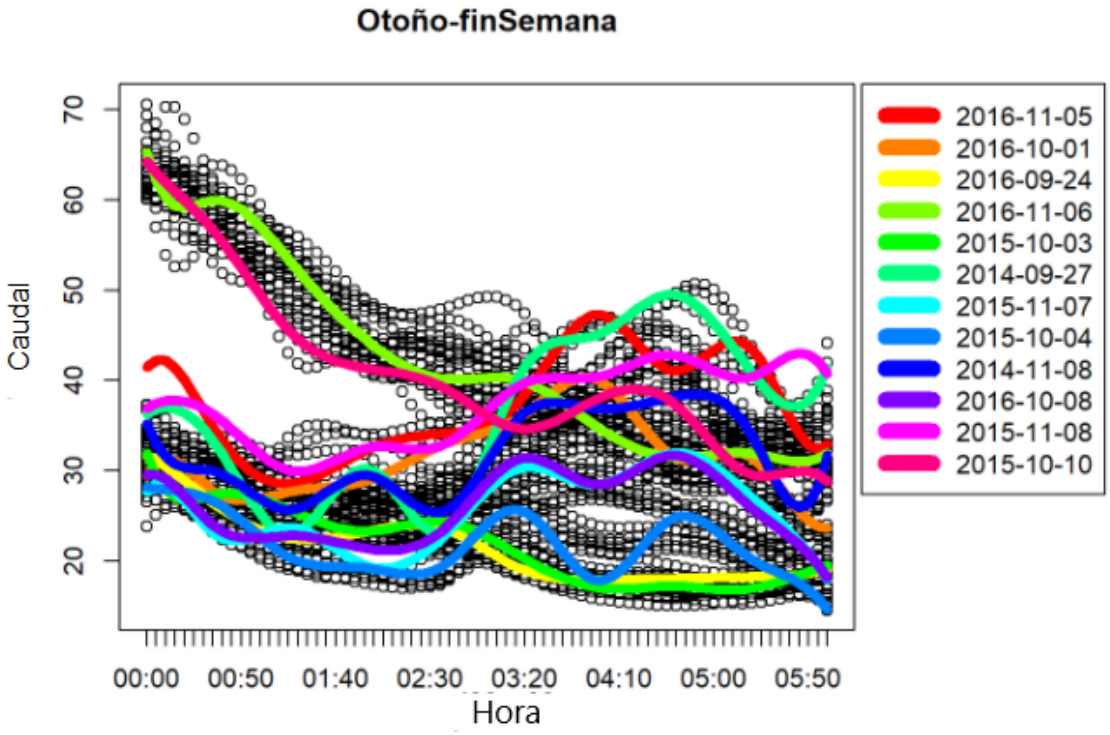
En esta sección se muestran los resultados de aplicar las distintas metodologías de detección de *outliers* al sector A.

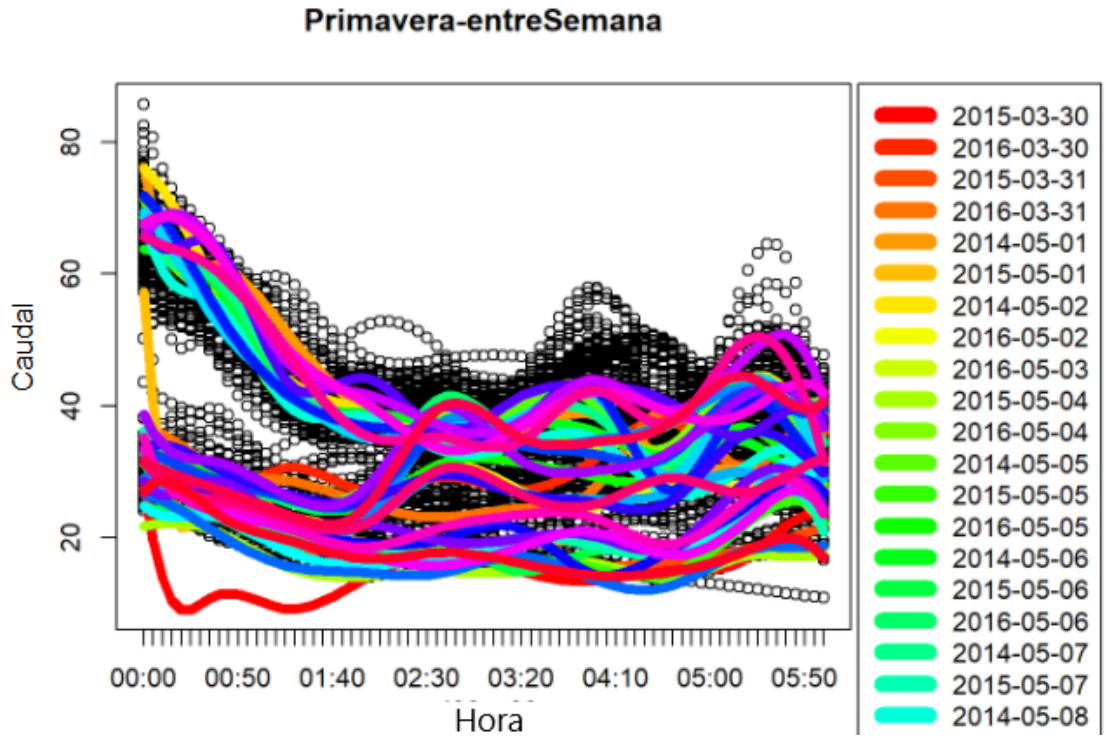
Los resultados obtenidos al aplicar los métodos 'depth.trim' y 'depth.pond' son idénticos para cada grupo y se representan en la Figura 5.5. La nube de puntos de color negro se corresponde con las funciones de caudal no detectadas como *outliers*. Se utilizan líneas gruesas de colores para marcar los *outliers* encontrados. Con esta representación se puede observar cómo son los *outliers* detectados en comparación con el resto de funciones.

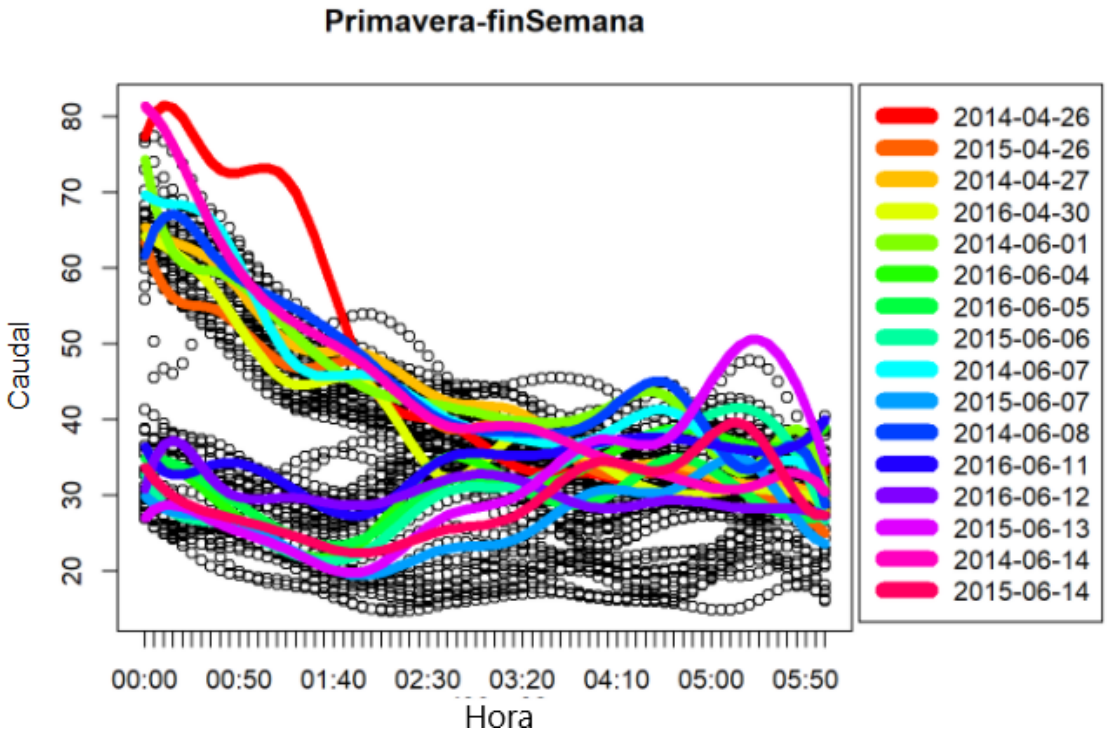

Invierno-entreSemana
Caudal
10
20
30
40
50
60
70
80
00:00
00:50
01:40
02:30
03:20
04:10
05:00
05:50
Hora
2014-12-25
2015-12-25
2014-12-26
2016-12-26
2016-12-27
2015-12-28
2016-12-28
2015-12-29
2016-12-29
2015-12-30
2016-01-01
2015-01-02
2016-01-04
2015-01-05
2016-01-05
2015-01-06
2016-01-06
2015-01-07
2016-01-07
2015-01-08


Invierno-finSemana
Caudal
20
30
40
50
60
70
00:00
00:50
01:40
02:30
03:20
04:10
05:00
05:50
Hora
2015-03-07
2015-03-14
2014-03-08
2015-03-15
2015-03-08
2016-03-19
2014-03-09
2016-03-12
2016-03-13
2014-03-15
2014-03-16


Otoño-entreSemana
Caudal
10
20
30
40
50
60
70
00:00
00:50
01:40
02:30
03:20
04:10
05:00
05:50
Hora
2016-09-23
2014-09-24
2015-09-24
2014-09-25
2015-09-25
2014-09-26
2016-09-26
2016-09-27
2015-09-28
2016-09-28
2015-09-29
2016-09-29
2015-09-30
2016-09-30


Otoño-finSemana
Caudal
20
30
40
50
60
70
00:00
00:50
01:40
02:30
03:20
04:10
05:00
05:50
Hora
2016-11-05
2016-10-01
2016-09-24
2016-11-06
2015-10-03
2014-09-27
2015-11-07
2015-10-04
2014-11-08
2016-10-08
2015-11-08
2015-10-10


Primavera-entreSemana
Caudal
20
40
60
80
00:00
00:50
01:40
02:30
03:20
04:10
05:00
05:50
Hora
2015-03-30
2016-03-30
2015-03-31
2016-03-31
2014-05-01
2015-05-01
2014-05-02
2016-05-02
2016-05-03
2015-05-04
2016-05-04
2014-05-05
2015-05-05
2016-05-05
2014-05-06
2015-05-06
2016-05-06
2014-05-07
2015-05-07
2014-05-08


Primavera-finSemana
Caudal
20
30
40
50
60
70
80
00:00
00:50
01:40
02:30
03:20
04:10
05:00
05:50
Hora
2014-04-26
2015-04-26
2014-04-27
2016-04-30
2014-06-01
2016-06-04
2016-06-05
2015-06-06
2014-06-07
2015-06-07
2014-06-08
2016-06-11
2016-06-12
2015-06-13
2014-06-14
2015-06-14

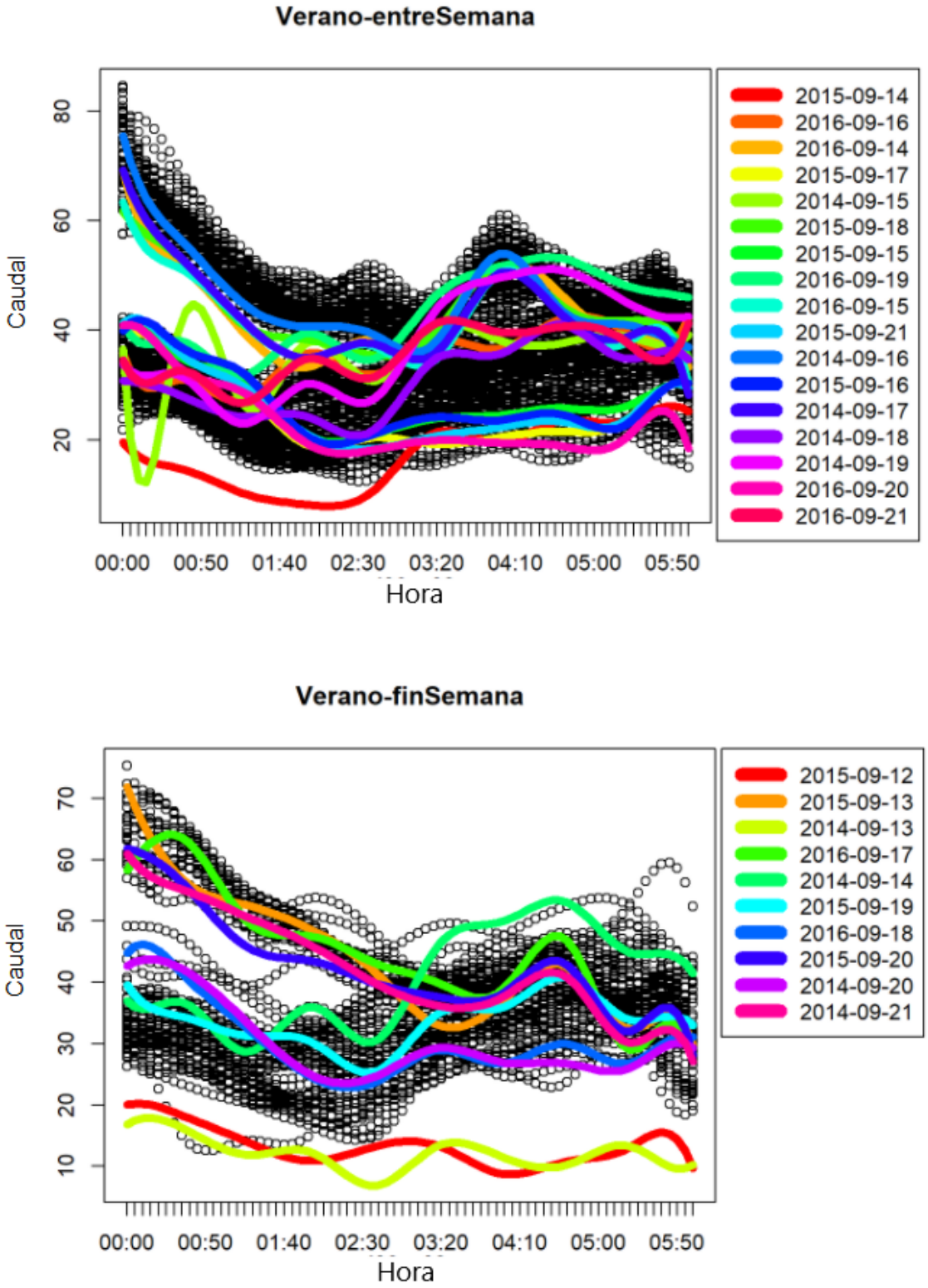


Figura 5.5: *Outliers* funcionales para el sector A con los métodos 'depth.trim' y 'depth.pond'.

Utilizando los métodos 'lrt' y 'HUoutliers' no se obtienen *outliers* en ninguno de los grupos del sector A. Tampoco se obtienen utilizando la instrucción *fbplot*.

Respecto del análisis de arquetipos cabe decir que variando el número de los mismos de 3 a 5, no hay ningún arquetipo que tenga coeficientes muy altos en muy pocos individuos. Por tanto, tampoco se obtienen *outliers* en este caso.

Conclusiones para el sector A

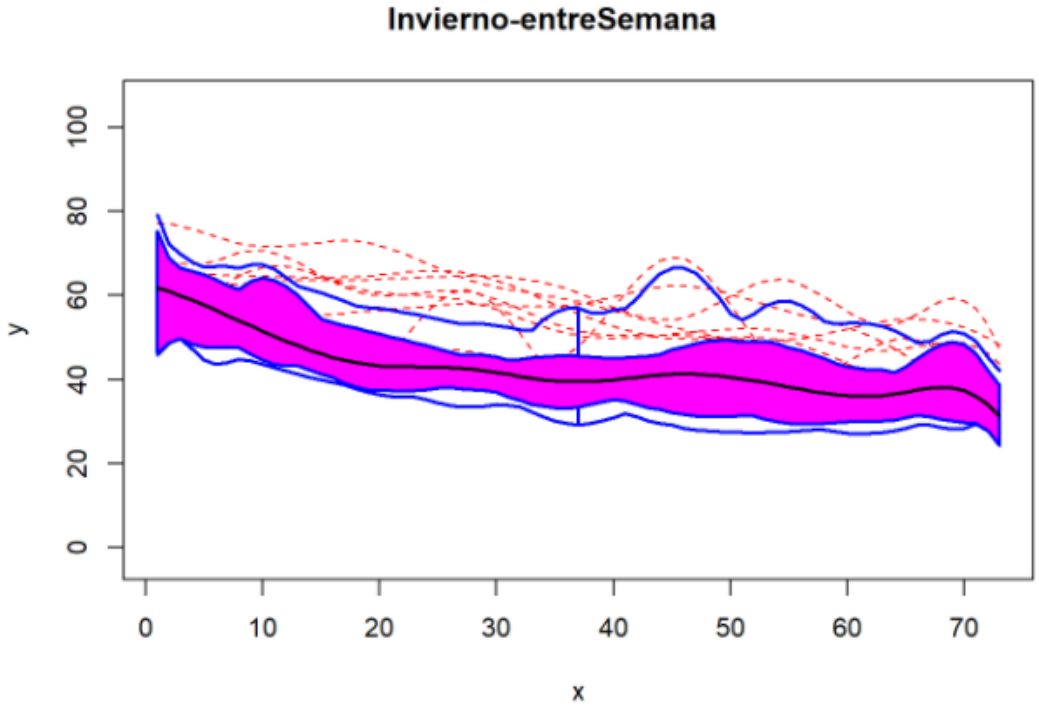
Observando los resultados obtenidos para el sector A, con las distintas metodologías incluidas en los paquetes de R, vemos que los métodos 'depth.trim' y 'depth.pond' de la instrucción *foutliers* son métodos muy sensibles que detectan bastantes *outliers* en la forma de las funciones. Sin embargo, se han obtenido muy pocos *outliers* en la magnitud de los datos que se puedan corresponder con fugas en la red de distribución de agua. El único *outlier* que se puede considerar en la magnitud de los datos se obtiene en invierno un día entre semana, pero se corresponde con el día 25/12/2014, que es un día festivo, y por tanto, se entiende que el consumo de agua pueda ser más elevado durante esa noche. Sin embargo, con el resto de métodos no se obtienen *outliers* para este sector, por tanto, atendiendo a los resultados extraídos, se puede concluir que durante los tres últimos años, en el sector A no ha habido fugas en la red de distribución de agua.

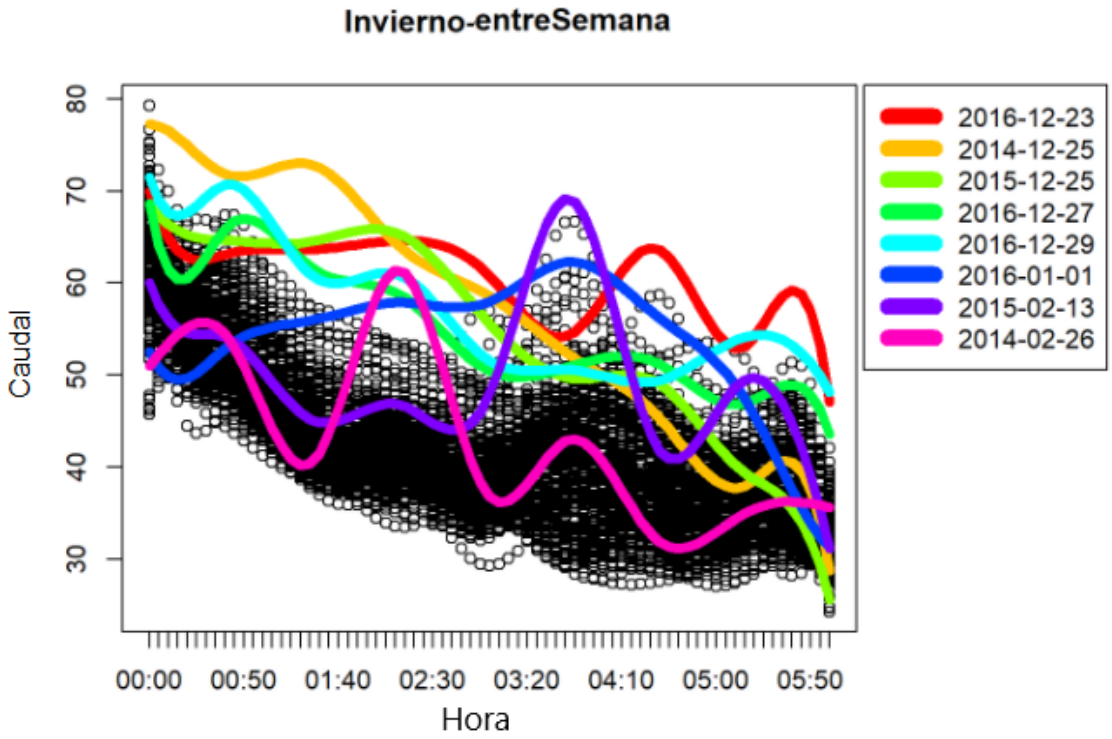
5.2.2. Sector B

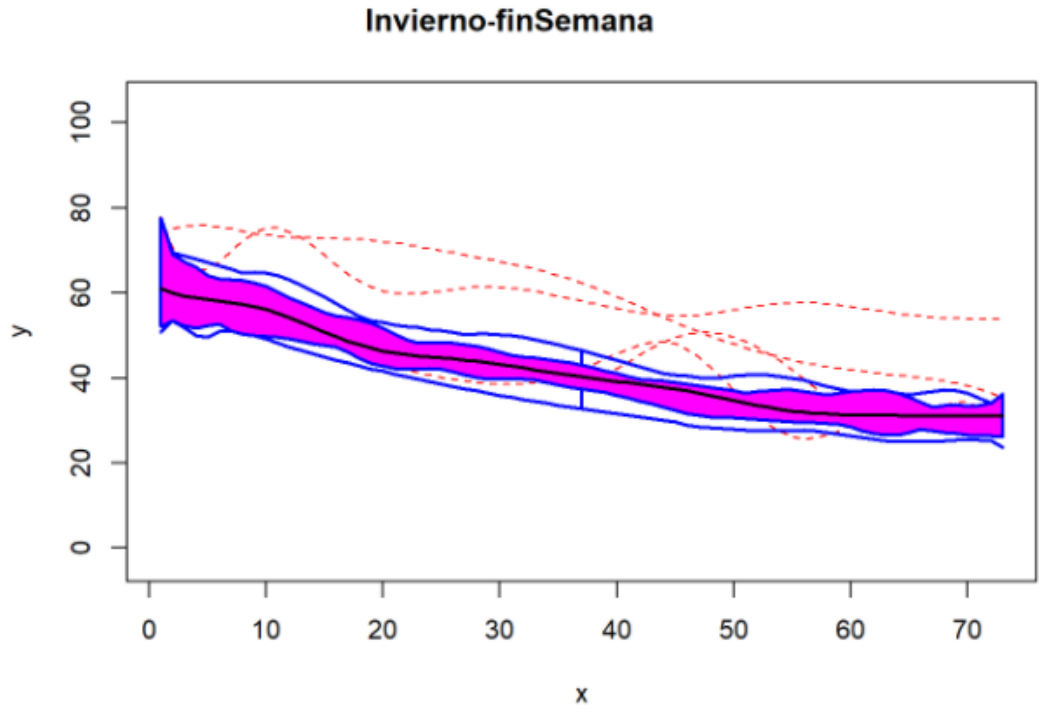
En esta sección se muestran los resultados de aplicar las distintas metodologías de detección de *outliers* al sector B.

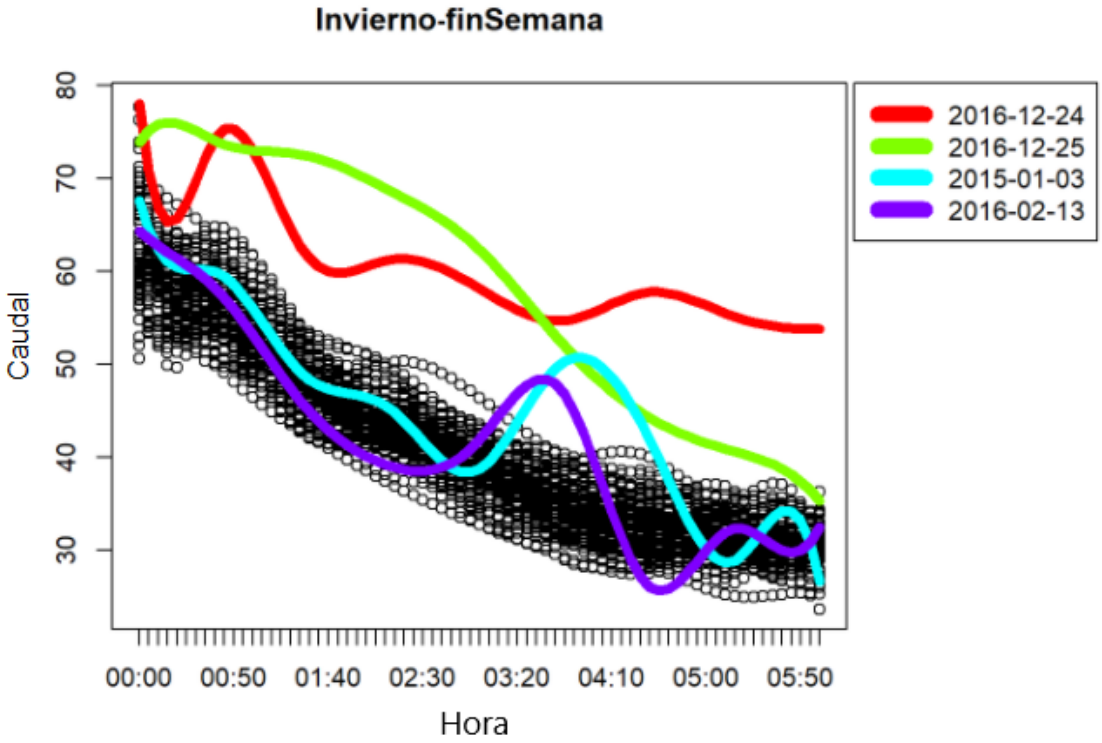
Los *outliers* obtenidos utilizando las distintas métodologías de *foutliers*, no resultan adecuados, pues, dependiendo del método escogido o se obtienen demasiados *outliers* en la forma de las funciones, o no se obtiene ninguno; por este motivo, se omiten los resultados. Sin embargo, utilizando la instrucción *fbplot* o el análisis de arquetipos, como se muestra a continuación, los resultados son más convenientes.

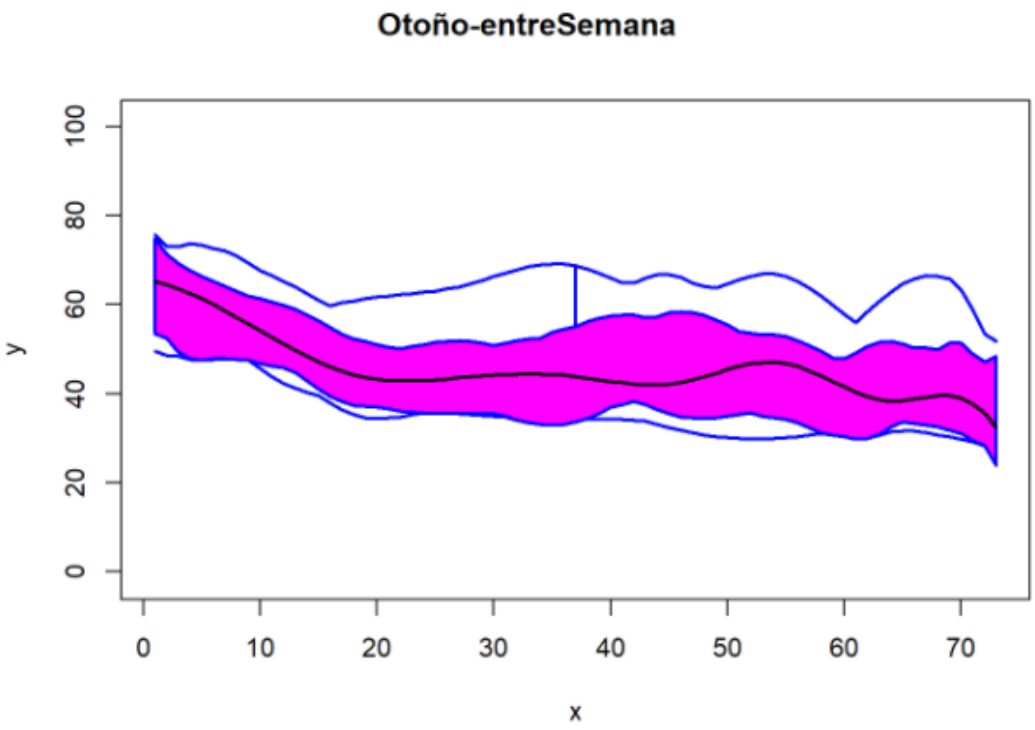
En la Figura 5.6 se muestran los resultados de aplicar la instrucción *fbplot* a los datos del sector B. Para cada grupo se representa en primer lugar el boxplot funcional y en segundo lugar la fecha de los *outliers* encontrados.

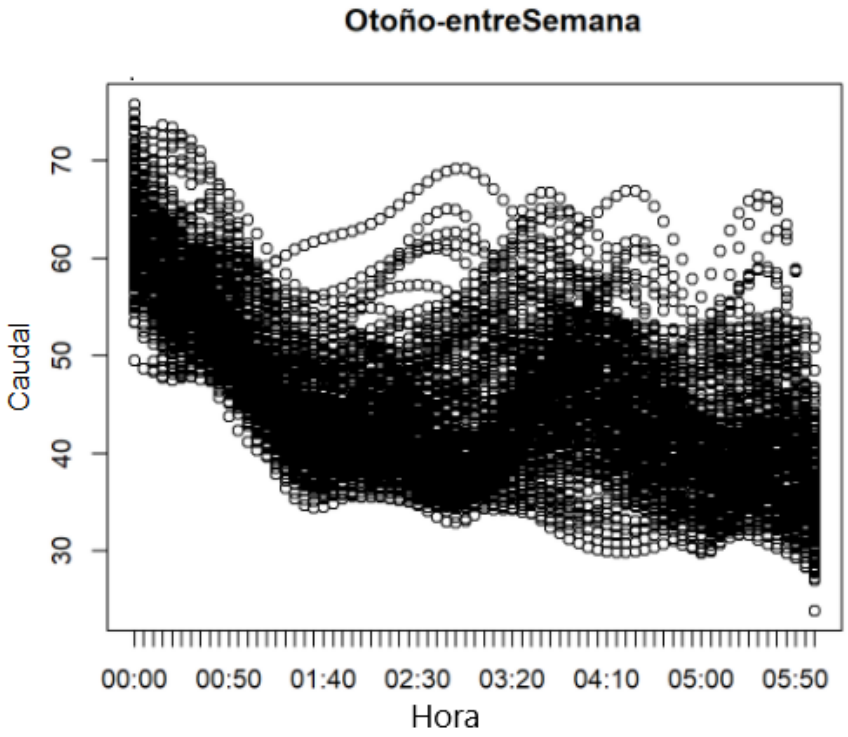

Invierno-entreSemana
y
x
0
20
40
60
80
100
0
10
20
30
40
50
60
70

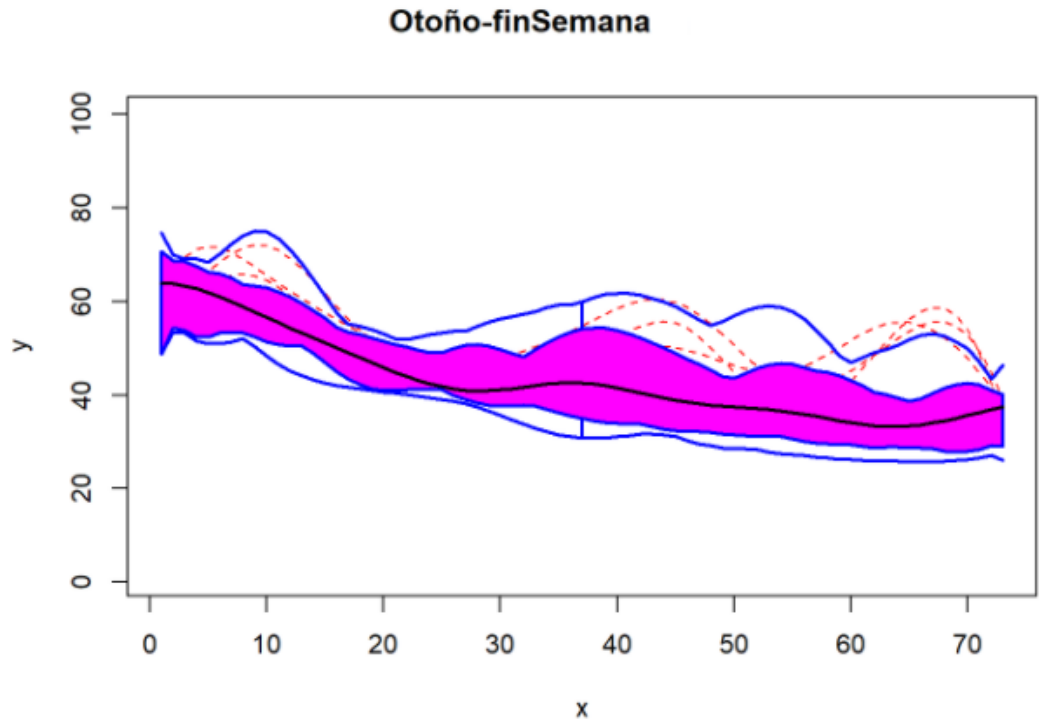

Invierno-entreSemana
Caudal
Hora
30
40
50
60
70
80
00:00
00:50
01:40
02:30
03:20
04:10
05:00
05:50
2016-12-23
2014-12-25
2015-12-25
2016-12-27
2016-12-29
2016-01-01
2015-02-13
2014-02-26

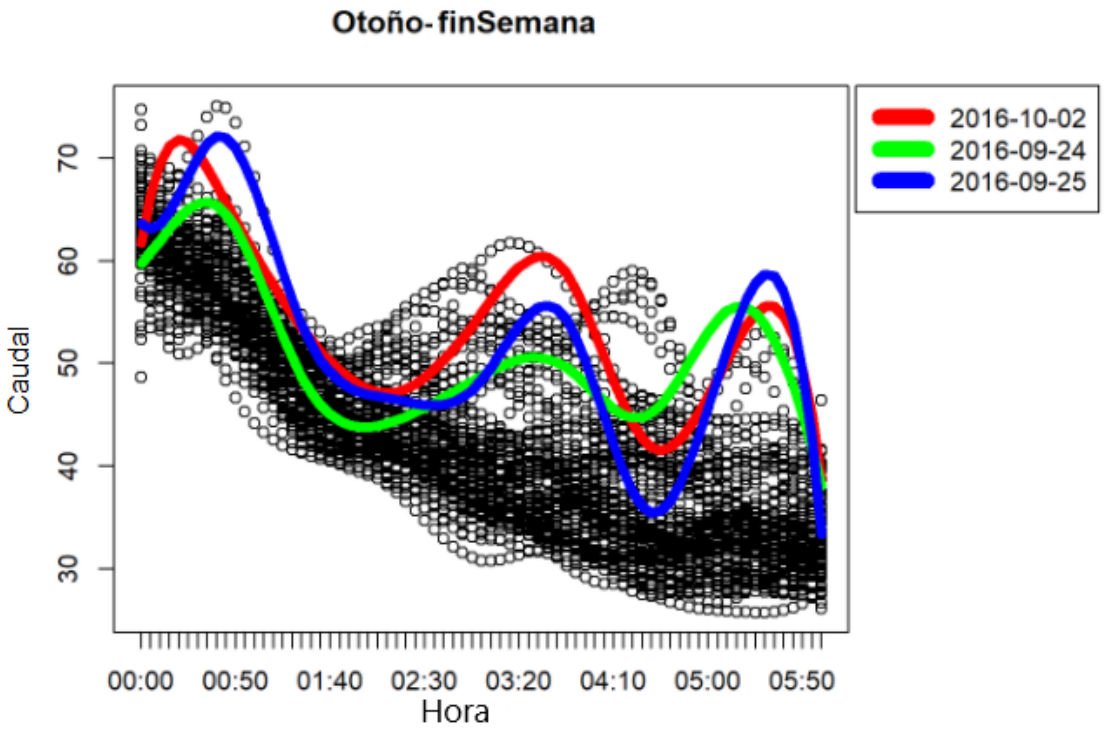

Invierno-finSemana
y
x
0
20
40
60
80
100
0
10
20
30
40
50
60
70

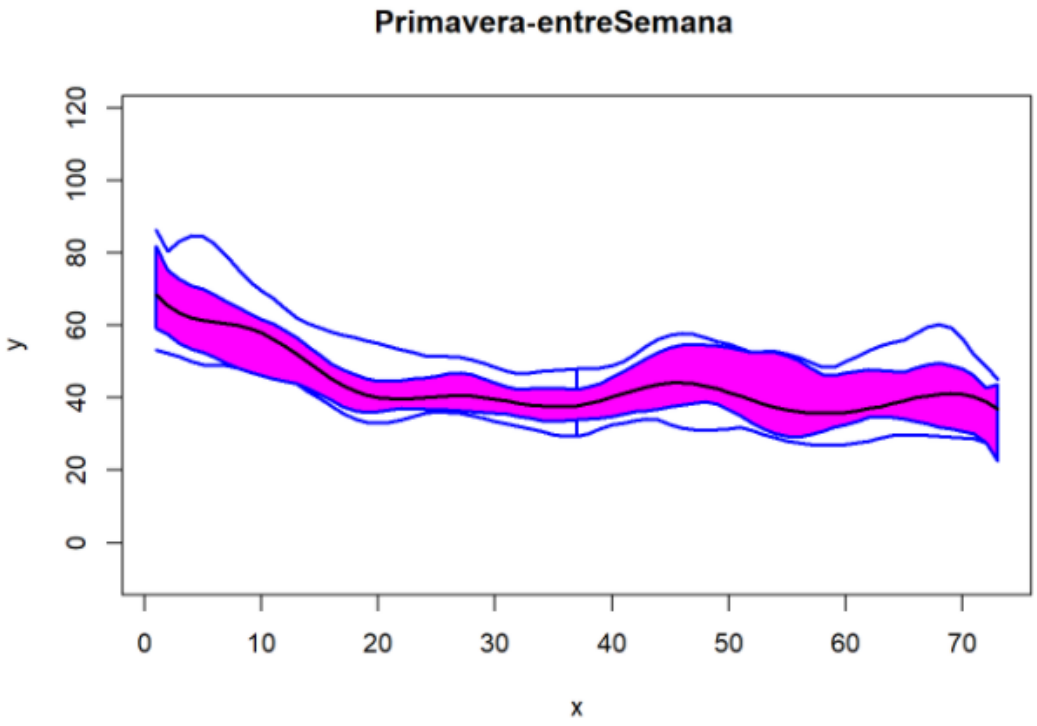

Invierno-finSemana
Caudal
Hora
30
40
50
60
70
80
00:00
00:50
01:40
02:30
03:20
04:10
05:00
05:50
2016-12-24
2016-12-25
2015-01-03
2016-02-13

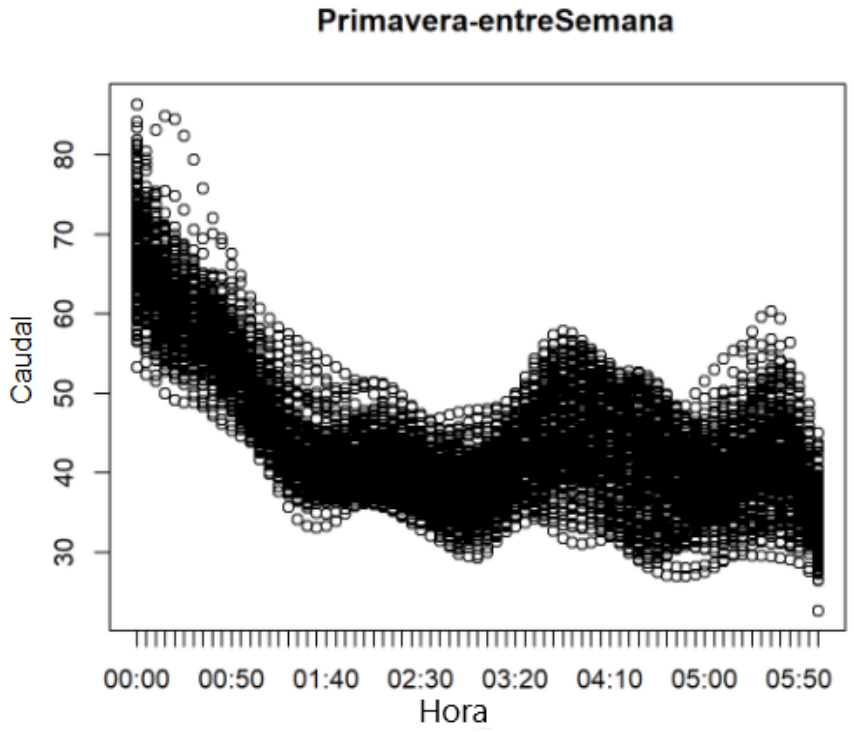

Otoño-entreSemana
y
x
0
20
40
60
80
100
0
10
20
30
40
50
60
70

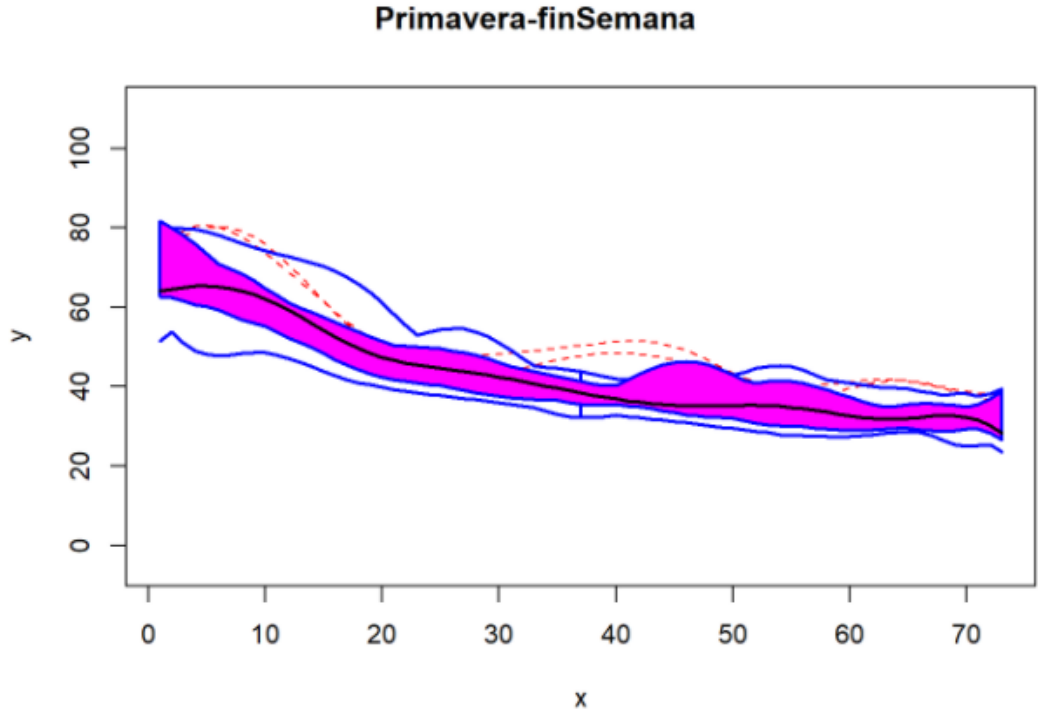

Otoño-entreSemana
Caudal
Hora
30
40
50
60
70
00:00
00:50
01:40
02:30
03:20
04:10
05:00
05:50

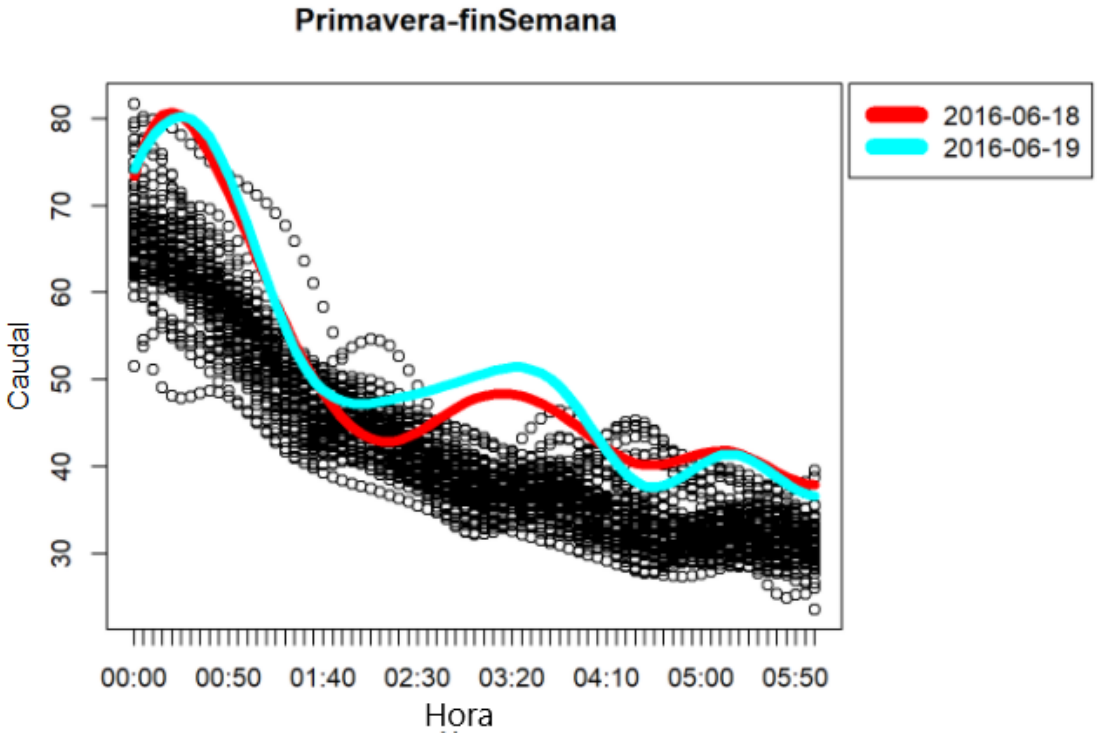

Otoño-finSemana
y
x
0
20
40
60
80
100
0
10
20
30
40
50
60
70

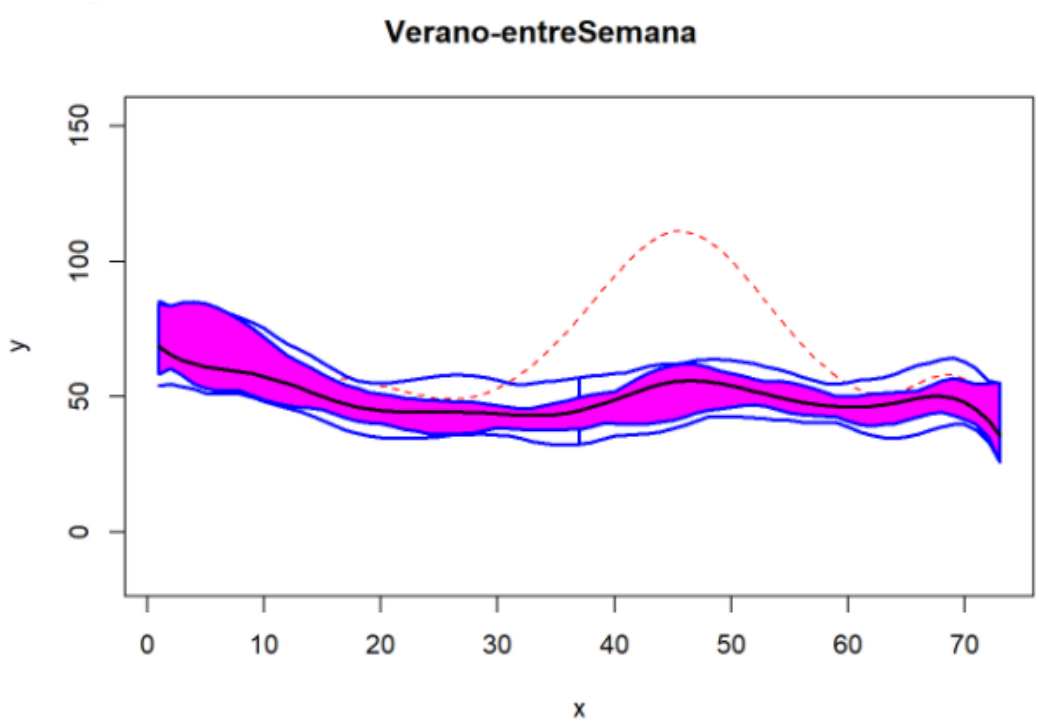

Otoño- finSemana
Caudal
Hora
2016-10-02
2016-09-24
2016-09-25
30
40
50
60
70
00:00
00:50
01:40
02:30
03:20
04:10
05:00
05:50

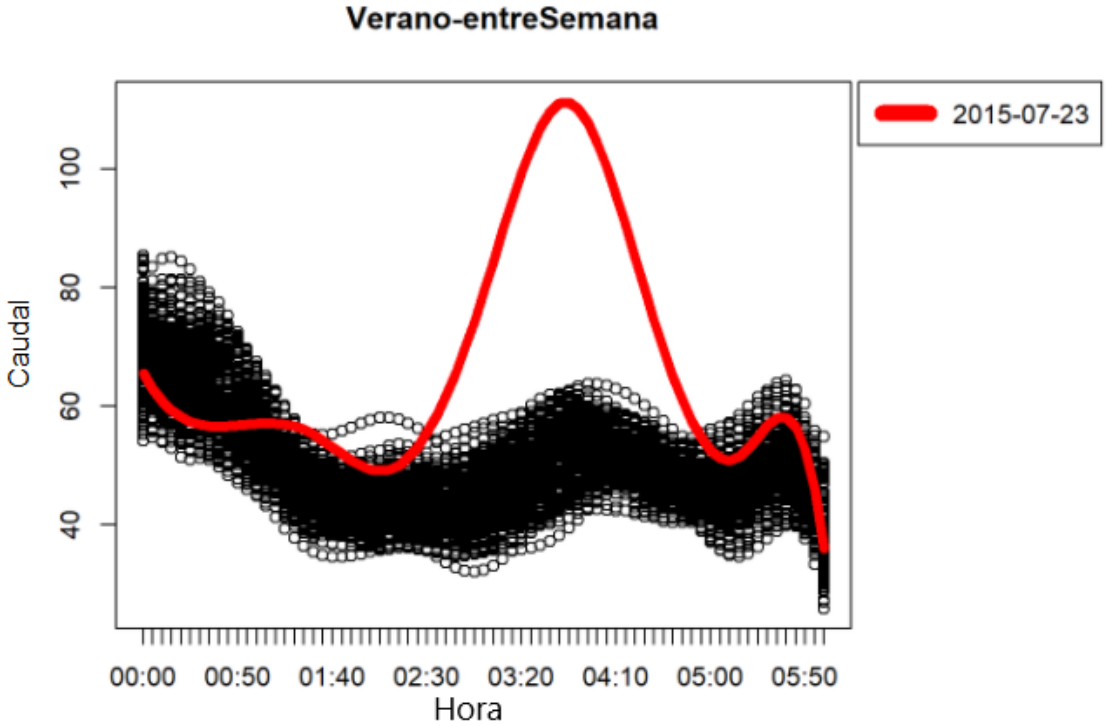

Primavera-entreSemana
y
0
20
40
60
80
100
120
0
10
20
30
40
50
60
70
x


Primavera-entreSemana
Caudal
30
40
50
60
70
80
00:00
00:50
01:40
02:30
03:20
04:10
05:00
05:50
Hora


Primavera-finSemana
y
x
0
20
40
60
80
100
0
10
20
30
40
50
60
70


Primavera-finSemana
Caudal
Hora
30
40
50
60
70
80
00:00
00:50
01:40
02:30
03:20
04:10
05:00
05:50
2016-06-18
2016-06-19


Verano-entreSemana
y
x
0
50
100
150
0
10
20
30
40
50
60
70


Verano-entreSemana
2015-07-23
Caudal
Hora
40
60
80
100
00:00
00:50
01:40
02:30
03:20
04:10
05:00
05:50

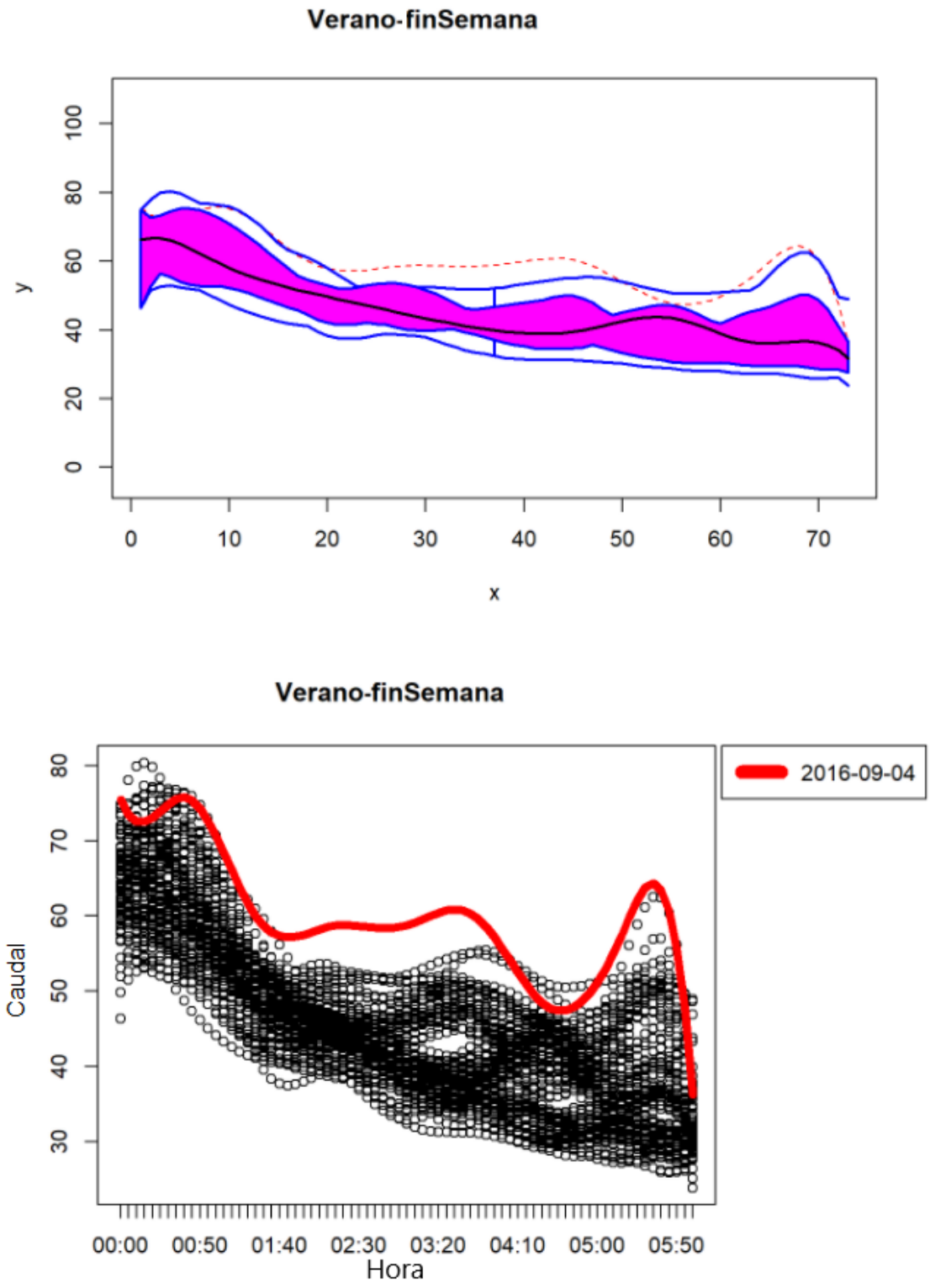


Figura 5.6: *Outliers* funcionales para el sector B con la instrucción *fbplot*.

Respecto del análisis de arquetipos, cabe decir que realizando cuatro arquetipos para cada uno de los grupos, se obtienen resultados muy similares a los que se consiguen aplicando la sentencia *fbplot*. Para invierno en fin de semana, se obtiene un arquetipo que sólo tiene coeficientes elevados para el día 25/12/2016 y otro que tiene valores elevados para los días 24/12/2016, 03/01/2016 y 14/02/2015. Además, para verano entre semana, se obtiene un arquetipo que sólo tiene valores altos para el día 23/07/2015.

Conclusiones para el sector B

En este caso, la instrucción *fbplot* y el análisis de arquetipos proporcionan mejores resultados que la sentencia *foutliers*. Entre los *outliers* obtenidos con el método *fbplot*, los que se corresponden con un mayor consumo de agua son en días festivos (24/12/2016 y 25/12/2016 para invierno en fin de semana y 25/12/2014, 25/12/2015 y 23/12/2016 para invierno entre semana), por lo que se entiende que el consumo de agua nocturno de esos días sea ligeramente superior. Del resto de *outliers*, la mayor parte difiere levemente en la forma de las funciones en comparación con el resto pero no se corresponden con consumos de agua más elevados de lo habitual. El único *outlier* que se podría corresponder con una fuga, se obtiene en verano un día entre semana (23/07/2015), donde el consumo de agua de las 2:00 a las 5:00h de la mañana alcanza valores mucho más elevados que en el resto de días de verano entre semana. Se trata de un *outlier* aislado.

5.2.3. Sector C

En esta sección se muestran los resultados de aplicar las distintas metodologías de detección de *outliers* al sector C.

De nuevo, los resultados obtenidos con las diferentes metodologías de *foutliers* no son convenientes y, por ello, se omiten. Sin embargo, los resultados obtenidos con *fbplot* y con análisis de arquetipos resultan más idóneos.

Los resultados de aplicar la instrucción *fbplot* a los datos del sector C se representan en la Figura 5.7. En ningún grupo se obtienen *outliers* salvo en verano entre semana.

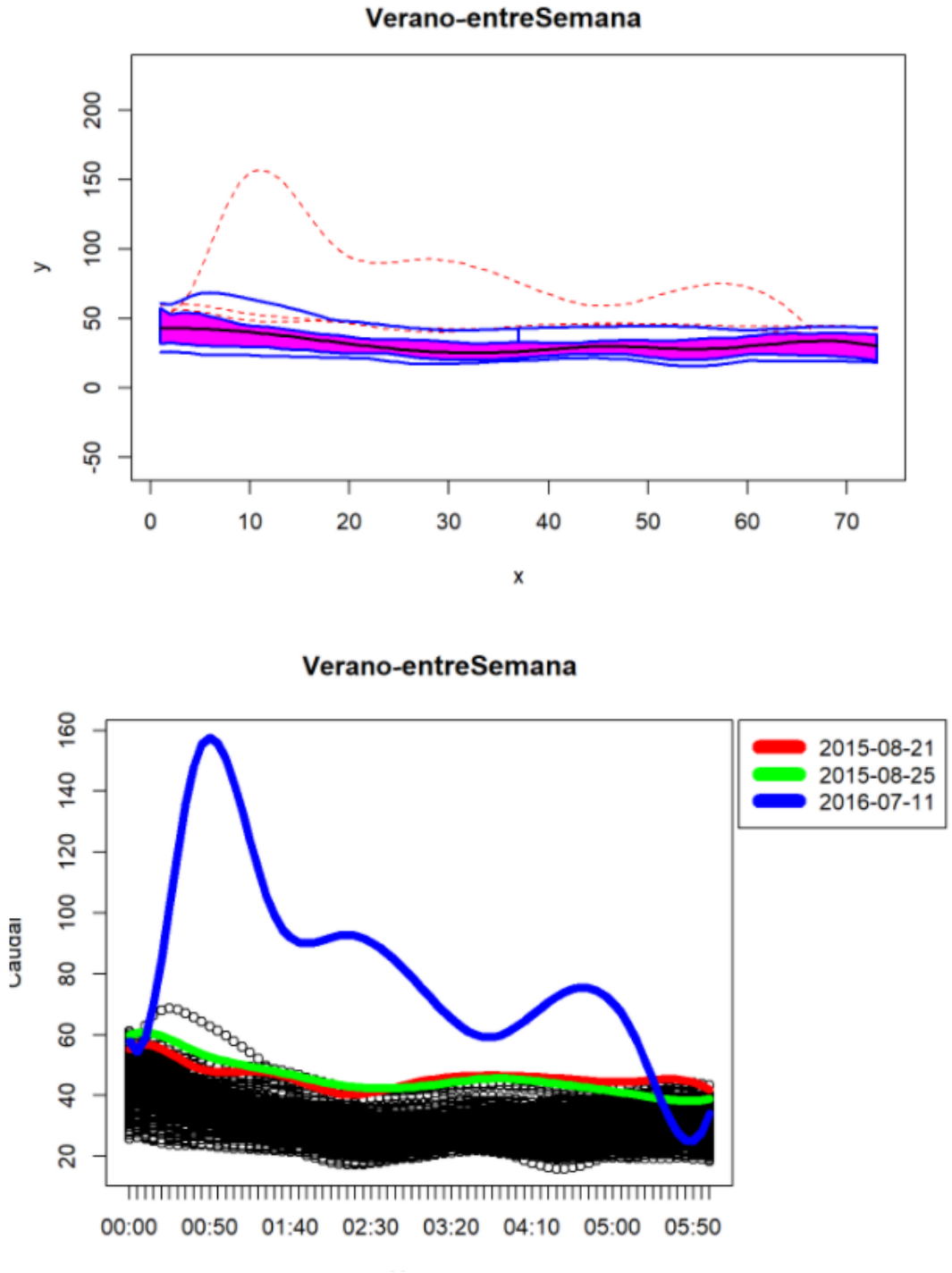


Figura 5.7: *Outliers* funcionales para el sector C con la instrucción *fbplot*.

Respecto del análisis de arquetipos, cabe decir que con tan sólo dos arquetipos ya se obtiene uno de ellos cuyos coeficientes son elevados para tan sólo una observación, el 11/07/2016, que coincide con el *outlier* en magnitud resultante de la instrucción *fbplot*.

Conclusiones para el sector C

Para el sector C, de nuevo los resultados obtenidos con *foutliers* no son adecuados para nuestro problema. Sin embargo, haciendo uso de la instrucción *fbplot* y de análisis de arquetipos se obtienen mejores resultados. En este caso, para ambos métodos se observa un claro *outlier* en verano durante un día entre semana (11/07/2016). El consumo durante toda la noche supera con creces a la cantidad de agua consumida durante el resto de días, por lo que muy probablemente se corresponda con una fuga en la red de distribución. Los otros dos *outliers* obtenidos mediante *fbplot* (21/08/2015 y 25/08/2015) no presentan valores mucho más elevados que el resto de días y, además, no se detectan como *outliers* en el análisis de arquetipos, independientemente del número de arquetipos elegido por lo que podemos descartarlos.

5.2.4. Detección de caudales anómalos nocturnos en tiempo real

Aunque el intervalo de estudio considerado para la realización del proyecto ha sido restringido a las 6 horas de la noche (de 00:00 a 06:00h), se pueden contemplar intervalos mayores o menores y empezar o acabar donde se quiera. Esto permite, entre otras cosas, poder obtener en tiempo real los posibles *outliers* de las últimas horas y, de este modo, poder detectar que algo está ocurriendo, sea o bien una fuga o un comportamiento extraño a investigar.

Para centrar el problema en un caso concreto, se toma como ejemplo el de la Figura 5.7; concretamente, el día 11/07/2016. En lugar de considerar todas las observaciones, se toman únicamente los primeros datos y se van añadiendo observaciones sucesivamente hasta que se detecte el día 11/07/2016 como un *outlier*. En la Figura 5.8 se observa que simplemente considerando los minutos comprendidos entre las 00:00 y las 00:25h, el día 11/07/2016 ya se detecta como *outlier* y, por tanto, ya se podría haber actuado en consecuencia evitando que la fuga permaneciera durante toda la noche.

Así pues, para la detección de *outliers* en tiempo real, se deben obtener los posibles *outliers* de las últimas horas o del intervalo de tiempo de detección que se haya establecido y repetir el proceso cada 5 minutos, es decir, cada vez que se registra una nueva observación. Cabe destacar que en cada ocasión se debería volver a calcular el número de funciones base necesario que se ajuste a las observaciones de la muestra.

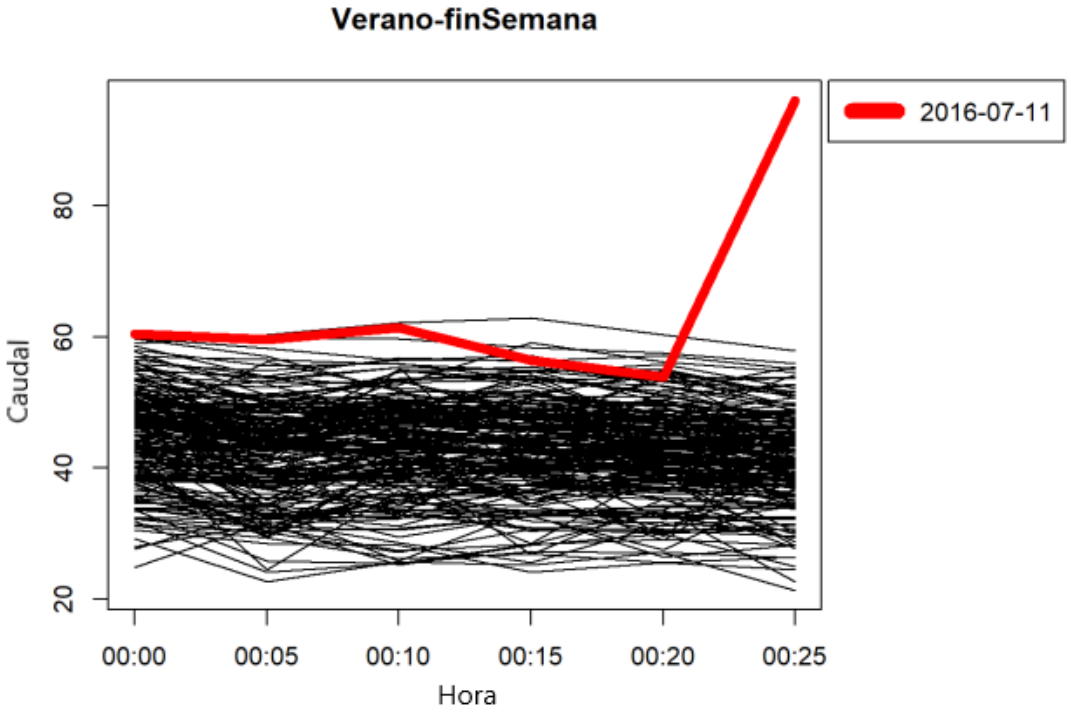


Figura 5.8: Detección de caudal anómalo para el día 11/07/2016 considerando tan sólo el intervalo comprendido de 00:00 a 00:25h.

Capítulo 6

Conclusiones generales y trabajo futuro

6.1. Conclusiones generales

El proyecto desarrollado ha sido llevado a cabo en una beca de investigación para la Cátedra Facsa de Innovación del Ciclo Integral del Agua de la UJI, con el fin de poder contribuir a reducir el problema de los elevados niveles de pérdidas de agua a los que se enfrenta cualquier empresa gestora del ciclo integral del agua.

Para impulsar la necesidad de elaborar propuestas con el objetivo de reducir la dimensión del problema, se ha decidido estudiar los valores de caudal con el fin de detectar caudales anómalos nocturnos que puedan corresponderse con pérdidas de agua reales. Así pues, se ha decidido plantear el problema desde la perspectiva de datos funcionales, ya que el consumo de agua se puede interpretar de forma natural como una función a lo largo del tiempo. Otras ventajas que han reforzado la idea de enfocar el problema con datos funcionales han sido que no es necesario tener los datos medidos en los mismos instantes de tiempo, que se puede eliminar el ruido en los datos aplicando técnicas de suavizado (ajustar los datos a una base) y que puede haber valores faltantes. Además, en el caso de una elevada presencia de valores faltantes en la muestra, se pueden utilizar datos funcionales *sparse* para estimar los valores faltantes, basándose en el consumo de agua del resto de días.

Para transformar los datos obtenidos de manera discreta a datos funcionales se han utilizado bases B-spline por su popularidad y su eficiente coste computacional. En concreto, se han escogido bases B-spline de orden 4 (porque no se iban a utilizar derivadas), nodos equiespaciados y, para la elección del número de funciones base, se ha ido probando con distinto número de bases en orden ascendente y representando la media de la varianza de los residuos de todas las funciones de la muestra frente al número de bases mientras ésta decreciera rápidamente. Se ha seleccionado el número de funciones base como el valor para el que la varianza media deja de decrecer rápidamente.

La detección de caudales anómalos nocturnos se ha realizado aplicando las distintas metodologías de los paquetes del software R. Las instrucciones que se han utilizado han sido *fbplot* del paquete *fda* y *foutliers* del paquete *rainbow*. Además, también se ha empleado el análisis de arquetipos con el objetivo de detectar *outliers*. Estas técnicas se han aplicado a los datos registrados, durante los tres últimos años, en tres sectores de la provincia de Castellón. Para su aplicación, se ha decidido dividir previamente la muestra por estaciones y por días laborales y días de fin de semana, ya que se ha detectado que dependiendo de las estaciones y de si se trata de días laborales o de fin de semana, el patrón de consumo es distinto. Una vez aplicadas las diversas metodologías, se han comparado los resultados obtenidos para extraer conclusiones. En este proyecto los resultados alcanzados mediante *fbplot* y mediante el análisis de arquetipos son más adecuados que los proporcionados con *foutliers*, ya que los diferentes métodos de la instrucción *foutliers* o son demasiado sensibles y obtienen muchos *outliers*, o por el contrario no obtienen ningún *outlier*.

Finalmente, observando los *outliers* de cada sector, las conclusiones que se han extraído son que para el sector A no ha habido ninguna fuga durante los años 2014, 2015 y 2016, mientras que para los otros dos sectores se han encontrado dos noches donde el consumo de agua ha sido mucho más elevado que en el resto de días y por tanto, se pueden corresponder con fugas en la red de distribución. En el caso del sector B, la anomalía se prolongó tan sólo durante unas horas, mientras que en el sector C duró prácticamente toda la noche.

6.2. Trabajo futuro

Las futuras líneas de investigación, que se pueden destacar, para la continuación del estudio realizado son:

- Diseñar un entorno gráfico de forma que los trabajadores de Facsa puedan interaccionar con el programa realizado, de forma más práctica y cómoda, sin necesidad de lidiar con el código en R.
- Implantar el algoritmo realizado en el sistema informático de la empresa.
- Emplear no sólo los valores de caudal sino otras variables funcionales como la presión. En ese caso, se deberían utilizar técnicas para datos funcionales multivariantes.
- Idear un nuevo método de detección de *outliers* para datos funcionales, ya que los métodos existentes en las librerías de R, o no detectan prácticamente *outliers*, o son demasiado sensibles y obtienen muchos *outliers*, que a simple vista no parecen tener una trayectoria que difiera en exceso del resto de curvas.
- Detectar *outliers* de datos funcionales en otros campos de consumo como podría ser en la industria eléctrica. Un ejemplo de ello se puede observar en P. Raña, J.M. Vilar y G. Aneiros (2013) [31].

- Aplicar técnicas de FDA en otros problemas relacionados con el consumo del agua. Se puede consultar la publicación realizada por Z. Noumir, A. Samé, N. Cheifetz, A.C. Sandraz et C. Féliers (2015) [32] donde se realizan clústers de datos funcionales para analizar el consumo de agua potable.

Bibliografía

[1] IBM, Information On Demand 2011. http://andresarayafalcone.blogspot.com.es/.

[2] F. Gavara (2015), *Estudio del comportamiento metrológico de los contadores en abastecimientos de agua. Optimización de su gestión para la reducción de las pérdidas comerciales.*

[3] Página web de Facsa: http://www.facsa.com/empresa/empresa

[4] Página web de Grupo Gimeno: http://www.grupogimeno.com/es/historia-presencia-localizacion

[5] R Core Team (2017), *R: A language y environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL `http://www.R-project.org/`

[6] B. Efron y T. Hastie (2016), *Computer Age Statistical Inference: Algorithms, Evidence, and Data Science*. Cambridge University Press.

[7] I. L. Dryden y J. T. Kent (2015), *Geometry Driven Statistics*. John Wiley and Sons.

[8] D. Peña (2014), *Big Data and Statistics: Trend or Change?*. Boletín de Estadística e Investigación Operativa Vol. 30, No. 3.

[9] J. Ramsay y B. Silverman (2005), *Functional Data Analysis.* Springer.

[10] F. Ferraty y P. Vieu (2006), *Nonparametric Functional Data Analysis: Theory and Practise.* Springer.

[11] C. de Boor (2002), *A Practical Guide to Splines*. Revised Edition, Springer.

[12] P. Craven y G. Wahba (1979), *Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation.* Numerische Mathematik.

[13] D. M. Hawkins (1980), *Identification of Outliers*. London, Chapman and Hall.

[14] P. Rousseeuw y A. Leroy (1987), *Robust Regression and Outlier Detection*. Wiley-Interscience.

[15] R. Maronna, D. Martin y V. Yohai (2006), *Robust Statistics: Theory and Methods*. Wiley.

[16] M. Febrero, P. Galeano y W. González-Manteiga (2008), *Outlier detection in functional data by depth measures, with application to identify abnormal NOx levels.*

[17] M. Hubert, P. Rousseeuw y P. Segaert (2015), *Multivariate Functional Outlier Detection.*

[18] F. Larsen, F. van den Berg y S. Engelsen (2006), *An exploratory chemometric study of NMR spectra of table wines.* Journal of Cheomometrics 20(5).

[19] M. Dyrby, S. Engelsen, L. Norgaard, M.Bruhn, L. Lundsberg-Nielsen (2002), *Chemometric quantization of the active substance in a pharmaceutical tablet using nearinfrared (NIR) transmittance and NIR FT-Raman spectra*. Applied Spectroscopy 56(5).

[20] M. Febrero, P. Galeano y W. González-Manteiga (2007), *A functional analysis of NOx levels: location and scale estimation and outlier detection.*

[21] R. Fraiman y G. Muniz (2001), *Trimmed means for functional data*. Test, 10, 419-440.

[22] A. Cuevas, M. Febrero, R. Fraiman (2006), *On the use of bootstrap for estimating functions with functional data.* Computational Statistics and Data Analysis 51: 1063–1074.

[23] A. Cuevas, M. Febrero, R. Fraiman (2007), *Robust estimation and classification for functional data via projection-based depth notions. Computational Statistics.*

[24] R. J. Hyndman y M.S. Ullah (2007), *Robust forecasting of mortality and fertility rates: A functional data approach.* Computational Statistics and Data Analysis, 51, 4942–4956.

[25] M. Hubert, P.J, Rousseeuw y S. Verboven (2002), *A fast method of robust principal components with applications to chemometrics*. Chemometrics and Intelligent Laboratory Systems, 60: 101-111.

[26] Y. Sun y M. Genton (2011), *Functional boxplots* .

[27] S. López-Pintado y J. Romo (2009), *On the Concept of Depth for Functional Data*. Journal of the American Statistical Association, 104, 718–734.

[28] A. Cutler y L. Breiman (1994), *Archetypal Analysis*. Technometrics, 36 (4).

[29] G. Vinué, I. Epifanio, S. Alemany (2015), *Archetypoids: A new approach to define representative archetypal data.* Computational Statistics and Data Analysis 87.

[30] I. Epifanio (2016), *Functional archetype and archetypoid analysis*. Computational Statistics and Data Analysis, 104, 24-34

[31] P. Raña, J.M. Vilar y G. Aneiros (2013), *Detección de atípicos en datos funcionales dependientes.*

[32] Z. Noumir, A. Samé, N. Cheifetz, A.C. Sandraz et C. Féliers (2015), *Décomposition et clustering de données fonctionnelles pour l'analyse de la consommation d'eau potable*

Anexo A

Listas de programas en R

A.1. Cargar paquetes *fda*, *rainbow* y *archetypes*

```
library(fda) #Para trabajar con datos funcionales (suavizado y
   deteccion de outliers con fbplot)
```

```
library(rainbow) #Para obtener outliers con la instruccion foutliers
```

```
library(archetypes) #Para obtener outliers haciendo uso de arquetipos
```

A.2. Calcular la media de la varianza de los residuos para elegir el número de funciones base

```
#Nombre de la zona que se va a analizar
zona = 'A'

#Elegimos las horas que queremos seleccionar
horaInicio = '00:00'
horaFin = '06:00'

#Creamos un objeto con la hora de inicio y la hora de fin para poder
   comparar el resto de horas con estas y saber si debemos
   seleccionarlas o no.
horaInicioTimeObj = strptime( horaInicio, format = "%H:%M" )
horaFinTimeObj = strptime( horaFin, format = "%H:%M" )

```

```
#Creamos un vector con las horas que nos interesan para poder despues hacer el plot y que nos ponga las horas correctamente en el eje de las x.
vectorHoras = c(horaInicio)
hora = horaInicioTimeObj
while( hora != horaFinTimeObj ){
  hora = hora + 5*60
  vectorHoras = c(vectorHoras, format(hora, '%H:%M'))
}


#Leemos los datos
datos = read.csv('Indicar entre comillas la ruta del archivo cuya extension debe ser .csv', header = T, sep=";", dec = ",")


#Nos quedamos solo con las columnas que nos interesen
datosZona = data.frame(datos$Fecha, datos$Dia, datos$Hora, datos[[zona]])

names(datosZona) = c("Fecha", "Dia", "Hora", "caudal")

#Tranformamos la columna de las horas a formato hora
datosDiaFormatoHora = strptime(datosZona$Hora,format="%H:%M")

#Incluimos esa columna al data.frame para poder quedarnos solo con las horas que esten comprendidas en un rango
datosZona = cbind(datosZona, datosDiaFormatoHora)

#Nos quedamos solo con las horas que queremos: las que estan entre la hora de inicio y la hora de fin.
datosDia = datosZona[datosZona$datosDiaFormatoHora  >= horaInicioTimeObj & datosZona$datosDiaFormatoHora <= horaFinTimeObj, c(1:dim(datosZona)[2]-1) ]

vectorDatosDia = split(datosDia, f = datosDia$Dia)

#Numero de observaciones en cada funcion
p = dim(vectorDatosDia[[1]])[1]

#Numero de funciones en la muestra
n = 0

for(dia in vectorDatosDia){

  if( sum(complete.cases(dia$caudal)) != 0){ #No todo NA

    #Aumentamos en una unidad el numero de funciones de la muestra
    n = n + 1

    #Rango de funciones base con el que probar
    basesIni = 4
```

```
    basesFin = 22

    #Vector con el numero de funciones base dentro del rango
    bases = seq(basesIni, basesFin)

    #Vector con la suma de los residuos para cada numero de funciones
       base
    suma = rep(0, basesFin - basesIni + 1)

    #Ajustar para cada numero de funciones base y calcular los residuos
    j = 1

    for(nbase in bases){

      mibspline = create.bspline.basis(rangeval = c(0,length(dia$caudal
         )), nbasis = nbase, norder = 4)

      aprxbspline = Data2fd(argvals = c(which(!is.na(dia$caudal))), y =
          c(na.omit(dia$caudal)), basisobj = mibspline)

      for( i in 1:length(dia$caudal) ){

        evalua = eval.fd(evalarg = i , fdobj = aprxbspline)

        suma[j] = suma[j] + (dia$caudal[i] - evalua)^2

      }
      j = j+1
    }
  }
}

s2 = suma/(n*(p - bases))

#Dibujar
plot(s2, type = "b", xlab = "Numero de funciones base", ylab = "
   Varianza estimada", xaxt = "n", col = "blue")
axis(1, at = seq(1,basesFin - basesIni + 1), labels = bases, las=2, cex
   .axis = 0.7)
```

A.3. Hacer grupos por estaciones y días laborales o días de fin de semana

```
#Indicamos la hora de inicio y la hora de fin que queremos considerar
horaInicio = '00:00'
horaFin = '06:00'

```

```
5  #Creamos un objeto con la hora de inicio y la hora de fin para poder
       realizar facilmente las comparaciones y seleccionar tan solo las
       horas comprendidas en el rengo de horas de interes
6  horaInicioTimeObj = strptime( horaInicio, format = "%H:%M" )
7  horaFinTimeObj = strptime( horaFin, format = "%H:%M" )
8
9  #Creamos un vector con las horas con las horas comprendidas entre la
       hora de inicio y la hora de fin para poder imprimirlo en el eje de
       las x al realizar los graficos
10 vectorHoras = c(horaInicio)
11 hora = horaInicioTimeObj
12 while( hora != horaFinTimeObj ){
13   hora = hora + 5*60 #Se aumenta la hora en 5 minutos
14   vectorHoras = c(vectorHoras, format(hora, '%H:%M'))
15 }
16
17 #Indicamos el sector que queremos analizar. Se debe corresponder con el
        titulo de la columna del fichero Excel que contiene los valores de
       caudal para el sector a considerar
18 zona = "A"
19
20 #Vector para almacenar los datos separados por grupos (estaciones y
       dias laborales o dias de fin de semana)
21 vectorDataFrames = list()
22 indice = 1
23
24 #Leemos los datos
25 datos = read.csv('Indicar entre comillas la ruta del archivo cuya
       extension debe ser .csv', header = T, sep = ";", dec = ",")
26
27 #Nos quedamos solo con algunas columnas
28 datos = data.frame(datos$Dia,datos$Mes, datos$SemanaOFinde, datos$
       Estacion, datos$Hora, datos[[zona]])
29 names(datos) = c('Dia', 'Mes', 'SemanaOFinde', 'Estacion', 'Hora', '
       caudal')
30
31 #Separamos por estaciones
32 vectorDatosEstacion = split(datos, f = datos$Estacion)
33
34 for( datosEstacion in vectorDatosEstacion ) {
35
36         #Separamos por dia entre semana o dia de fin de semana
37         vectorDatosSemanaOFinde = split( datosEstacion, f =
               datosEstacion$SemanaOFinde )
38
39         for( semanaOFinde in vectorDatosSemanaOFinde ) {
40
41                 #Ponemos la hora a formato de hora con strptime
42                 datosFormatoHora = strptime(semanaOFinde$Hora,format="%
                       H:%M")
43
```

```
                #Incluimos esa columna al data.frame para poder
                   quedarnos solo con las horas que esten comprendidas
                   en un rango (noche)
                semanaOFinde = cbind(semanaOFinde, datosFormatoHora)

                #Nos quedamos solo con las horas que queremos: las que
                   estan entre la hora de inicio y la hora de fin.
                semanaOFinde2 = semanaOFinde[semanaOFinde$
                   datosFormatoHora  >= horaInicioTimeObj &
                   semanaOFinde$datosFormatoHora <= horaFinTimeObj, ]

                vectorDataFrames[[indice]] =  semanaOFinde2[,-c(dim(
                   semanaOFinde)[2])]

                indice = indice + 1

        }
}
```

A.4. Suavizar

```
#Lista para almacenar los datos suavizados de cada grupo
listaDatosSuavizados = list()
indice = 1

for( grupo in vectorDataFrames ){

  #Separamos por meses
  vectorDatosMes = split(grupo, f = grupo$Mes)

  #Creamos una hoja de datos para almacenar los datos suavizados
  datosSuavizados = data.frame()

  #Creamos un vector para almacenar las fechas y poder asociarlos a los
      nombres de la hoja de datos
  dia = c()

  for( datosMes in vectorDatosMes ){

    #Separamos cada mes por dias
    datosPorDia = split( datosMes, f = datosMes$Dia )

    for( datosDia in datosPorDia ){

       #Elegir numero de funciones base
       nbase = 12

       #Crear el sistema de bases B-spline
```

```
27          mibspline = create.bspline.basis(rangeval = c(0,length(datosDia$
                caudal)), nbasis = nbase, norder = 4)
28
29          #Ajustar los datos a la base
30          aprxbspline = Data2fd(argvals = c(which(!is.na(datosDia$caudal))
                ), y = c(na.omit(datosDia$caudal)), basisobj = mibspline)
31
32          #Vector para almacenar los valores suavizados de un dia
33          diaSuavizado = c()
34
35          #Almacenar las fechas
36          dia = c(dia, toString(as.Date(datosDia$Dia[1], format = "%d/%m/%
                Y")))
37
38          #Obtener los valores suavizados
39          for( i in 1:length(datosDia$caudal) ){
40
41              evalua = eval.fd(evalarg = i , fdobj = aprxbspline)
42
43              diaSuavizado = c(diaSuavizado, evalua)
44          }
45          #Almacenar los datos suavizados de un dia en la hoja de datos
                donde se almacenan todos los datos suavizados de un grupo
46          datosSuavizados = rbind(datosSuavizados, diaSuavizado)
47      }
48    }
49    #Poner las horas y las fechas en la hoja de datos suavizados de un
          grupo
50    names(datosSuavizados) = vectorHoras
51    row.names(datosSuavizados) = dia
52
53    #Almacenar los datos suavizados de un grupo en una posicion de la
          lista listaDatosSuavizados
54    listaDatosSuavizados[[indice]] = datosSuavizados
55
56    indice = indice + 1
57  }
```

A.5. Búscar y dibujar *outliers* con *foutliers* y *fbplot*

A.5.1. Búscar *outliers* con *foutliers*

```
1
2  #Lista con los nombres de los grupos
3  estaciones = c("Invierno-entreSemana", "Invierno-finSemana", "Otono-
       entreSemana", "Otono-finSemana", "Primavera-entreSemana", "Primavera
       -finSemana", "Verano-entreSemana", "Verano-finSemana")
```

```
4
5  #Lista para almacenar los outliers de cada grupo
6  listaOutliers = list()
7  indice = 1
8
9  for( grupo in listaDatosSuavizados ){
10
11          #Busqueda de outliers
12      f = fts(x=1:dim(grupo)[2], y=t(grupo))
13      out = foutliers(f, method = "lrt") # A elegir entre "lrt", "depth.
            trim", "depth.pond" y "HUoutliers"
14
15      #Almacenar outliers
16      listaOutliers[[indice]] = out$outliers
17
18      #Imprimir outliers
19      print(estaciones[indice])
20      print(row.names(grupo[out$outliers),])
21
22      indice = indice + 1
23 }
```

A.5.2. Búscar *outliers* con *fbplot*

```
1  #Lista con los nombres de los grupos
2  estaciones = c("Invierno-entreSemana", "Invierno-finSemana", "Otono-
       entreSemana", "Otono-finSemana", "Primavera-entreSemana", "Primavera
       -finSemana", "Verano-entreSemana", "Verano-finSemana")
3
4  #Lista para almacenar los outliers de cada grupo
5  listaOutliers = list()
6  indice = 1
7
8  for( grupo in listaDatosSuavizados ){
9      #Busqueda de outliers
10     trans = t(grupo)
11     names(trans) = row.names(grupo)
12     out = fbplot(t(grupo))
13
14     #Almacenar outliers
15     listaOutliers[[indice]] = out$outpoint
16
17     #Imprimir outliers
18     print(estaciones[indice])
19     print(names(trans[out$outpoint]))
20
21     indice = indice + 1
22 }
```

A.5.3. Dibujar *outliers* obtenidos con *foutliers*

```
for( j in 1:length(listaDatosSuavizados) ){

  #Variable booleana indicando que se trata del primer dia
  primero = TRUE

  par(mar=c(3.6, 3.6, 4.1, 8.1), xpd=TRUE)

  indice = 1

  for( i in 1:dim(listaDatosSuavizados[[j]])[1] ){

    if(length(grupo[i,]) != 0){

      #Comprobar si se trata del primer dia
      if( primero ){

        #Poner primero a FALSE para indicar que en la siguiente
           iteracion ya no sera el primer dia
        primero = FALSE

        #Comprobar si no se trata de un outlier
        if( !is.element(i,listaOutliers[[j]]) ){

           #Dibujar
           plot( 1:length(listaDatosSuavizados[[j]][i,]),
              listaDatosSuavizados[[j]][i,], xaxt = "n", ylim = c(min(
              listaDatosSuavizados[[j]], na.rm = TRUE), max(
              listaDatosSuavizados[[j]], na.rm = TRUE)), xlab = "Horas"
              , ylab = "Caudal", main = estaciones[j])

           axis(1, at=1:length(listaDatosSuavizados[[j]][i,]), labels=
              vectorHoras)

        }

      #Si no es el primer dia
      }else{

          #Comprobar si no se trata de un outlier
          if( !is.element(i,listaOutliers[[j]]) ){

             #Incluir el grafico sobre el anterior con points
             points(1:length(listaDatosSuavizados[[j]][i,]),
                listaDatosSuavizados[[j]][i,])
          }
       }
    }
  }

  #Dibujar los outliers con colores sobre el grafico
```

```
  for( i in 1:dim(listaDatosSuavizados[[j]])[1] ){

     if( is.element(i,listaOutliers[[j]]) ){

         points(1:length(listaDatosSuavizados[[j]][i,]),
            listaDatosSuavizados[[j]][i,], col = rainbow(length(
            listaOutliers[[j]]))[indice], type = 'l', lwd = 6)

         indice = indice + 1
     }
  }

  #Imprimir leyenda
  legend("topright", inset=c(-0.34,0),legend = row.names(
     listaDatosSuavizados[[j]][listaOutliers[[j]], ]), col = rainbow(
     length(listaOutliers[[j]])), lwd = 10)

}
```

A.5.4. Dibujar *outliers* obtenidos con *fbplot*

```
for( j in 1:length(listaDatosSuavizados) ){

    #Variable booleana indicando que se trata del primer dia
    primero = TRUE

    par(mar=c(3.6, 3.6, 4.1, 8.1), xpd=TRUE)

    indice = 1

    for( i in 1:dim(listaDatosSuavizados[[j]])[1] ){

        if(length(grupo[i,]) != 0){

            #Comprobar si se trata del primer dia
            if( primero ){

                #Poner primero a FALSE para indicar que en la siguiente
                    iteracion ya no sera el primer dia
                primero = FALSE

                #Comprobar si no se trata de un outlier
                if( !(row.names(datosSuavizados[i,]) %in%  names(trans[
                   listaOutliers[[j]]])) ){

                    #Dibujar
                    plot( 1:length(listaDatosSuavizados[[j]][i,]),
                       listaDatosSuavizados[[j]][i,], xaxt = "n", ylim
                       = c(min(listaDatosSuavizados[[j]], na.rm = TRUE)
```

```
                        , max(listaDatosSuavizados[[j]], na.rm = TRUE)),
                         xlab = "Horas", ylab = "Caudal", main =
                        estaciones[j])
25
26                    axis(1, at=1:length(listaDatosSuavizados[[j]][i,]),
                          labels=vectorHoras)
27                }
28
29            #Si no es el primer dia
30            }else{
31                #Comprobar si no se trata de un outlier
32                if( !(row.names(datosSuavizados[i,]) %in%  names(trans[
                      listaOutliers[[j]]])) ){
33
34                #Incluir el grafico sobre el anterior con points
35                points(1:length(listaDatosSuavizados[[j]][i,]),
                      listaDatosSuavizados[[j]][i,])
36            }
37        }
38    }
39  }
40
41  #Dibujar los outliers con colores sobre el grafico
42  for( i in 1:dim(listaDatosSuavizados[[j]])[1] ){
43
44    if( row.names(datosSuavizados[i,]) %in%  names(trans[listaOutliers
        [[j]]]) ){
45
46      points(1:length(listaDatosSuavizados[[j]][i,]),
          listaDatosSuavizados[[j]][i,], col = rainbow(length(
          listaOutliers[[j]]))[indice], type = 'l', lwd = 6)
47      indice = indice + 1
48    }
49  }
50  if(length(listaOutliers[[j]]) != 0){
51
52    #Imprimir leyenda
53    legend("topright", inset=c(-0.34,0),legend = row.names(
        listaDatosSuavizados[[j]][listaOutliers[[j]], ]), col = rainbow
        (length(listaOutliers[[j]])), lwd = 10)
54  }
55 }
```

A.6. Búscar *outliers* mediante arquetipos

Una vez aplicado el código de la sección A.3, para separar la muestra por grupos (estaciones y días entre semana o de fin de semana), se debe aplicar el siguiente código para la realización

de arquetipos.

```
#Cargar las funciones de http://www3.uji.es/~epifanio/RESEARCH/faa.rar
source("C:/Users/Laura/Downloads/faa/FFunctions_to_calculate_real_
    archetypes_with_swap2.R")
source("C:/Users/Laura/Downloads/faa/FFunctions_to_calculate_real_
    archetypes_with_swap2B.R")
source("C:/Users/Laura/Downloads/faa/FstepArchetypesMod.R")
source("C:/Users/Laura/Downloads/faa/FstepArchetypesModB.R")
source("C:/Users/Laura/Downloads/faa/FstepArchetypoids.R")
source("C:/Users/Laura/Downloads/faa/FstepArchetypoidsB.R")

listaDatos = list()
indice = 1

for( grupo in vectorDataFrames ){

  vectorDatosMes = split(grupo, f = grupo$Mes)
  datosGrupo = data.frame()
  dia = c()

  for( datosMes in vectorDatosMes ){

    #Separamos cada mes por dias
    datosPorDia = split( datosMes, f = datosMes$Dia )

    for( datosDia in datosPorDia ){

       if( dim(datosDia)[1] != 0 ){

          datosGrupo = rbind(datosGrupo, datosDia$caudal)
          dia = c(dia, toString(as.Date(datosDia$Dia[1], format = "%d/%
              m/%Y")))

        }
      }
    }
  names(datosGrupo) = vectorHoras
  row.names(datosGrupo) = dia
  listaDatos[[indice]] = datosGrupo
  indice = indice + 1
}


#Busqueda de arquetipos para cada grupo
estaciones = c("Invierno-entreSemana", "Invierno-finSemana", "Otono-
    entreSemana", "Otono-finSemana", "Primavera-entreSemana", "Primavera
    -finSemana", "Verano-entreSemana", "Verano-finSemana")

for( f in 1:8 ){

  grupo = na.omit(listaDatos[[f]])
```

```
tempmat <- t(grupo)
mibspline = create.bspline.basis(rangeval = c(1,73), nbasis = 12,
    norder = 4)
dimnames(tempmat) <- list(names(grupo), row.names(grupo))
tempfd <- Data2fd(tempmat, seq(1,73), mibspline)

X=t(tempfd$coefs)

PM=eval.penalty(mibspline, rng=c(1,73))

K=5 #Numero de arquetipos
set.seed(2015)

norep=20
lass10 <- FstepArchetypesMod(data=X,k=K,verbose=FALSE,nrep=norep,PM=
    PM,saveHistory=F)
ai <- bestModel(lass10[[1]]) #arquetipos

an4 <-  FstepArchetypoids(K,TRUE,X,lass10,FALSE,K-1,PM)

aw4 <-  FstepArchetypoids(K,FALSE,X,lass10,FALSE,K-1,PM)

ini_arch=c()
for (j in 1:K){
  ini_arch[j]=which.max(ai$betas[j,])
}

abeta <- FrealArchetypes(X,huge = 200, K, ini_arch,PM)


#Convertir en funciones para dibujar
a4=ai$archetypes #son coeficientes en base de arquetipos

yaf=fd(t(a4),mibspline)
plot(yaf)

#Sacar los alphas de cada individuo para cada arquetipo
data=X
huge=200

n <- ncol(t(data))
x_gvv <- rbind(t(data), rep(huge, n))

zs <- rbind(t(a4), rep(huge, K))
zs <- as.matrix(zs)
alphascce <- matrix(0, nrow = K, ncol = n)

for (j in 1 : n){
  alphascce[, j] = coef(nnls(zs, x_gvv[,j]))
}

names(alphascce) = row.names(grupo)
```

```
96
97    #Determinar a que arquetipo pertenece cada elemento
98    arquetipo1 = c()
99    arquetipo2 = c()
100   arquetipo3 = c()
101   arquetipo4 = c()
102   arquetipo5 = c()
103
104   for( j in 1:dim(alphascce)[2] ){
105     maximo = alphascce[1,j]
106     g = 1
107     for( i in 2:dim(alphascce)[1] ){
108       if( alphascce[i,j] > maximo ){
109         maximo = alphascce[i,j]
110         g = i
111       }
112     }
113     if(g == 1){
114       arquetipo1 = c(arquetipo1, row.names(grupo)[j])
115     }
116     else if(g == 2){
117       arquetipo2 = c(arquetipo2, row.names(grupo)[j])
118     }
119     else if(g == 3){
120       arquetipo3 = c(arquetipo3, row.names(grupo)[j])
121     }
122     else if(g == 4){
123       arquetipo4 = c(arquetipo4, row.names(grupo)[j])
124     }
125     else {
126       arquetipo5 = c(arquetipo5, row.names(grupo)[j])
127     }
128   }
129   print(estaciones[f])
130   print("ARQUETIPO 1")
131   print(arquetipo1)
132   print("ARQUETIPO 2")
133   print(arquetipo2)
134   print("ARQUETIPO 3")
135   print(arquetipo3)
136   print("ARQUETIPO 4")
137   print(arquetipo4)
138   print("ARQUETIPO 5")
139   print(arquetipo5)
140
141 }
```