



Universidad Autónoma de Querétaro

Facultad de Ingeniería

Análisis de Datos Funcionales

TESIS

Que como requisito para obtener el título de

Licenciada en Matemáticas Aplicadas

Presenta:

María Guzmán Martínez

Dirigida por:

Dr. Eduardo Castaño Tostado

Santiago de Querétaro, Qro. Octubre de 2008.

BIBLIOTECA CENTRAL, U.A.Q.

No. Adq. H74242

No. Título _____

Clas TS

001.422

G993a

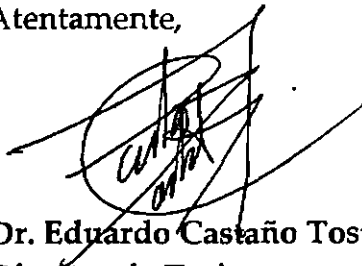
Centro Universitario., 14 de Noviembre de 2008

Dr. Gilberto Herrera Ruíz
Director de la Facultad de Ingeniería
Presente:

Por este conducto, me permito comunicar a Usted, que una vez revisada la tesis individual titulada "Análisis de Datos Funcionales", de la pasante de la licenciatura en Matemáticas Aplicadas la **C. María Guzmán Martínez**, de acuerdo al artículo 20 del inciso h) del reglamento de Titulación vigente.

Emito mi Voto Aprobatorio.

Atentamente,

A handwritten signature in black ink, appearing to be 'Eduardo Castaño Tostado', is written over a circular stamp. The signature is somewhat stylized and overlaps the stamp.

Dr. Eduardo Castaño Tostado
Director de Tesis

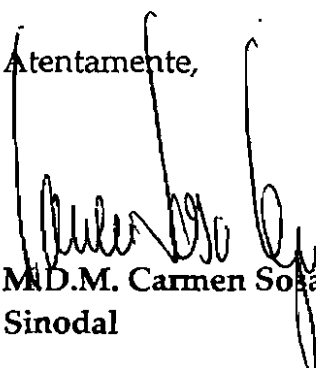
Centro Universitario., 14 de Noviembre de 2008

Dr. Gilberto Herrera Ruíz
Director de la Facultad de Ingeniería
Presente:

Por este conducto, me permito comunicar a Usted, que una vez revisada la tesis individual titulada "Análisis de Datos Funcionales", de la pasante de la licenciatura en Matemáticas Aplicadas la C. María Guzmán Martínez, de acuerdo al artículo 20 del inciso h) del reglamento de Titulación vigente.

Emito mi Voto Aprobatorio.

Atentamente,



M.D.M. Carmen Soledad Garza
Sinodal

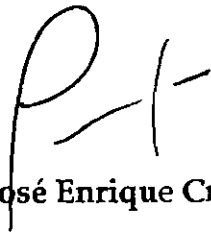
Centro Universitario., 14 de Noviembre de 2008

Dr. Gilberto Herrera Ruíz
Director de la Facultad de Ingeniería
Presente:

Por este conducto, me permito comunicar a Usted, que una vez revisada la tesis individual titulada "Análisis de Datos Funcionales", de la pasante de la licenciatura en Matemáticas Aplicadas la **C. María Guzmán Martínez**, de acuerdo al artículo 20 del inciso h) del reglamento de Titulación vigente.

Emito mi Voto Aprobatorio.

Atentamente,



M. en C. José Enrique Crespo Baltar
Sinodal

Voto Aprobatorio

C.U. 14 de Noviembre de 2008

Nombre

PRESENTE

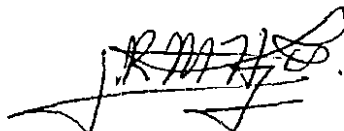
Por este conducto comunico a usted que he revisado el trabajo de Tesis titulado:

Análisis de Datos Funcionales

del (la) alumno(a): María Guzmán Martínez
de la licenciatura en Matemáticas Aplicadas

habiéndolo encontrado satisfactorio, por lo cual doy mi voto aprobatorio.

ATENTAMENTE



Rocio Minerva Hidalgo Flores
Nombre y Firma

Presidente, Secretario, Vocal, Suplente, Suplente

C. U. 15 de octubre de 2007

C. MARÍA GUZMAN MARTÍNEZ

Pasante de la Licenciatura en Matemáticas Aplicadas

Presente.

Con relación a su oficio enviada al H. Consejo Académico de la Facultad en el que solicita titularse bajo/la opción de tesis individual, me permito informarle que en la sesión ordinaria del 15 de octubre del año en curso, este cuerpo colegiado acordó aceptar la opción de titulación por lo que deberá trabajar en el tema **"Análisis de datos funcionales"** bajo la dirección del Dr. Eduardo Castañeda Astudillo.

El Contenido aceptado por el H. Consejo Académico es el siguiente:

1. Introducción

2. Conceptos Básicos

- 2.1. Ejemplos de datos funcionales
- 2.2. Representación de funciones con bases Spline y Wavelets
 - 2.2.1. Spline
 - 2.2.2. Ondeletas (Wavelets)

3. Suavizamiento de datos funcionales

- 3.1. Suavizamiento de funciones por el método de mínimos cuadrados
- 3.2. Suavizamiento de funciones por penalización de sus rugosidades
- 3.3. Método de validación cruzada para escoger el parámetro de suavizamiento

4. Modelos lineales funcionales

- 4.1. Modelo lineal funcional para respuesta funcional y covariables escalares
 - 4.1.1. Modelo de Ramsay y Silverman (2005)
 - 4.1.2. Modelo de Abramovich et al. (2004)
 - 4.1.3. Significancia Estadística
 - 4.1.4. Aplicación
- 4.2. Modelo lineal funcional para respuesta funcional y covariables funcionales
 - 4.2.1. Modelo general
 - 4.2.2. Intervalos de confianza para los parámetros funcionales de la regresión
 - 4.2.3. Modelo funcional lineal completo
 - 4.2.4. Aplicación al conjunto de datos de la temperatura global de la tierra
- 4.3. Modelo lineal funcional para respuesta escalar y covariables funcionales
 - 4.3.1. Modelo general
 - 4.3.2. Intervalos de confianza para los parámetros funcionales de la regresión

Universidad Autónoma de Querétaro
Facultad de Ingeniería
Dirección

4.3.3. Aplicación al conjunto de datos de la temperatura global de la tierra

5. Ecuaciones diferenciales ordinarias en el análisis de datos funcionales

5.1. Ecuaciones diferenciales con coeficientes constantes para datos funcionales

5.2. Ecuaciones diferenciales con coeficientes variables para datos funcionales

5.3. Estimación de operadores diferenciales en el análisis de datos funcional

5.4. Aplicaciones

5.4.1. Conjunto de datos que representan la pérdida de peso de en ajos mexicanos

5.4.2. Datos del Indicador Global de la Actividad Económica Mexicana (1993-2005)

6. Conclusiones

A. Código base en Splus para las aplicaciones

B. Funciones de SPLUS

C. Bibliografía

También hago de su conocimiento las disposiciones de nuestra Facultad, en el sentido que antes del Examen profesional deberá cumplir con los requisitos de nuestra legislación y deberá imprimir el presente oficio en todos los ejemplares de su tesis.

Atentamente

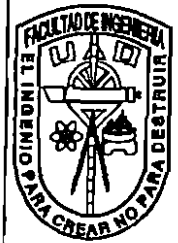
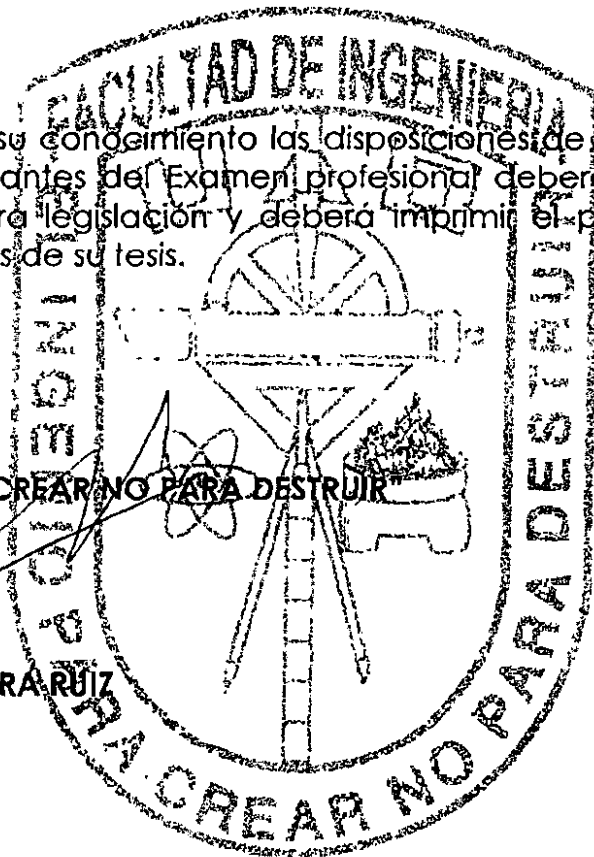
"EL INGENIO PARA CREAR NO PARA DESTRUIR"

DR. GILBERTO HERRERA RUIZ

Director

c.c.p. Archivo

*GHR/DHM.



AGRADECIMIENTOS

Agradezco a Dios, por la vida que me dio regalándome a mis padres. Agradezco a la vida por la oportunidad que me ha dado de conocer personas que me llevo en el alma.

Le agradezco al Dr. Castaño, por el conocimiento que me ha dado, sin recibir beneficio; muchas Gracias maestro.

Desde luego también doy las gracias a todos mis maestros de la Licenciatura en Matemáticas Aplicadas.

Índice general

| | |
|--|-----------|
| Índice de figuras | V |
| 1. Introducción | 1 |
| 2. Conceptos Básicos | 4 |
| 2.1. Ejemplos de datos funcionales..... | 4 |
| 2.2. Representación de funciones con bases spline y wavelet..... | 6 |
| 2.2.1. Spline..... | 6 |
| 2.2.2. Ondeletas (Wavelets)..... | 9 |
| 3. Suavizamiento de datos funcionales | 14 |
| 3.1. Suavizamiento de funciones por el método de mínimos cuadrados..... | 16 |
| 3.2. Suavizamiento de funciones por penalización de sus rugosidades..... | 18 |
| 3.3. Método de validación cruzada para escoger el parámetro de suavizamiento..... | 20 |
| 4. Modelos lineales funcionales | 24 |
| 4.1. Modelo lineal funcional para respuesta funcional y covariables escalares..... | 25 |
| 4.1.1. Modelo de Ramsay y Silverman (2005)..... | 25 |
| 4.1.2. Modelo de Abramovich et al. (2004)..... | 30 |
| 4.1.3. Significancia estadística..... | 31 |
| 4.1.4. Aplicación..... | 38 |
| 4.2. Modelo lineal funcional para respuesta funcional y covariables funcionales..... | 50 |
| 4.2.1. Modelo general..... | 50 |
| 4.2.2. Intervalos de confianza para los parámetros funcionales de la regresión..... | 53 |
| 4.2.3. Modelo funcional lineal completo..... | 54 |
| 4.2.4. Aplicación al conjunto de datos de la temperatura global de la tierra..... | 56 |
| 4.3. Modelo lineal funcional para respuesta escalar y covariables funcionales..... | 66 |
| 4.3.1. Modelo general..... | 66 |
| 4.3.2. Intervalos de confianza para los parámetros funcionales de la regresión..... | 67 |
| 4.3.3. Aplicación al conjunto de datos de la temperatura global de la tierra..... | 68 |

| | |
|--|------------|
| 5. Ecuaciones diferenciales ordinarias en el análisis de datos funcionales | 74 |
| 5.1. Ecuaciones diferenciales con coeficientes constantes para datos funcionales..... | 75 |
| 5.2. Ecuaciones diferenciales con coeficientes variables para datos funcionales..... | 77 |
| 5.3. Estimación de operadores diferenciales en el análisis de datos funcionales..... | 78 |
| 5.4. Aplicaciones..... | 89 |
| 5.4.1. Conjunto de datos que representan la pérdida de peso en ajos mexicanos..... | 89 |
| 5.4.2. datos del Indicador Global de la Actividad Económica Mexicana (1993-2005)..... | 96 |
| 6. Conclusiones | 101 |
| A. Código base en Splus para las aplicaciones | 103 |
| B. Funciones de Splus | 149 |
| Bibliografía | 157 |

Índice de figuras

| | | |
|------|--|----|
| 1.1 | Datos del <i>IGAEM</i> | 2 |
| 2.1 | Regresión Lineal..... | 5 |
| 2.2 | Interpolación lineal..... | 5 |
| 2.3 | Velocidad del <i>IGAEM</i> | 9 |
| 2.4 | Partición de $L_2(\mathbb{R})$ | 12 |
| 3.1 | Ajuste funcional del conjunto de los datos del <i>IGAEM</i> | 23 |
| 4.1 | Datos escalares de las respuestas..... | 40 |
| 4.2 | Datos escalares de las respuestas..... | 40 |
| 4.3 | 16 respuestas funcionales suavizadas..... | 41 |
| 4.4 | Coefficientes funcionales estimados..... | 41 |
| 4.5 | Coefficientes funcionales del modelo..... | 42 |
| 4.6 | Respuestas funcionales estimadas..... | 42 |
| 4.7 | Desviación estándar de los residuales..... | 43 |
| 4.8 | Intervalos de confianza para los coeficientes funcionales..... | 44 |
| 4.9 | Coefficientes funcionales con sus intervalos de confianza..... | 44 |
| 4.10 | Usando sólo 16 observaciones de cada respuesta y_i | 46 |
| 4.11 | Coefficientes de la transformación discreta wavelet de $y_i(t)$ | 47 |
| 4.12 | Valores de la temperatura global anual del 1998 al 2003..... | 57 |
| 4.13 | Dióxido de carbono global anual del 1998 al 2003..... | 57 |
| 4.14 | Coefficiente funcional de la regresión..... | 58 |
| 4.15 | Gráfica de $\hat{y}(t)$ estimada de las 16 observaciones..... | 58 |
| 4.16 | Gráfica de los residuales..... | 59 |
| 4.17 | Error estándar del parámetro funcional $\hat{\beta}(t)$ | 59 |
| 4.18 | Intervalo del parámetro funcional $\hat{\beta}(t)$ | 60 |
| 4.19 | Funciones estimadas del CO_2 | 61 |
| 4.20 | Datos escalares de las temperaturas globales mensuales de la tierra y el océano..... | 62 |
| 4.21 | 16 funciones $\hat{y}_i(t)$ de las temperaturas globales mensuales de la tierra y el océano..... | 62 |

| | | |
|------|---|-----|
| 4.22 | Término funcional independiente del modelo de regresión..... | 63 |
| 4.23 | Función bivariada de la regresión..... | 64 |
| 4.24 | Temperaturas globales mensuales funcionales estimadas..... | 64 |
| 4.25 | Funciones \hat{y}_i vs sus promedios funcionales y el valor del R^2 | 65 |
| 4.26 | Funciones \hat{y}_i vs sus promedios funcionales y el valor del R^2 | 65 |
| 4.27 | Promedios de las temperaturas globales anuales..... | 69 |
| 4.28 | Dióxido de carbono emitido durante los años 1998-2003..... | 70 |
| 4.29 | Representación funcional del CO_2 | 70 |
| 4.30 | Coefficiente funcional estimado $\beta(t)$ | 71 |
| 4.31 | Recta de regresión ajustada a los datos y_j | 72 |
| 4.32 | Intervalo puntual del parámetro funcional $\beta(t)$ | 73 |
| 5.1 | 18 curvas de los valores escalares..... | 90 |
| 5.2 | Los seis tratamientos con sus tres réplicas..... | 91 |
| 5.3 | Réplicas funcionales del tratamiento seis..... | 92 |
| 5.4 | Parámetros funcionales estimados para el tratamiento seis..... | 92 |
| 5.5 | Soluciones estimadas para el tratamiento seis..... | 93 |
| 5.6 | Ajuste de la función $\hat{y}_1(t)$ para la primer réplica..... | 94 |
| 5.7 | Ajuste de la función $\hat{y}_2(t)$ para la segunda réplica..... | 95 |
| 5.8 | Ajuste de la función $\hat{y}_3(t)$ para la tercer réplica..... | 95 |
| 5.9 | 156 datos del <i>IGAEM</i> | 97 |
| 5.10 | Función $y(t)$ estimada del <i>IGAEM</i> | 98 |
| 5.11 | Parámetros funcionales del <i>IGAEM</i> | 98 |
| 5.12 | Las tres soluciones $\hat{\beta}_i(t)$ del operador diferencial lineal planteado..... | 99 |
| 5.13 | Residual funcional $f(t)$ | 100 |
| 5.14 | Modelo diferencial funcional estimado vs los datos del <i>IGAEM</i> | 100 |

Resumen

En el contexto estadístico, existen conjuntos de datos que pueden ser modelados por medio de una función que depende de un tiempo t , si esto es posible podemos hacer un análisis más rico de los datos al explicar mejor la variabilidad de estos datos. Nuestro objetivo es encontrar la función de los datos que pueden ser representados funcionalmente, para después trabajar con esta función en el contexto estadístico, pero desde un aspecto funcional.

Este trabajo aplica el análisis estadístico para conjuntos de datos desde una perspectiva funcional, es decir ahora más que analizar datos, analizamos funciones que están implícitas en los datos. Además La ventaja desde el punto funcional, es que estas funciones dependen de un argumento t , que generalmente es el tiempo en segundos días o años. En el análisis estadístico funcional se utiliza la teoría de estadística multivariada como punto de partida para generalizar después a la teoría de estadística funcional de datos. El hecho de poder explicar el comportamiento de un fenómeno reflejado en un conjunto de datos, por medio de una función, le da un plus al análisis estadístico de los datos.

Dado que se trabaja con funciones es importante revisar la teoría de las funciones, algunas de éstas son que sean continuas, suaves, diferenciales, etc.

En este análisis estadístico además de trabajar con teoría funcional, se trabaja con teoría de ecuaciones diferenciales ordinarias además y obviamente con teoría estadística multivariada.

Una vez que se haya encontrado la función correcta que modele nuestros datos, será necesario hacer el análisis estadístico, el cual consiste en verificar que se cumplan los supuestos del modelo funcional que se esté utilizando para analizar los datos. Usando la teoría de ecuaciones diferenciales se encuentran también modelos diferenciales funcionales para conjuntos de datos con demasiada variabilidad, o simplemente para conjuntos de datos en donde nos interesa el comportamiento de sus derivadas y su interpretación.

Como gran parte del trabajo se apoya en el libro de J.O. Ramsay y B.W. Silverman nuestra metodología consiste en estudiar el libro de este autor, analizar los ejemplos que propone en su libro con el paquete Splus para análisis estadísticos, para luego buscar nuestras propias aplicaciones con datos de nuestro interés.

Capítulo 1

Introducción

En la actualidad muchos análisis de datos provenientes tanto de experimentos como de estudios observacionales generan datos para una misma unidad (experimental u observacional) que pudiere presumir que son manifestaciones discretas de una función subyacente,

$$[U : y_1 \dots y_n]$$

es decir, para una unidad U se puede pensar que sus datos provienen de una función $y(t)$, $t \in \mathbb{R}$; así $\{y_j\}_{j=1}^n$ se llaman datos funcionales.

Un objetivo de la estadística, en general, es entender cuáles son las fuentes que hacen diferentes a las unidades en sus respuestas. En el contexto de datos funcionales tal objetivo debe resolverse utilizando plenamente las propiedades funcionales de los datos observados. A esta área de la estadística se le denominará Análisis de Datos Funcionales (*ADF*) (Ramsay y Silverman, 2005). El Análisis de Datos Funcionales tiene amplias aplicaciones

en la economía, la medicina, la industria, entre muchos otros campos laborables.

Los objetivos en un *ADF* son:

- a) Representar los datos de una manera funcional adecuada para un análisis posterior.
- b) Mostrar los datos de tal forma que se resalten las características importantes de éstos.
- c) Estudiar las fuentes de variación de los datos.
- d) Estudiar relaciones entre variables dependientes y variables independientes.

Generalmente los conjuntos de datos que son funcionales tienen patrones difíciles de modelar, por ejemplo como es el caso del conjunto de 156 datos del Indicador Global de la Actividad Económica Mexicana (*IGAEM*) (Fuente: INEGI), cuya gráfica se muestra en la Figura 1.1.

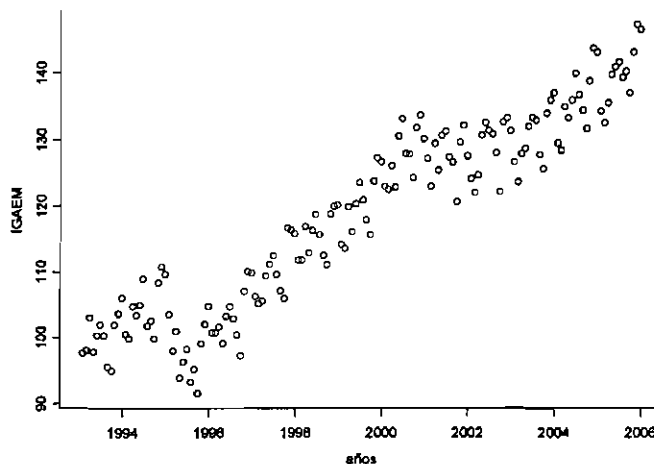


Figura 1.1: Datos del *IGAEM*.

La función $y(t)$ que subyace a los datos debe ser estimada. Este trabajo aplica el análisis estadístico a conjuntos de datos desde una perspectiva funcional, es decir, ahora

más que analizar datos como escalares, analizamos funciones que están implícitas en los datos.

El hecho de poder explicar el comportamiento de un fenómeno reflejado en un conjunto de datos por medio de una función suave, le da valor agregado al análisis estadístico de los datos: el uso de derivadas de las funciones en un contexto estadístico.

Capítulo 2

Conceptos Básicos

2.1. Ejemplos de datos funcionales

El objetivo inicial en el *ADF* es ajustar una función $y(t)$ a un conjunto de datos discretos y_1, \dots, y_n , que refleje una variación suave y que pudiera ser evaluada, en principio, tan frecuentemente como se desee en un intervalo $t \in [a, b]$. La pregunta obligada es qué función ajustar a y_1, \dots, y_n y cómo debe de ser definida.

Empezando por lo más simple, supongamos que tenemos un conjunto de datos discretos y_1, \dots, y_6 en el intervalo $[-30, 30]$ y se hace una regresión lineal, lo que se tiene entonces es un polinomio de primer grado que intenta aproximarse a los datos discretos, tal y como se muestra en la Figura 2.1.

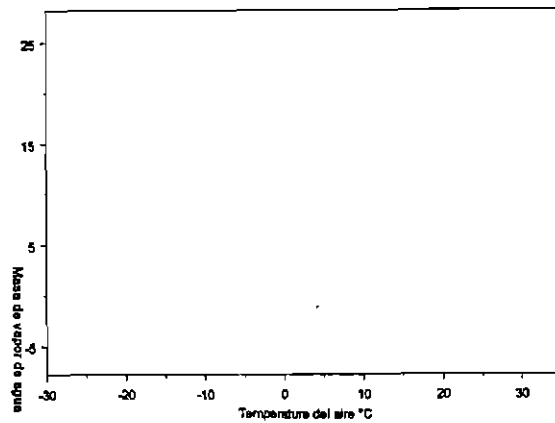


Figura 2.1: Regresión lineal.

Este polinomio puede a lo más minimizar para algunos y_j los residuales $e_j = (y_j - \hat{y}_j)$, que son los errores de la predicción hecha por el polinomio de primer grado. Pero si lo que se quiere es minimizar cada uno de los e_j , entonces podemos hacer una interpolación de tal manera que todos los y_j caigan sobre en la curva estimada, tal y como se muestra en la Figura 2.2.

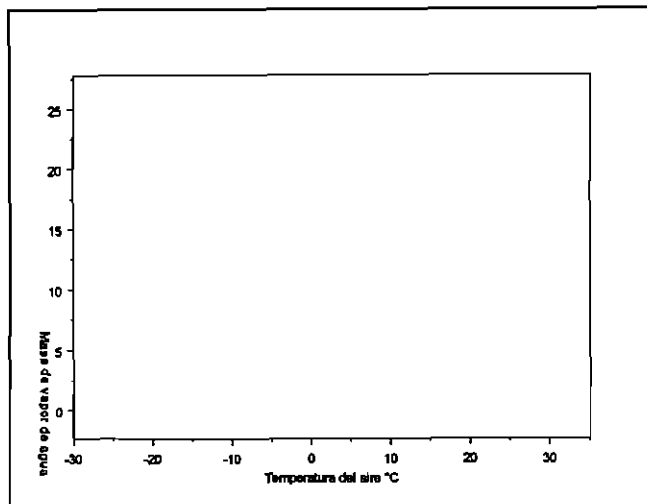


Figura 2.2: Interpolación lineal.

Claramente se ve que estos datos son aproximados por segmentos de línea en el

intervalo $[-30, 30]$, y esto es precisamente lo que en análisis de datos funcionales se trata de evitar. Hay distintas maneras de arreglar esto.

2.2. Representación de funciones con bases spline y wavelets

Se desea entonces, para un conjunto de datos discretos y_1, \dots, y_n , estimar $y(t)$ subyacente. Una forma de encontrar una representación de $y(t)$ es a partir de una base de funciones, es decir,

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

donde $\{\phi_k(t)\}$ es un conjunto de funciones base y $\{c_k\}$ son los coeficientes a ser estimados a partir de $\{y_1, \dots, y_n\}$, que corresponden a $y(t)$ en tal base. Bases hay muchas, *spline*, *wavelets*, *exponenciales*, *Fourier*, etc. En este trabajo nos centramos en las bases tipo *spline* y tipo *wavelet*.

2.2.1. Spline

Los segmentos de polinomios de grado m que determinan un *spline*, satisfacen algunas condiciones (Sánchez et al. 1996):

Definition 1 Para todos los polinomios P de grado menor o igual que m , es decir

$$P_m = \{p : \mathbb{R} \rightarrow \mathbb{R} \ni D^{(m+1)}p(t) = 0\}$$

$\dim(P_m) = m + 1$ y cada polinomio $P \in P_m$, se puede escribir como

$$p(t) = \alpha_1 p_1(t) + \alpha_2 p_2(t) + \dots + \alpha_m p_m(t) + \alpha_{m+1} p_{m+1}(t)$$

donde $\{p_1(t), p_2(t), \dots, p_{m+1}(t)\}$ es un conjunto de polinomios independientes de grado menor o igual que m , que opera como una base para expresar cualquier $p(t) \in P_m$.

La familia de polinomios es muy útil para resolver problemas de aproximación y ello se debe entre otras razones a la siguiente propiedad fundamental (Sánchez et al. 1996):

Proposition 2 Dado $n+1$ puntos distintos (t_j, y_j) existe un único polinomio $p(t)$ de grado menor o igual que m tal que

$$P(t_j) = y_j, \quad j = 1, \dots, n$$

si $t_i \neq t_j$.

Una de las bases más usadas es la *spline*. Conocer los diferentes tipos de bases que existen para determinar la función $y(t)$ no es suficiente para elegir la función que mejor explique la variabilidad de los puntos y_1, \dots, y_n . De hecho si se decide generar a $y(t)$ por bases *spline*, las preguntas obligadas son: ¿cómo se genera la función $\phi(t)$? y ¿cómo se determina la mejor función $y(t)$ para el conjunto de puntos y_1, \dots, y_n ?

A continuación mostramos algo de teoría para funciones *spline*, es decir, para funciones generadas por bases *spline* de orden m , lo cual implica trabajar con piezas de polinomios de grado $m - 1$, y estas piezas de polinomios tienen derivadas continuas hasta orden $m - 2$. Por lo general siempre se trabaja con funciones *spline* de orden cuatro.

Supongamos que tenemos los números reales t_1, \dots, t_n en algún intervalo $[a, b]$ tal que $a < t_1 < t_2 < \dots < t_n < b$ donde $t_0 = a$ y $t_{n+1} = b$; una función $y(t)$ definida sobre $[a, b]$ es un *spline* cúbico si se satisfacen dos condiciones:

1. En cada uno de los intervalos $(a, t_1), (t_1, t_2), \dots, (t_n, b)$ $y(t)$ es un polinomio cúbico.
2. Las piezas de polinomios se ajustan a los puntos t_j de tal forma que $y(t)$ y su primera y segunda derivadas sean continuas en cada t_j y por lo tanto en todo el intervalo $[a, b]$, los puntos t_j son llamados *nodos*.

Una forma de expresar un *spline* cúbico es dar los cuatro coeficientes polinomiales c_i , tal que

$$y(t) = c_4(t - t_1)(t - t_2)(t - t_3) + c_3(t - t_1)(t - t_2) + c_2(t - t_1) + c_1$$

$$y(t) = c_4p_1(t) + c_3p_2(t) + c_2p_3(t) + c_1.$$

Un *spline* cúbico en un intervalo $[a, b]$ se dice que es un *spline* cúbico natural, si su segunda derivada y tercera derivada son cero en a y b . Estas condiciones son llamadas condiciones naturales de frontera. Esto implica que $y(t)$ sea lineal en los dos extremos del intervalo $[a, t_1]$ y $[t_n, b]$.

Dado que tenemos un conjunto de valores y_1, \dots, y_n para los puntos t_1, \dots, t_n y lo que queremos es encontrar una curva suave $y(t)$ tal que interpole los puntos (t_j, y_j) , es decir $y(t_j) = y_j$ para todo $j = 1, \dots, n$, enunciamos el siguiente teorema (Green y Silverman, 1994):

Theorem 3 Sea $n \geq 2$ tal que $t_1 < \dots < t_n$. Entonces dados los valores y_1, \dots, y_n , existe un único *spline* cúbico $y(t)$ con nodos en los puntos t_j que satisface

$$y(t_j) = y_j \text{ para } j = 1, \dots, n.$$

Lo anterior contestaría nuestra primer pregunta de cómo generar $y(t)$. Para dar respuesta a la última pregunta, definimos $\mathcal{S}[a, b]$ de todas las funciones $y(t)$ sobre $[a, b]$ que

tienen dos derivadas continuas, entonces una función es *suave* si está en $\mathcal{S}[a, b]$. Si queremos que $y(t)$ sea *suave* pero que aproxime los puntos dados, se puede determinar a la función $y(t)$ como aquella que tenga el valor más pequeño al calcular $\int [D^2y(t)]^2 dt$ de todas las curvas que aproximen los datos, esto contesta la última pregunta. En el siguiente capítulo explicaremos el significado de $\int [D^2y(t)]^2 dt$.

2.2.2. Ondeletas (Wavelets)

Las bases *wavelet*, son de gran utilidad en análisis estadísticos, pues son útiles en la estimación de funciones cuyo comportamiento no es suave, es decir, la señal tiene cambios en el tiempo con brinco muy bruscos, un ejemplo de este tipo de señales se muestra en la Figura 2.3.

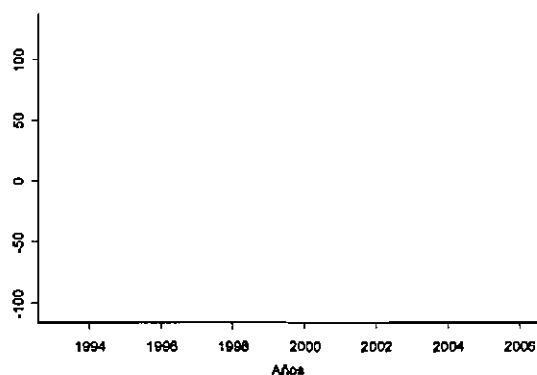


Figura 2.3: Velocidad del IGAEM.

Para estimar la función $y(t)$, que explica el comportamiento del conjunto de datos discretos y_1, \dots, y_n , con bases *wavelets*, primero definimos las funciones

$$\psi_{j,\kappa}(t) = 2^{\frac{1}{2}}\psi(2^j t - \kappa) \text{ y } \phi_{j,\kappa}(t) = 2^{\frac{1}{2}}\phi(2^j t - \kappa) \text{ con } j \text{ fijo, } \kappa \in \mathbb{Z}$$

que son conocidas con los nombres *wavelet mother* ψ (comportamiento de detalle fino) y *wavelet father* ϕ (comportamiento global grueso). La función $\psi_{j,\kappa}(t) = 2^{\frac{1}{2}}\psi(2^j t - \kappa)$ al ser dilatada con j y trasladada con el entero κ genera la base ortonormal $\{\psi_{j,\kappa}(t), j \text{ fijo, } \kappa \in \mathbb{Z}\}$ también la función $\phi_{j,\kappa}(t) = 2^{\frac{1}{2}}\phi(2^j t - \kappa)$ al ser dilatada con j y trasladada con el entero κ genera la base ortonormal $\{\phi_{j,\kappa}(t), j \text{ fijo, } \kappa \in \mathbb{Z}\}$. Cada uno de estos conjuntos $\{\psi_{j,\kappa}(t), j \text{ fijo, } \kappa \in \mathbb{Z}\}$ y $\{\phi_{j,\kappa}(t), j \text{ fijo, } \kappa \in \mathbb{Z}\}$ por ser bases ortonormales tienen la propiedad de generar al espacio $L_2(\mathbb{R})$, donde por definición $L_2(\mathbb{R})$ consiste de todas las funciones cuadradas integrables

$$y(t) \in L_2(\mathbb{R}) \text{ si } \int |f|^2 < \infty.$$

De hecho, dada una secuencia de subespacios cerrados $V_j, j \in \mathbb{Z}$ que viven en una contención jerárquica

$$\dots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \dots$$

se puede definir $L_2(\mathbb{R})$ como la unión cerrada de los subespacios V_j

$$L_2(\mathbb{R}) = \overline{\cup_j V_j}$$

con $\cap_n V_j = \{0\}$. Dado que el conjunto $\{\phi_{j,\kappa}(t), j \text{ fijo, } \kappa \in \mathbb{Z}\}$ es una base ortonormal, se denota con V_j al subespacio que genera esta base; lo anterior se conoce como análisis de multi-resolución (Vidakovic, 1995). De hecho Vidakovic (1995) señala que $\{\psi_{j,\kappa}(t), j \text{ fijo, } \kappa \in \mathbb{Z}\}$

también es una base ortonormal para $L_2(\mathbb{R})$, y se denota con W_j al subespacio que genera esta base. Así en el análisis de multirresolución se tiene $V_j \subset L_2(\mathbb{R})$ y $W_j \subset L_2(\mathbb{R})$, entonces se define $W_j = V_{j+1} \ominus V_j$, es decir, como la diferencia de subespacios V_j y $V_j = V_{j-1} \oplus W_{j-1}$.

Cualquier función $y(t)$ en $L_2(\mathbb{R})$ puede ser representada entonces como

$$y(t) = \sum_{j,\kappa} d_{j,\kappa} \psi_{j,\kappa}(t)$$

donde los $d_{j,\kappa}$ son los coeficientes wavelet, y esta representación única corresponde a la descomposición de multirresolución

$$L_2(\mathbb{R}) = \bigoplus_{j=-\infty}^{\infty} W_j.$$

También para cualquier j_0 fijo se tiene

$$L_2(\mathbb{R}) = V_{j_0} \oplus \left(\bigoplus_{j=j_0}^{\infty} W_j \right)$$

lo que permite tener una descomposición de $y(t)$ en $L_2(\mathbb{R})$ como

$$y(t) = \sum_{\kappa} c_{j_0,\kappa} \phi_{j_0,\kappa}(t) + \sum_{j \geq j_0} \sum_{\kappa} d_{j,\kappa} \psi_{j,\kappa}(t).$$

En un análisis de multirresolución ortonormal, el subespacio V_0 , puede ser representado como

$$V_0 = \bigoplus_{j_0 \leq j < 0} W_j \oplus V_{j_0}$$

para algún $j_0 < 0$.

En otras palabras, W_j es el complemento ortonormal de V_j en V_{j-1} , tal como se muestra en la Figura 2.4 para $j = 3$.

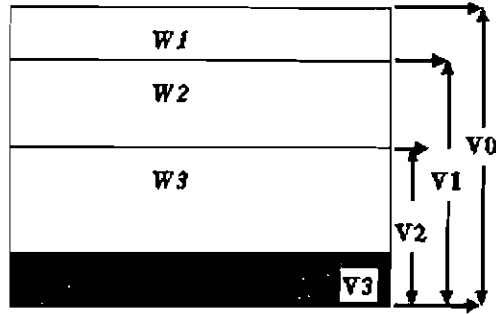


Figura 2.4: Partición de $L_2(\mathbb{R})$.

La función ψ que vive en W_0 puede ser representada en W_j como

$$\psi_{j,\kappa}(t) = \sqrt{2} \sum_{\kappa=0} g_{\kappa} \psi(2t - \kappa),$$

y para definir ψ en V_j se tiene

$$\psi_{j,\kappa}(t) = \sqrt{2} \sum_{\kappa=0} g_{\kappa} \phi(2t - \kappa).$$

Análogamente la función $\phi(t) \in V_0$, puede ser representada en V_j como

$$\phi_{j,\kappa}(t) = \sqrt{2} \sum_{\kappa=0} h_{\kappa} \phi(2t - \kappa),$$

y para definir ϕ en W_j se define

$$\phi_{j,k}(t) = \sqrt{2} \sum_{k=0} g_k \psi(2t - k).$$

De hecho los parámetros g_{κ} y $h_{1-\kappa}$ llamados *quadrature mirror filters* (filtros de espejo de cuadratura), están relacionados por la ecuación

$$g_{\kappa} = (-1)^{\kappa} h_{1-\kappa},$$

conocida con el nombre de *quadrature mirror* (espejo de cuadratura).

La descomposición de $L_2(\mathbb{R})$ en subespacios W_j y V_j se hace con el objetivo de separar la función $y(t)$ en componentes *finos* y *gruesos*, esta descomposición nos permite hacer una estimación de la señal $y(t)$ tomando sólo en cuenta los coeficientes que nos interese de estos dos tipos de componentes.

Entonces, dado un conjunto de datos discretos y_1, \dots, y_n para estimar una función $y(t)$ subyacente, el objetivo será conocer los coeficientes de la descomposición

$$y(t) = \sum_{\kappa} c_{J-1,\kappa} \phi_{J-1,\kappa}(t) + \sum_{\kappa} d_{J-1,\kappa} \psi_{J-1,\kappa}(t) + \sum_{\kappa} d_{J-2,\kappa} \psi_{J-2,\kappa}(t) + \dots + \sum_{\kappa} d_{1,\kappa} \psi_{1,\kappa}(t)$$

$c_{J-1,\kappa}$ y $d_{j,\kappa}$ son los coeficientes wavelets estimados por las integrales

$$c_{J-1,\kappa} \approx \int \phi_{J-1,\kappa} x(t) dt$$

$$d_{j,\kappa} \approx \int \psi_{j,\kappa} x(t) dt \quad j = 0, 2, \dots, J-1, \quad \kappa = 0, \dots, 2^j - 1.$$

La magnitud de estos coeficientes da una medida de la aportación de la correspondiente función *wavelet*. El *wavelet father*, es bueno en representar la parte *suave*, de baja frecuencia, de la función $y(t)$ y la *wavelet mother* es buena en representar la parte de los detalles, de alta frecuencia, de la función $y(t)$. Así a los $c_{J-1,\kappa}$ se les llama coeficientes *suaves* y a los $d_{j,\kappa}$ coeficientes de detalle. Los coeficientes del nivel *grueso*, que son los $c_{J-1,\kappa}$, corresponden a las funciones base que se encuentran en el subespacio V_j , y los coeficientes de *detalle*, que son los $d_{j,\kappa}$, corresponden a las funciones base en el subespacio W_j .

Existen varios tipos de familias *wavelets*, generadas por su respectiva *wavelet mother*, entre ellas se encuentran la familia *Haar*, *Shannon*, *Meyer*, *Franklin*, *Daubechies*, etc. Cada una de estas familias ofrece propiedades inherentes, que son un tanto heredadas por la *wavelet mother*, que las genera. Algunas de estas bases ortonormales, son continuas como la *Daubechies* o simétricas como la *Haar* (Vidakovic, 1995).

Capítulo 3

Suavizamiento de datos funcionales

El presente capítulo es un resumen de Ramsay y Silverman (2005) de los capítulos cuatro y cinco.

¿Qué es el suavizamiento de datos funcionales? El suavizamiento de datos funcionales se da cuando estamos interesados en construir una función *suave* que aproxime un conjunto de datos, sin que necesariamente los interpole. El problema de suavizar los datos (t_j, y_j) , se puede formular entonces, como el construir una función $y(t)$ *suave* que no sólo se aproxime a los puntos (t_j, y_j) , $j = 1, \dots, n$, si no que además esta función $y(t)$ sea capaz de explicar la variabilidad inherente de los datos en el intervalo donde están contenidos, de acuerdo a los objetivos del análisis.

Dado que una función en el análisis funcional está dada por la representación

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

entonces el concepto de suavizamiento tiene que ver con el número de derivadas de $\phi_k(t)$.

Generalmente cuando la función $y(t)$ es una función *spline* de orden cuatro, ésta se considera *suave* si las piezas de polinomios cúbicos que la definen son continuas y diferenciablemente continuas hasta la segunda derivada. El concepto de *suavidad* de una función $y(t)$ es muy sutil, pues para lo que unos es *suave* para otros es *rugoso* dependiendo de los objetivos del análisis. Es por esto que necesitamos trabajar sobre este concepto.

Cuando se tiene la función

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

se puede escoger $K = n$, donde n es el número de observaciones; si se hiciera esto lo que se tendría sería una interpolación exacta de los datos. En ocasiones esto no es lo mejor, pues trae como consecuencia una mala estimación, por ejemplo si $K = n$ entonces la primer derivada de la función $y(t)$

$$D\hat{y}(t) = \sum_{k=1}^K \hat{c}_k D\phi_k(t) = \hat{c}D\phi(t)$$

tendería a oscilar más de lo normal y por esta razón se recomienda tener cuidado con el número de funciones base que se pretende usar. Un ejemplo de esto lo tenemos con la primer derivada de la función $y(t)$ del conjunto de datos del *IGAEM*. Para este conjunto se cuenta con 156 observaciones, haciendo $K = 156$, la función $Dy(t)$ se muestra en la Figura 2.3, ésta muestra más ruido que la variabilidad de la velocidad del *IGAEM*.

A continuación mostramos dos métodos para suavizar una función $y(t)$: el método de mínimos cuadrados y el método que incluye la penalización de la rugosidad de $y(t)$.

3.1. Suavizamiento de funciones por el método de mínimos cuadrados

Nuestro objetivo es modelar los datos discretos y_j , $j = 1, \dots, n$ usando el modelo

$$y_j = y(t_j) + \epsilon_j$$

en donde estamos usando expansiones de funciones base *spline* para $y(t)$; un suavizamiento lineal es obtenido cuando determinamos los coeficientes c_k de la expansión por el criterio de mínimos cuadrados (*SM*).

$$\min_{\mathbf{c}} SMSSE(\mathbf{y} | \mathbf{c}) = \sum_{j=1}^n [y_j - \sum_k^K c_k \phi_k(t_j)]^2$$

que en notación matricial equivale a

$$\min_{\mathbf{c}} SMSSE(\mathbf{y} | \mathbf{c}) = (\mathbf{y} - \Phi \mathbf{c})' (\mathbf{y} - \Phi \mathbf{c}) = \|\mathbf{y} - \Phi \mathbf{c}\|^2.$$

Derivando $SMSSE(\mathbf{y} | \mathbf{c})$ con respecto a \mathbf{c} e igualando a cero, se obtiene

$$2\Phi\Phi'\mathbf{c} - 2\Phi'\mathbf{y} = 0$$

luego resolviendo para \mathbf{c}

$$\hat{\mathbf{c}} = (\Phi'\Phi)^{-1}\Phi'\mathbf{y}$$

entonces el vector de valores estimados es

$$\hat{y}(t) = \hat{\mathbf{y}} = \Phi\hat{\mathbf{c}} = \Phi(\Phi'\Phi)^{-1}\mathbf{y}$$

La estimación por mínimos cuadrados es adecuada cuando se satisfacen los supuestos de que los ϵ_j son independientes, con media cero y varianza constante. Por lo general en el análisis

de datos funcionales los ϵ_j no cumplen con estos supuestos porque no son estacionarios y/o están correlacionados. Entonces será necesario tratar de manera diferente los pesos de los residuales, y esto se logra de la siguiente forma

$$\min_{\mathbf{c}} SMSSSE(\mathbf{y} | \mathbf{c}) = (\mathbf{y} - \Phi\mathbf{c})'\mathbf{W}(\mathbf{y} - \Phi\mathbf{c}) \quad (3.1)$$

donde \mathbf{W} es una matriz simétrica definida positiva que permite dar un peso diferente para los cuadrados y productos de residuales. La matriz $\mathbf{W} = \sum_{\epsilon}^{-1}$ es la matriz de varianza y covarianza de los residuales ϵ_j . En aplicaciones donde una estimación completa de \sum_{ϵ} no está disponible, la covarianza entre residuales se iguala a cero, haciendo así a \sum_{ϵ} una matriz diagonal. Entonces el criterio de mínimos cuadrados ponderados estima el vector \mathbf{c} de coeficientes como

$$\hat{\mathbf{c}} = (\Phi'\mathbf{W}\Phi)^{-1}\Phi'\mathbf{W}\mathbf{y}.$$

El criterio de mínimos cuadrados para suavizar los datos, tiene la propiedad de ser lineal, ya que se puede representar a $\hat{y}(t)$ como

$$\hat{y}(t) = \mathbf{S}\mathbf{y},$$

donde \mathbf{S} es una matriz de suavizamiento, que en el caso del criterio de mínimos cuadrados no ponderados se tiene

$$\mathbf{S} = \Phi(\Phi'\Phi)^{-1}\Phi',$$

mientras que la matriz de suavizamiento para el caso de mínimos cuadrados ponderados es

$$\mathbf{S} = \Phi(\Phi'\mathbf{W}\Phi)^{-1}\Phi\mathbf{W}.$$

3.2. Suavizamiento de funciones por penalización de sus rugosidades

El suavizamiento de una función se puede dar cuando penalizamos la rugosidad de ésta. El significado de suavizamiento es expresado implícitamente en el nivel del criterio a ser optimizado y no al número de funciones base a ser utilizado.

Cuando tenemos una función $y(t)$ por estimar a partir valores $y_j = y(t_j) + \epsilon_j$, generalmente deseamos que $y(t)$ sea *suave* tomando en cuenta dos objetivos:

1. Asegurar que la curva $y(t)$ estimada, se ajuste bien a los datos haciendo que la suma de cuadrados de los residuales $\sum [y_j - y(t_j)]$ sea pequeño.
2. No deseamos que el ajuste sea demasiado perfecto, pues tendríamos una curva $y(t)$ demasiado variable y por consiguiente problemas con sus derivadas.

Si definimos el Error Cuadrado Medio como (Ramsay y Silverman, 2005)

$$\begin{aligned} MSE &= \text{Error Cuadrado Medio} = (\text{sesgo})^2 + \text{Varianza muestral} \\ &= [y(t) - E[\hat{y}(t)]]^2 + E \left[\{\hat{y}(t) - E[\hat{y}(t)]\}^2 \right] \end{aligned}$$

entonces podemos reducir MSE sacrificando algún sesgo para reducir la varianza y de esta forma suavizar la curva estimada. Permitiendo así que la estimación varíe suavemente de dato a dato, aunque con ello estaríamos perdiendo información entre los datos.

¿Cómo cuantificar la rugosidad de una función? El cuadrado de la segunda derivada de la función $y(t)$ nos ayuda en este sentido. El término $[D^2y(t)]$ es llamado frecuentemente la curvatura en t , y se toma como una medida natural de la rugosidad de la función $y(t)$, pensando entonces estimar $y(t)$ atendiendo a su ajuste a los datos pero penalizando la

pérdida de suavidad, entonces $D^2y(t)$ se puede pensar como una función de penalización

$$PEN_2(y(t)) = \int [D^2y(t)]^2 dt$$

Una función altamente variable, se espera que tenga valores grandes para $PEN_2(y(t))$, ya que su segunda derivada puede ser grande en algún lugar del intervalo de nuestro interés.

En general podemos definir la penalización como

$$PEN_m(y(t)) = \int [D^m y(t)]^2 dt.$$

ahora ya podemos modificar el criterio de mínimos cuadrados, con lo que el criterio es

$$\min_{\mathbf{c}} PENSSE_{\lambda}(y | \mathbf{y}) = [\mathbf{y} - \mathbf{y}(t)]' \mathbf{W} [\mathbf{y} - \mathbf{y}(t)] + \lambda \times PEN_2(y),$$

entonces la función $y(t)$ que minimiza $PENSSE_{\lambda}(y)$ es escogida sobre el espacio de funciones $y(t)$ para el cual $PEN_2(y)$ está definida. El parámetro λ , es el *parámetro de suavizado* que representa qué tan importante es el grado de suavidad de $y(t)$, cuando $\lambda \rightarrow \infty$ la función $y(t)$ tiende a ser una línea recta, mientras que cuando $\lambda \rightarrow 0$, la función $y(t)$ tiende a ser una interpolación para los datos, es decir, $y(t_j) = y_j$ para todo j .

Cuando estamos penalizando la rugosidad de la función $y(t)$, entonces en lugar de (3.1), tenemos

$$PENSSE_m(\mathbf{y} | \mathbf{c}) = [\mathbf{y} - \Phi \mathbf{c}]' \mathbf{W} [\mathbf{y} - \Phi \mathbf{c}] + \lambda \mathbf{c}' \mathbf{R} \mathbf{c}$$

donde $\mathbf{R} = \int D^m \phi(t) D^m \phi'(s) dt$, es decir

$$PEN_m(y) = \int [D^m y(t)]^2 dt = \mathbf{c}' \mathbf{R} \mathbf{c}.$$

Tomando la derivada de $PENSSE_m(y | c)$ con respecto de c e igualando a cero, y resolviendo para éste, tenemos

$$\hat{c} = (\Phi'W\Phi + \lambda R)^{-1}\Phi'Wy$$

entonces el correspondiente vector de valores estimados para los datos es

$$\hat{y}(t) = \hat{y} = \Phi(\Phi'W\Phi + \lambda R)^{-1}\Phi'Wy.$$

Si definimos la matriz *sombbrero* $S_{\phi,\lambda}$ de orden n como

$$S_{\phi,\lambda} = \Phi(\Phi'W\Phi + \lambda R)^{-1}\Phi'W \quad (3.2)$$

resulta ser

$$\hat{y}(t) = S_{\phi,\lambda}y.$$

3.3. Método de validación cruzada para escoger el parámetro de suavizamiento

A la técnica algorítmica para escoger el parámetro de suavizamiento λ se le llama validación cruzada (Green y Silverman, 1994), y consiste básicamente de la siguiente idea: dado el conjunto de datos discretos y_1, \dots, y_n , definidos en el intervalo $[a, b]$; quitamos una observación y_j de la muestra, ajustamos $y(t)$ al resto de la muestra la cual es ahora de tamaño $n - 1$ y fijamos un intervalo $[c, d]$ tal que $c \leq \lambda \leq d$. La observación y_j recibe el nombre de *muestra de validación* y la muestra de tamaño $n - 1$ recibe el nombre de *muestra de entrenamiento*. Sea $\hat{y}(t; \lambda)^{(-j)}$ la curva estimada de la *muestra de entrenamiento* tal que

cumple la tarea de minimizar

$$\sum_{j \neq i} \{y_j - y(t_j)\}^2 + \lambda \int [D^2 y]^2.$$

Luego con la función $\hat{y}(t; \lambda)^{(-j)}$ se hace una estimación del valor y_j que no fue incluido en el modelo. El procedimiento anterior es repetido para cada observación y_j , $j = 1, \dots, n$, resultando así una suma de los errores al cuadrado sobre todos los valores del conjunto de datos, es decir, la suma de los errores al cuadrado de la *validación cruzada*. Dado que $c \leq \lambda \leq d$ escogemos el valor λ que minimice la suma de los errores al cuadrado de la *validación cruzada*.

La eficiencia del procedimiento con el parámetro de suavizamiento λ puede ser cuantificada por la función residual de validación cruzada

$$CV(\lambda) = n^{-1} \sum_{j=1}^n \{y_j - \hat{y}(t_j; \lambda)^{(-j)}\}^2$$

que felizmente es equivalente a

$$CV(\lambda) = n^{-1} \sum_{j=1}^n \left(\frac{y_j - \hat{y}(t_j)}{1 - S_{jj}} \right)^2$$

donde S_{jj} es el elemento j -ésimo de la diagonal de $\mathbf{S}_{\phi, \lambda} = \Phi(\Phi' \mathbf{W} \Phi + \lambda \mathbf{R})^{-1} \Phi' \mathbf{W}$, así la idea básica de la validación cruzada es escoger el valor λ que minimice $CV(\lambda)$.

Un supuesto que se toma en cuenta en la *validación cruzada*, es que las observaciones son relativamente independientes.

A los elementos S_{jj} se les suele dar el nombre de valores *leverage*; éstos determinan la cantidad en que los valores ajustados \hat{y}_j son influenciados por una observación particular y_j . Si algunos valores *leverage* son particularmente altos, entonces los valores ajusta-

dos deben ser tratados con mucho cuidado. Para poder saber si el ajuste es el apropiado, podemos examinar los residuales $y_j - \hat{y}(t_j)$ y los residuales suprimidos $(y_j - \hat{y}(t_j)) / (1 - S_{jj})$ esta última expresión da el residual entre y_j y el valor estimado $\hat{y}(t_j)$ del conjunto de datos en donde j es eliminado.

La técnica de *validación cruzada* tiene el inconveniente de inducir un bajo suavizamiento para los datos. Como consecuencia de lo anterior surge la *generalización de la validación cruzada* (*GCV*) que evita la necesidad de *re-suavizar* n veces; a diferencia de la *validación cruzada*. Su criterio es

$$GCV(\lambda) = n^{-1} \frac{\sum_{j=1}^n (y_j - \hat{y}(t_j))^2}{\{1 - n^{-1} \text{traza}(\mathbf{S}_{\phi, \lambda})\}}$$

dado que $df(\lambda) = \text{traza}(\mathbf{S}_{\phi, \lambda})$ simplificando la expresión anterior tenemos

$$\begin{aligned} GCV(\lambda) &= \frac{n^{-1} SSE}{[n^{-1} \text{traza}(\mathbf{I} - \mathbf{S}_{\phi, \lambda})]^2} \\ GCV(\lambda) &= \left(\frac{n}{n - df(\lambda)} \right) \left(\frac{SSE}{n - df(\lambda)} \right) \end{aligned}$$

nótese que $\left(\frac{SSE}{n - df(\lambda)} \right)$ es un estimador insesgado del σ^2 .

En el capítulo 5 de este trabajo se hace un análisis con los datos que corresponden al *IGAEM* desde el aspecto funcional. En la Figura 3.1 se presenta una estimación de una función subyacente a los datos del *IGAEM* ya mostrados.

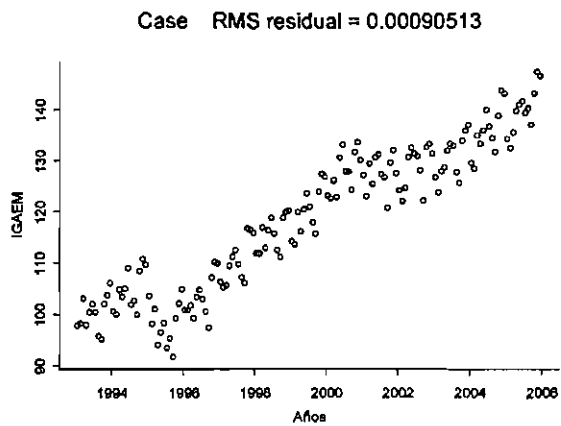


Figura 3.1: Ajuste funcional al conjunto de los datos del *IGAEM*.

Capítulo 4

Modelos lineales funcionales

Hemos explorado la estimación de una variable funcional; una vez obtenida tal función surgen preguntas tales como ¿qué tanto su forma puede ser asociada a otra variable? Esta asociación puede ser estudiada mediante los llamados modelos lineales funcionales.

El análisis de varianza, la regresión lineal simple o multivariada y los modelos lineales generalizados, sirven para este propósito. Ahora estas herramientas serán usadas en el contexto funcional.

Los modelos lineales funcionales se clasifican en tres categorías (Ramsay y Silverman, 2005):

- a) La variable respuesta es funcional y las covariables escalares.
- b) La variable respuesta es funcional y las covariables también son funciones.
- c) La variable respuesta es escalar y las variables independientes son funciones.

Para el caso con respuesta funcional y covariables escalares, describimos dos pro-

puestas: la de Ramsay y Silverman (2005) y la de Abramovich et al. (2004). Ambos casos proponen un modelo lineal funcional; la diferencia del modelo lineal funcional de Ramsay y el de Abramovich radica en la estructura del error y en la selección de bases. En esta sección analizaremos cada uno de los modelos y los contrastaremos haciendo un análisis estadístico con un mismo conjunto de datos, para apreciar de forma práctica las similitudes y diferencias que existen entre estos modelos.

4.1. Modelo lineal funcional para respuesta funcional y covariables escalares

4.1.1. Modelo de Ramsay y Silverman (2005)

En los modelos lineales funcionales con respuesta funcional y covariables escalares, se trata de predecir el comportamiento de una respuesta funcional $y_i(t)$, previamente estimada a partir del conjunto de datos y_{ij} , $i = 1, \dots, N$, $j = 1, \dots, n$, por medio de un conjunto de variables escalares predictoras o covariables escalares z_l donde $l = 1, \dots, q$.

Sin pérdida de generalidad para una variable funcional $y(t)$ (posiblemente ya estimada y suavizada; de ser muy ruidosas las observaciones y_j , $j = 1, \dots, n$, de acuerdo a los objetivos del estudio) el modelo general, que proponen Ramsay y Silverman (2005) para la respuesta funcional $y(t)$ y covariables escalares en $\mathbf{z} = (z_1, \dots, z_q)$

$$y(t) = \mathbf{z}'\boldsymbol{\beta}(t) + \epsilon(t)$$

en este caso $\boldsymbol{\beta}(t)$ es un vector funcional conteniendo q parámetros funcionales de la regresión

$$\beta = \begin{pmatrix} \beta_1(t) \\ \beta_2(t) \\ \cdot \\ \cdot \\ \cdot \\ \beta_q(t) \end{pmatrix}$$

y $\epsilon(t)$ es el error funcional del modelo suponiendo que para cada t $E[\epsilon(t)] = 0$ y $V[\epsilon(t)] = \sigma^2 I$. En este modelo se supone entonces que el valor esperado de $y(t)$ para cada t está dado por $\mathbf{z}'\beta(t)$.

Cualquier restricción lineal sobre los parámetros β es expresado por $L\beta = 0$ para una matriz apropiada L de q columnas.

Ya en aplicaciones a partir de un conjunto de N datos se define a

$$\mathbf{y}(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \\ \cdot \\ \cdot \\ \cdot \\ y_N(t) \end{pmatrix}$$

y a

$$\mathbf{Z} = \begin{pmatrix} \mathbf{z}'_1 \\ \mathbf{z}'_2 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{z}'_N \end{pmatrix}$$

es decir, $\mathbf{y}(t)$ contiene N funciones; \mathbf{Z} contiene N condiciones. Entonces para estimar β se propone minimizar el criterio de mínimos cuadrados

$$\begin{aligned} \min_{\beta} LMSSE(\beta) &= \int [\mathbf{y}(t) - \mathbf{Z}\beta(t)]' [\mathbf{y}(t) - \mathbf{Z}\beta(t)] dt & (4.1) \\ &= \|\mathbf{y}(t) - \mathbf{Z}\beta(t)\|^2 \end{aligned}$$

La estrategia computacional es minimizar sobre β , $\|\mathbf{y}(t) - \mathbf{Z}\beta(t)\|^2$ para cada t . Esto es, se puede calcular $\hat{\beta}(t)$ en una malla de valores de t usando el análisis de regresión simple y luego interpolar entre estos valores de $\hat{\beta}(t)$ obtenidos. Otra forma de estimar $\beta(t)$ es expresar primero a $\mathbf{y}(t)$ y a los parámetros funcionales $\beta_i(t) \in \beta(t)$ como expansiones de funciones base

$$\mathbf{y}(t) = \mathbf{C}\phi(t), \beta(t) = \mathbf{B}\theta(t)$$

donde ϕ es un vector de longitud $K_{\mathbf{y}}$ conteniendo funciones base linealmente independientes, \mathbf{C} es una matriz $N \times K_{\mathbf{y}}$ conteniendo los coeficientes de expansión funcional de las funciones $y_i(t)$, θ es un vector de longitud K_{β} conteniendo funciones base linealmente independientes

y \mathbf{B} es una matriz de $q \times K_\beta$ que contiene los coeficientes de la expansión funcional de las funciones $\beta_i(t)$.

Partiendo de los datos discretos y_{ij} , $i = 1, \dots, N$, $j = 1, \dots, n$, la estimación de β está dada por dos criterios:

Primer criterio: Suavizamiento de datos discretos y_{ij}

Si partimos de que el conjunto de variables respuesta en $\mathbf{y}(t)$ ya es el resultado de un suavizamiento previo de los datos discretos, es decir, si $\mathbf{y}(t) = \mathbf{C}\phi(t)$, entonces se define un operador diferencial \mathbf{L} para penalizar la rugosidad de $\beta(t)$

$$PEN_L(\beta) = \int [\mathbf{L}\beta(t)]' [\mathbf{L}\beta(t)] dt \quad (4.2)$$

ahora, el criterio de mínimos cuadrados para obtener una estimación de β está dado por

$$\min_{\beta} LMSSSE(\beta) = \|\mathbf{y}(t) - \mathbf{Z}\beta(t)\|^2 + PEN_L(\beta);$$

lo anterior es equivalente a tener

$$\min_{\mathbf{c}, \beta} PENSSE(\mathbf{y}/\beta) = \int (\mathbf{C}\phi - \mathbf{Z}\mathbf{B}\theta)' (\mathbf{C}\phi - \mathbf{Z}\mathbf{B}\theta) + \lambda \int \int (\mathbf{L}\mathbf{B}\theta)' (\mathbf{L}\mathbf{B}\theta),$$

como el operador de integración y el operador suma involucrados en este producto de matrices puede ser intercambiado y con la definición de las siguientes matrices

$$\mathbf{J}_{\phi\phi} = \int \phi\phi', \mathbf{J}_{\theta\theta} = \int \theta\theta', \mathbf{J}_{\phi\theta} = \int \phi\theta', \mathbf{R} = \int (\mathbf{L}\theta)(\mathbf{L}\theta)',$$

nótese que $\mathbf{J}_{\phi\phi}$ es una matriz constante simétrica de orden K_y de productos integrados pues ϕ es un vector columna de longitud K_y de funciones base, $\phi\phi'$ es entonces una matriz

cuadrada de orden K_y , conteniendo todos los posibles pares de estas funciones base. Entonces se tiene (Ramsay y Silverman, 2005)

$$\begin{aligned} \text{mín}_{\mathbf{c}, \beta} = PENSSE(y/\beta) = & \text{traza}(\mathbf{C}'\mathbf{C}\mathbf{J}_{\phi\phi}) + \text{traza}(\mathbf{Z}'\mathbf{Z}\mathbf{B}\mathbf{J}_{\theta\theta}\mathbf{B}') \\ & - 2\text{traza}(\mathbf{B}\mathbf{J}_{\theta\phi}\mathbf{C}'\mathbf{Z}) + \lambda\text{traza}(\mathbf{B}\mathbf{R}\mathbf{B}'). \end{aligned} \quad (4.3)$$

Calculando la derivada de la expresión (4.3) con respecto de la matriz \mathbf{B} e igualando a cero el resultado, se obtiene que \mathbf{B} satisface el sistema de ecuaciones

$$(\mathbf{Z}'\mathbf{Z}\mathbf{B}\mathbf{J}_{\theta\theta} + \lambda\mathbf{B}\mathbf{R}) = \mathbf{Z}'\mathbf{C}\mathbf{J}_{\phi\phi}$$

aplicando el producto Kronecker a esta última ecuación se tiene (Ramsay y Silverman, 2005)

$$\text{vec}(\hat{\mathbf{B}}) = [(\mathbf{J}_{\theta\theta} + \lambda\mathbf{R}) \otimes (\mathbf{Z}'\mathbf{Z})]^{-1} \text{vec}(\mathbf{Z}'\hat{\mathbf{C}}\mathbf{J}_{\phi\phi}), \quad (4.4)$$

como $\beta(t) = \mathbf{B}\theta(t)$, entonces

$$\beta(t) = \text{vec}(\hat{\mathbf{B}})\theta(t)$$

el parámetro de suavizamiento λ en la ecuación (4.4) impone el mismo nivel de suavizamiento para cada componente β_i del vector β .

Para este primer enfoque, entonces se toma la respuesta funcional $\mathbf{y}(t)$ suavizada y la estimación de los parámetros funcionales $\beta(t)$ son penalizados por un parámetro de suavizamiento λ .

Segundo criterio: Sin suavizamiento de datos discretos y_{ij}

Cuando la variable funcional $y_i(t)$ no es el resultado de un suavizamiento previo de los datos discretos y_{ij} , $i = 1, \dots, N$, $j = 1, \dots, n$, se usa la matriz \mathbf{Y} con los valores discretos para estimar los parámetros funcionales de la regresión.

Dado que $\beta(t) = \mathbf{B}\theta(t)$, se define Θ como la matriz de $n \times K_\beta$ que contiene los valores de las funciones base linealmente independientes contenidas en el vector θ de longitud K_β , entonces minimizando el criterio de mínimos cuadrados para la matriz \mathbf{Y} se tiene

$$\min_{\beta} = LMSS E(\beta) = \|\mathbf{Y} - \mathbf{ZB}\theta(t)\|^2 + \lambda \|\mathbf{L}\beta(t)\|^2$$

realizando un procedimiento análogo al primer caso, Ramsay y Silverman, (2005) calculan la expresión

$$\text{vec}(\hat{\mathbf{B}}) = [(\Theta'\Theta) \otimes (\mathbf{Z}'\mathbf{Z}) + \mathbf{R} \otimes \lambda\mathbf{I}]^{-1} (\Theta' \otimes \mathbf{Z}') \text{vec}(\mathbf{Y}),$$

de donde

$$\beta(t) = \text{vec}(\hat{\mathbf{B}})\theta(t).$$

4.1.2. Modelo de Abramovich et al. (2004)

Básicamente el modelo lineal funcional general que proponen, está dado por

$$dy(t) = \mathbf{z}'\beta(t) + \epsilon dW(t) \quad (4.5)$$

donde ϵ es una constante, y $W(t)$ representa una realización de un proceso de Wiener (un proceso de Wiener es un proceso estocástico de tiempo continuo, que se caracteriza por ser: un proceso de Markov que tiene incrementos independientes y las variaciones en el proceso producidas en un intervalo finito de tiempo se distribuyen normalmente). La versión discreta de este modelo es lo que se conoce como regresión no paramétrica para estimar $y(t)$ a partir

de datos observados y_j , $j = 1, \dots, n$. Brown y Low (1996) documentan la equivalencia entre (4.5) y su versión discreta

$$y_j = \mathbf{z}'\beta(t_j) + \frac{\sigma}{\sqrt{n}}(u_j - u_{j-1}), \quad j = 1, \dots, n \quad (4.6)$$

donde $u_j \sim \mathcal{N}(0, 1)$ independientes, donde σ^2 es la varianza de $\{y_j\}$. Nótese entonces que bajo este modelo discreto, la varianza del lado derecho de (4.6) debe ser multiplicada por $\sqrt{\frac{n}{2}}$ para mantener la igualdad en varianzas en (4.6).

4.1.3. Significancia estadística

Significancia en el modelo de Ramsay y Silverman (2005)

La significancia es corroborada usando intervalos de confianza para los parámetros de la regresión lineal funcional, que son los $\beta_l(t)$, $l = 1, \dots, q$; para calcular los intervalos de confianza puntuales de estos parámetros funcionales, es necesario hacer primero dos mapeos.

Dada la matriz de datos \mathbf{Y} de $N \times n$, se mapea la matriz \mathbf{Y} a una matriz de coeficientes \mathbf{C} de $N \times k_y$, quedando expresada la matriz \mathbf{C} como

$$\mathbf{C} = \mathbf{Y}\mathbf{S}_{\phi, \lambda_y}$$

expresando en producto kronecker esta ecuación, tenemos

$$\text{vec}(\mathbf{C}) = (\mathbf{S}_{\phi, \lambda_y} \otimes \mathbf{I}) \text{vec}(\mathbf{Y}),$$

entonces definimos el primer mapeo como

$$\text{Map1} = \mathbf{S}_{\phi, \lambda_y} \otimes \mathbf{I}$$

tal que la matriz $\mathbf{S}_{\phi, \lambda_y} \otimes \mathbf{I}$ mapea la matriz \mathbf{Y} a la matriz de coeficientes \mathbf{C} , a su vez, esta matriz \mathbf{C} , para las respuestas funcionales, necesita ser mapeada a una matriz \mathbf{B} de $q \times K_\beta$ para los parámetros funcionales del vector β .

Como la matriz de mapeo $\mathbf{S}_{\phi, \lambda_y}$, se define como

$$\mathbf{S}_{\phi, \lambda_y} = \Phi(\Phi'\Phi + \lambda_y \mathbf{R}_y)^{-1} \Phi'$$

entonces

$$\hat{\mathbf{C}} = \mathbf{Y} [\Phi(\Phi'\Phi + \lambda_y \mathbf{R}_y)^{-1} \Phi'] .$$

Dado que $\text{vec}(\mathbf{B}) = [\mathbf{J}_{\theta\theta} \otimes (\mathbf{Z}'\mathbf{Z}) + \mathbf{R} \otimes \lambda \mathbf{I}]^{-1} \text{vec}(\mathbf{Z}'\mathbf{C}\mathbf{J}_{\phi\theta})$, entonces ya tenemos a la matriz que mapea a la matriz \mathbf{C} a la matriz \mathbf{B} de parámetros funcionales de la regresión, esta es $[\mathbf{J}_{\theta\theta} \otimes (\mathbf{Z}'\mathbf{Z}) + \mathbf{R}_\beta \otimes \lambda_\beta \mathbf{I}]^{-1} [\mathbf{J}'_{\phi\theta} \otimes \mathbf{Z}']$.

Luego

$$\text{vec}(\hat{\mathbf{B}}) = [\mathbf{J}_{\theta\theta} \otimes (\mathbf{Z}'\mathbf{Z}) + \mathbf{R}_\beta \otimes \lambda_\beta \mathbf{I}]^{-1} [\mathbf{J}'_{\phi\theta} \otimes \mathbf{Z}'] \text{vec}(\hat{\mathbf{C}}'),$$

aquí λ_β es el parámetro de suavizamiento y \mathbf{R}_β es la matriz que penaliza la rugosidad de las funciones suavizadas β_t , por lo tanto la segunda matriz de mapeo es

$$\mathbf{S}_\beta = [\mathbf{J}_{\theta\theta} \otimes (\mathbf{Z}'\mathbf{Z}) + \mathbf{R}_\beta \otimes \lambda_\beta \mathbf{I}]^{-1} [\mathbf{J}'_{\phi\theta} \otimes \mathbf{Z}']$$

$\text{Map2} = \mathbf{S}_\beta$ por lo tanto

$$\begin{aligned} \text{vec}(\hat{\mathbf{B}}) &= \mathbf{S}_\beta \text{vec}(\hat{\mathbf{C}}') \\ &= \mathbf{S}_\beta (\mathbf{S}_{\phi, \lambda_y} \otimes \mathbf{I}) \text{vec}(\mathbf{Y}') \end{aligned}$$

definimos $Map3 = S_\beta (\mathbf{S}_{\phi, \lambda_y} \otimes \mathbf{I}) \text{vec}(\mathbf{Y}')$; cabe señalar que la varianza de los datos \mathbf{Y} está dado por $\text{Var}[\text{vec}(\mathbf{Y}')] = \sum_e \otimes \mathbf{I}$, donde \mathbf{I} es de orden N , \sum_e es la matriz de varianza y covarianza del vector de los residuales en el modelo $\mathbf{y}(t) = \mathbf{Z}\beta(t) + \epsilon(t)$, que no son otra cosa que los errores involucrados en el suavizamiento de los datos discretos de la variable de respuesta.

Finalmente tenemos

$$\begin{aligned} \text{Var}[\text{vec}(\hat{\mathbf{B}})] &= \text{Var}[\mathbf{S}_\beta (\mathbf{S}_{\phi, \lambda_y} \otimes \mathbf{I}) \text{vec}(\mathbf{Y})] \\ &= \mathbf{S}_\beta (\mathbf{S}_{\phi, \lambda_y} \otimes \mathbf{I}) \left[\sum_e \otimes \mathbf{I} \right] (\mathbf{S}_{\phi, \lambda_y} \otimes \mathbf{I}) \mathbf{S}'_\beta \end{aligned}$$

esto implica

$$\text{Var}[\text{vec}(\hat{\beta})] = (\Theta \times \mathbf{I}) \mathbf{S}_\beta (\mathbf{S}_{\phi, \lambda_y} \otimes \mathbf{I}) \left(\sum_e \otimes \mathbf{I} \right) (\mathbf{S}_{\phi, \lambda_y} \otimes \mathbf{I}) \mathbf{S}'_\beta (\Theta \otimes \mathbf{I})'$$

Si $\mathbf{J}_{\theta\theta}$ y $\mathbf{Z}'\mathbf{Z}$ son invertibles, entonces

$$\text{Var}[\text{vec}(\hat{\beta})] = \left[\mathbf{J}_{\theta\theta}^{-1} \mathbf{J}'_{\phi\theta} \mathbf{S}_{\phi, \lambda_y} \sum_e \mathbf{S}_{\phi, \lambda_y} \mathbf{J}_{\phi\theta} \mathbf{J}_{\theta\theta}^{-1} \right] \otimes (\mathbf{Z}'\mathbf{Z})^{-1}.$$

Cuando la variable funcional $y_i(t)$ no es el resultado de un suavizamiento previo de los datos discretos, entonces la expresión que resulta es

$$\text{Var}[\text{vec}(\hat{\beta})] = \left[\Theta (\Theta' \Theta)^{-1} \Theta' \sum_e \Theta (\Theta' \Theta)^{-1} \Theta' \right] \otimes (\mathbf{Z}'\mathbf{Z})^{-1}.$$

Con la varianza del estimador de β se pueden construir intervalos de confianza para cada uno de los $\hat{\beta}_l$ y poder evaluar si son o no significativos los $\beta_l(t)$, $l = 1, \dots, q$ en el

modelo de regresión lineal que se tiene.

En este tipo de análisis el poder averiguar si los efectos funcionales son distintos de cero e investigar en qué tiempo t pasa esto es muy informativo. La diferencia entre un análisis de varianza ordinario y un análisis de varianza funcional (*FANOVA*) es que en este último tenemos a $\beta(t)$ funcional que depende de un argumento t , la F en este contexto, también es funcional dependiendo de un argumento t , definiendo el coeficiente F funcionalmente como (Ramsay y Silverman, 2005)

$$F = \frac{df(error) [SSY(t) - SSE(t)]}{df(regresión)SSE(t)}$$

Para esta expresión df denota los grados de libertad y SS denota la suma de los cuadrados respectivos, pero funcionales.

Los intervalos de confianza puntuales son calculados al agregar y sustraer un múltiplo del error estándar funcional, es decir la raíz cuadrada de la varianza muestral para el ajuste actual. Un intervalo puntual con el 95 % de confianza para una función $\beta_l(t)$, está dado por

$$\left(\hat{\beta}_l(t) - 2 \times \sqrt{Var(\hat{\beta}_l(t))}, \hat{\beta}_l(t) + 2 \times \sqrt{Var(\hat{\beta}_l(t))} \right).$$

donde $Var(\hat{\beta}_l(t)) = \theta_l(t)'Var(\hat{\mathbf{B}})\theta_l(t)$. Estos intervalos son puntuales porque reflejan la región de confianza para valores individuales de t , y no para toda la curva $\beta_l(t)$.

Significancia en el modelo lineal funcional de Abramovich et al. (2004)

En este modelo, la significancia es corroborada por medio de contraste de hipótesis.

Como primer paso importante el valor de σ debe ser estimado; esto se hace a partir

de los coeficientes *wavelets* que se encuentran en el nivel más fino de la descomposición *wavelet* $\{y_j\}$ (Donoho y Johnstone, 1995): si $\hat{\sigma} = \text{median}(|d_{J-1, \kappa}| : 0 \leq \kappa < 2^{J-1})/0,6745$, y J está dado por $n = 2^J$ que es el número de observaciones a considerar. Recordemos que si $y(t)$ representa a $\{y_j\}$ funcionalmente entonces en la lógica *wavelet*,

$$y(t) = \sum_{\kappa} c_{J-1, \kappa} \phi_{J-1, \kappa}(t_j) + \sum_{\kappa} d_{J-1, \kappa} \psi_{J-1, \kappa}(t_j) + \sum_{\kappa} d_{J-2, \kappa} \psi_{J-2, \kappa}(t_j) + \dots + \sum_{\kappa} d_{1, \kappa} \psi_{1, \kappa}(t_j).$$

Dada la equivalencia entre (4.5) y (4.6) la estimación de β , en el caso en que \mathbf{Z} sea de rango completo, está dado por $\hat{\beta} = (\mathbf{Z}'\mathbf{Z})^{-1} \mathbf{Z}'\mathbf{Y}$; la representación de cada β_l es, de acuerdo a la lógica *wavelet*,

$$\beta_l(t) = \sum_{\kappa} c_{J-1, \kappa} \phi_{J-1, \kappa}(t) + \sum_{\kappa} d_{J-1, \kappa} \psi_{J-1, \kappa}(t) + \sum_{\kappa} d_{J-2, \kappa} \psi_{J-2, \kappa}(t) + \dots + \sum_{\kappa} d_{1, \kappa} \psi_{1, \kappa}(t). \quad (4.7)$$

Debemos probar

$$H_0 : \beta_l(t) = 0 \text{ vs } H_1 : \beta_l(t) \neq 0$$

que es equivalente a probar

$$H_0 : c_{J-1, \kappa} = 0 \text{ y } d_{j, \kappa} = 0, \forall j \geq 0; \kappa = 0, \dots, 2^j - 1 \quad \text{vs} \quad (4.8)$$

$H_1 : c_{J-1, \kappa} \neq 0$ o existe al menos un $j \geq 0, \kappa = 0, \dots, 2^j - 1$ tal que $H_1 : d_{j, \kappa} \neq 0$.

Como las bases

$$\{\psi_{j,\kappa}(t)\}, \quad j \text{ fijo}, \kappa \in \mathbb{Z}, \kappa = 0, \dots, 2^j - 1$$

$$\{\phi_{j,\kappa}(t)\}, \quad j \text{ fijo}, \kappa \in \mathbb{Z}, \kappa = 0, \dots, 2^j - 1$$

son periódicas y ortonormales, entonces los $\beta_l(t)$ pertenecen a espacios de Besov $\mathbb{B}_{p,q}^s(C)$ (Vidakovic, 1995), donde el parámetro p , es visto como un grado de no homogeneidad de las funciones $\hat{\beta}_l(t)$, s indica el número de derivadas que tienen estas funciones, además $s > 0$, $C > 0$ y $1 \leq p, q \leq \infty$. Dados estos parámetros de las bolas de Besov entonces para contrastar las hipótesis (4.8), Abramovich et al. (2004) proponen hacer uso de un procedimiento de contraste de hipótesis funcionales *asintóticamente minimax* en dos modalidades, *no-adaptativa y adaptativa*.

La prueba *asintótica minimax no-adaptativa*, consiste en suponer primero que conocemos todos los parámetros de la bola de Besov, se define a \mathcal{J} como el conjunto de todos los niveles de resolución *wavelet*, luego se impone la partición $\mathcal{J} = \mathcal{J}_- \cup \mathcal{J}_+$, donde $\mathcal{J}_- = \{0, \dots, j(s) - 1\}$ denota los niveles más gruesos y $\mathcal{J}_+ = \{j(s), \dots, j_\eta - 1\}$ denota los niveles más finos, se define $j(s) = \frac{2}{4s+1} \log_2(C\eta^{-2})$, donde $s' = \min\left\{s, s - \frac{1}{2p} + \frac{1}{4}\right\}$, que delimita los niveles de resolución *wavelet* a ser considerados en el procedimiento de contraste de hipótesis y el error estándar se denota con η ; así para cada $j \in \mathcal{J}_-$, se define S_j como

$$S_j = \sum_{\kappa=0}^{2^j-1} (d_{j,\kappa}^2 - \eta^2) \quad (4.9)$$

y para $j \in \mathcal{J}_+$ y $\lambda > 0$, se define $S_j(\lambda)$ como

$$S_j(\lambda) = \sum_{\kappa=0}^{2^j-1} [(d_{j,\kappa}^2 \mathbf{1}(|d_{j,\kappa}^2| > \eta\lambda) - \eta^2 b(\lambda))] \quad (4.10)$$

donde $b(\lambda) = \mathbb{E} [\xi^2 \mathbf{1}(|\xi| > \lambda)]$, donde ξ es una variable aleatoria Gaussiana y $\mathbf{1}(A)$, es la función indicadora en A . Dada la partición de \mathcal{J} se definen los parámetros $T(j(s))$ y $Q(j(s))$ como

$$T(j(s)) = \sum_{j=0}^{j(s)-1} S_j \text{ y } Q(j(s)) = \sum_{j=j(s)}^{j_\eta-1} S_j(\lambda_j). \quad (4.11)$$

Finalmente se definen $v_0^2(j(s))$ y $w_0^2(j(s))$ como

$$v_0^2(j(s)) = 2\eta^4 2^{j(s)}$$

$$w_0^2(j(s)) = \eta^4 \sum_{j=j(s)}^{j_\eta-1} 2^j d(\lambda_j)$$

donde $\lambda_j = \sqrt{(j - j(s) + 8) \ln 2}$ (umbral universal, Johnstone y Donoho; 1994; Fan, 1994),

es utilizado para descartar coeficientes no importantes.

Entonces para un nivel de significancia $\alpha \in (0, 1)$ se define la estadística de prueba

$$\phi^* = \left\{ \begin{array}{l} \mathbf{1} \{T(j(s)) > v_0(j(s))z_{1-\alpha}\} \text{ si } p \geq 2 \\ \mathbf{1} \{T(j(s)) + Q(j(s)) > \sqrt{v_0^2(j(s)) + w_0^2(j(s))}z_{1-\alpha}\} \text{ si } 1 \leq p < 2 \end{array} \right\} \quad (4.12)$$

donde $z_{1-\alpha}$ es el percentil $(1-\alpha)100\%$ de la distribución Gaussiana. Si $\phi^* = 1$ se rechaza H_0 .

Los parámetros $v_0^2(j(s))$ y $w_0^2(j(s))$ son las varianzas de $T(j(s))$ y $Q(j(s))$ respectivamente.

Para la prueba *asintótica minimax adaptativa*, se supone que los parámetros de la bola de Besov son desconocidos, pero que $0 < s \leq s_{\text{máx}}$, $1 \leq p, q \leq \infty$, $s > \frac{1}{p}$ y $0 < C \leq C_{\text{máx}}$. La idea de la prueba *adaptativa* es considerar todo el rango de $\rho(s) = \rho_{\text{mín}}, \dots, \rho_\eta - 1$ y rechazar H_0 si es rechazada en al menos un nivel de $j(s)$, dando por resultado

$$\phi_{\eta}^* = \left\{ \begin{array}{l} 1 \left[\max_{j: \min \leq |j(s)| \leq j\eta-1} \left\{ \frac{T(j(s))}{\sqrt{v_0^2(j(s))}} \right\} > \sqrt{2 \ln \ln(\eta^{-2})} \right] \text{ si } p \geq 2 \\ 1 \left[\max_{j: \min \leq |j(s)| \leq j\eta-1} \left\{ \frac{T(j(s))+Q(j(s))}{\sqrt{v_0^2(j(s))+w_0^2(j(s))}} \right\} > \sqrt{2 \ln \ln(\eta^{-2})} \right] \text{ si } 1 \leq p < \infty \end{array} \right\}. \quad (4.13)$$

4.1.4. Aplicación

Descripción del experimento

El ejemplo trata de la elaboración de quesos, y queremos saber qué tratamientos son significativos en la calidad del queso. Los datos se ordenan de la siguiente forma, la matriz Y de 173×16 contiene 173 observaciones y 16 réplicas, matriz que define la variable respuesta. Se sabe que el número de observaciones se determina al multiplicar los niveles de cada factor y el total de réplicas, pero en el ejemplo sólo se consideran 173 observaciones.

La matriz de diseño es de 16×11 en ésta se incluyen las covariables escalares: pH con 5 niveles $(-1,27, -1,11, -0,94, 0,010, 1,13)$, $\%prot$ con 3 niveles $(-1,14, -0,07, 1,01)$, $\%hum$ con 3 niveles $(-1,28, -0,93, 1,17)$, $tempext$ con 3 niveles $(-0,89, 0,21, 1,30)$, y las interacciones dobles de estas covariables $pH * \%prot$, $pH * \%hum$, $pH * tempext$,

$\%prot * \%hum$, $\%prot * tempext$ y $\%hum * tempext$, donde $\%prot$, es la proteína, $\%hum$, es la humedad y $tempext$, es la temperatura.

El modelo que se tiene es

$$\begin{aligned} y(t) = & \mu(t) + \beta_{pH}(t)pH + \beta_{\%prot}(t)\%prot + \beta_{\%hum}(t)\%hum + \\ & \beta_{tempext}(t)tempext + \beta_{pH \times \%prot}(t)pH \times \%prot + \beta_{pH \times \%hum}(t)pH \times \%hum + \\ & \beta_{pH \times tempext}(t)pH \times tempext + \beta_{\%prot \times \%hum}(t)\%prot \times \%hum + \\ & \beta_{\%prot \times tempext}(t)\%prot \times tempext + \beta_{\%hum \times tempext}(t)\%hum \times tempext, \end{aligned}$$

y las hipótesis a probar son las siguientes:

$$\begin{aligned}
 H_0 : \beta_{pH}(t) = 0 & \quad vs \quad H_1 : \beta_{pH}(t) \neq 0 \\
 H_0 : \beta_{\%prot}(t) = 0 & \quad vs \quad H_1 : \beta_{\%prot}(t) \neq 0 \\
 H_0 : \beta_{\%hum}(t) = 0 & \quad vs \quad H_1 : \beta_{\%hum}(t) \neq 0 \\
 H_0 : \beta_{tempezt}(t) = 0 & \quad vs \quad H_1 : \beta_{tempezt}(t) \neq 0 \\
 H_0 : \beta_{pH \times \%prot}(t) = 0 & \quad vs \quad H_1 : \beta_{pH \times \%prot}(t) \neq 0 \\
 H_0 : \beta_{pH \times \%hum}(t) = 0 & \quad vs \quad H_1 : \beta_{pH \times \%hum}(t) \neq 0 \\
 H_0 : \beta_{pH \times tempezt}(t) = 0 & \quad vs \quad H_1 : \beta_{pH \times tempezt}(t) \neq 0 \\
 H_0 : \beta_{\%prot \times \%hum}(t) = 0 & \quad vs \quad H_1 : \beta_{\%prot \times \%hum}(t) \neq 0 \\
 H_0 : \beta_{\%prot \times tempezt}(t) = 0 & \quad vs \quad H_1 : \beta_{\%prot \times tempezt}(t) \neq 0 \\
 H_0 : \beta_{\%hum \times tempezt}(t) = 0 & \quad vs \quad H_1 : \beta_{\%hum \times tempezt}(t) \neq 0
 \end{aligned}$$

Efectuando un análisis estadístico funcional, los resultados son los siguientes:

Análisis con el modelo de Ramsay y Silverman

En las Figuras 4.1 y 4.2 se muestran las 16 curvas de los valores discretos de las respuestas, de donde se puede apreciar el mismo patrón de dispersión para cada una de ellas.

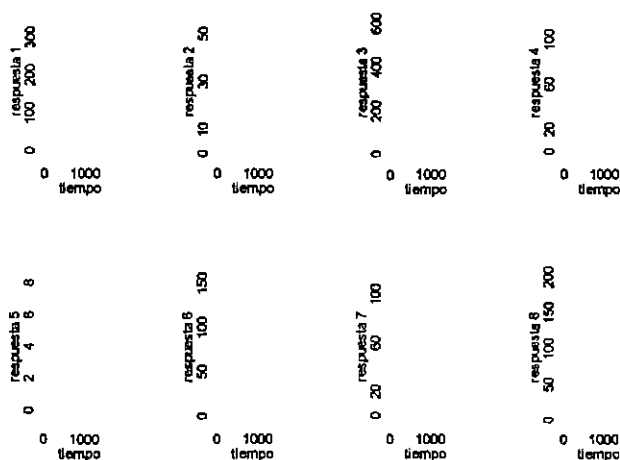


Figura 4.1: Datos escalares de las respuestas

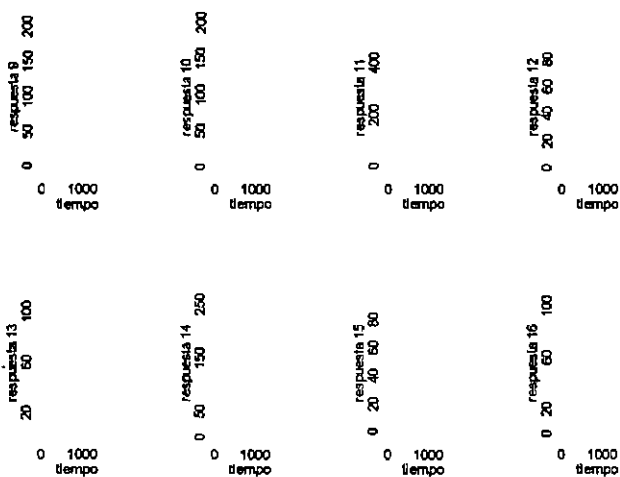


Figura 4.2: Datos escalares de las respuestas.

En la Figura 4.3 están las 16 respuestas funcionales $y_i(t)$ previamente suavizadas y estimadas de los valores de las respuestas discretas dadas en la matriz \mathbf{Y} de 173×16 .

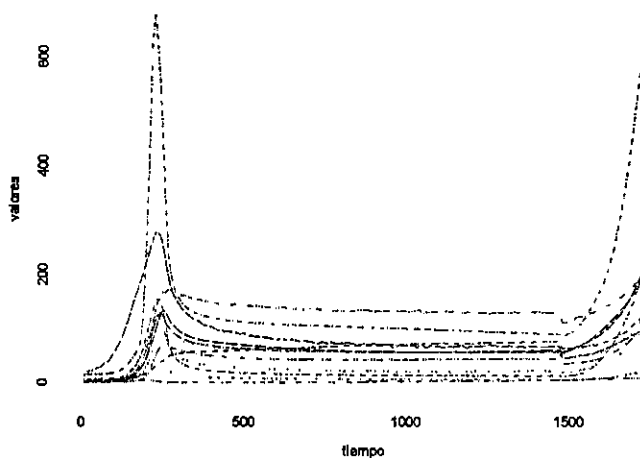


Figura 4.3: 16 respuestas funcionales suavizadas.

En las Figuras 4.4 y 4.5 se muestran los 11 parámetros funcionales $\hat{\beta}_1(t)$. Algunas de ellas son funciones negativas y otras positivas esto quiere decir que tienen o bien un impacto positivo o negativo en la respuesta funcional. En cambio el coeficiente funcional $pH*tempext$ parece no influir de manera importante en las respuestas funcionales.

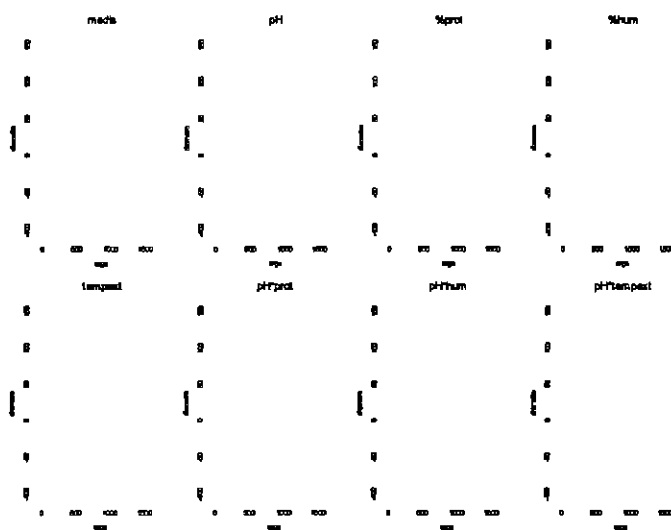


Figura 4.4: Coeficientes funcionales estimados.

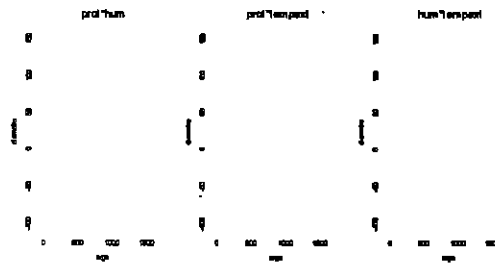


Figura 4.5: Coeficientes funcionales del modelo.

Las trayectorias de las respuestas funcionales $\hat{y}_i(t)$ que se muestran en la Figura 4.6 son muy semejantes a las mostradas en la Figura 4.3, las funciones $\hat{y}_i(t)$ son generadas dadas las estimaciones de parámetros funcionales $\beta_i(t)$.

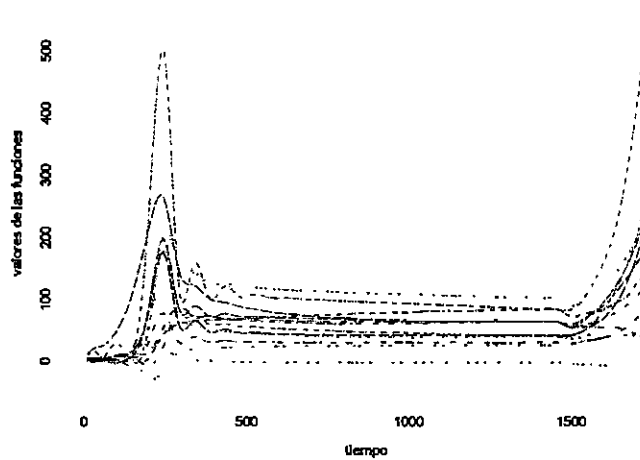


Figura 4.6: Respuestas funcionales estimadas.

La Figura 4.7 muestra los valores de desviación estándar de los residuales, donde los residuales están dados por $\epsilon = y_i - \hat{y}_i$ la trayectoria de esta curva muestra finalmente que los residuales no son del todo independientes ni normales.

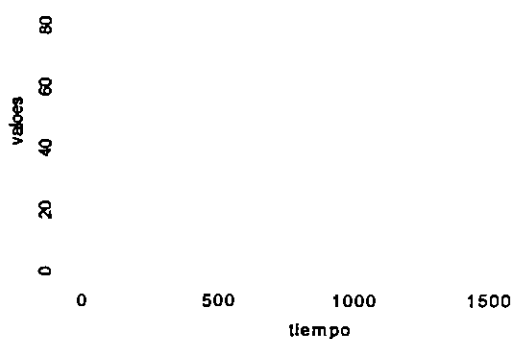


Figura 4.7: Desviación estándar de los residuales.

Dado que los intervalos de confianza puntuales para cada uno de los $\beta_i(t)$ con el 95 % de confianza está dado por

$$\left(\hat{\beta}_i(t) - 2 \times \sqrt{\text{Var}(\hat{\beta}_i(t))}, \hat{\beta}_i(t) + 2 \times \sqrt{\text{Var}(\hat{\beta}_i(t))} \right)$$

entonces observando las Figuras 4.8 y 4.9 concluimos finalmente que los efectos de los tratamientos pH , y $\%hum*tempert$ son importantes en la variabilidad de la respuesta, mientras que el resto de los tratamientos y las interacciones de estos no tienen impacto en la variable de respuesta $y_i(t)$ pues el intervalo para éstos tratamientos contiene al cero.

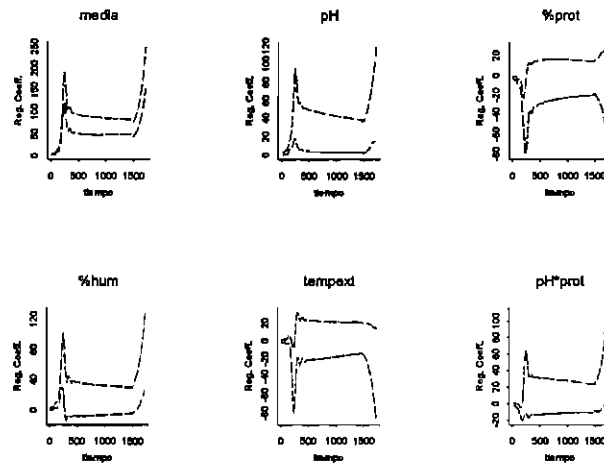


Figura 4.8: Intervalos de confianza para los coeficientes funcionales.

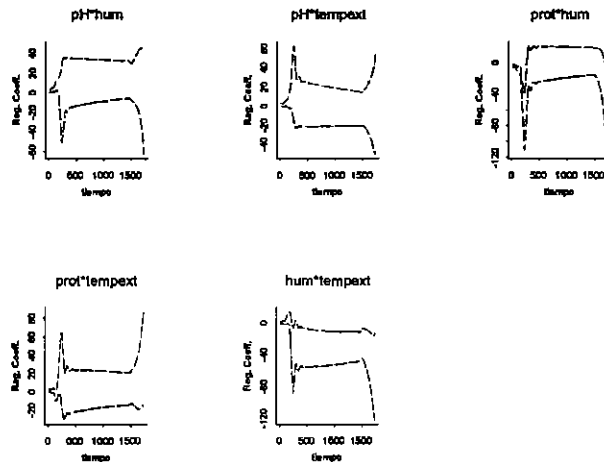


Figura 4.9: Coeficientes funcionales con sus intervalos de confianza.

Análisis con el modelo de Abramovich et al. (2004)

Para el análisis de este mismo conjunto con bases *wavelet* se presenta la siguiente consideración: en la estimación de σ tuvimos problemas; usando la descomposición *wavelet* de los datos funcionales observados, se partió del número total de observaciones, en este

caso cada dato funcional contaba con 173 observaciones. En el programa MATLAB pudimos obtener la descomposición de cada dato funcional en seis niveles de resolución, éstos disponibles por contar con tal número de observaciones en cada dato funcional observado. Dadas las características de estos datos funcionales, contar con 173 datos produce un sobremuestro, y con ello el último nivel de resolución utilizado para estimar σ , implicaba que se obtuviera prácticamente $\hat{\sigma} \approx 0$. Así en este procedimiento el sobremuestreo causaba que todos los efectos se declararan significativos. Por lo anterior, decidimos tomar una fracción del conjunto de datos ($16 = 2^4$) para realizar la descomposición *wavelet* y de ahí proceder al contraste de hipótesis. Estos 16 datos son suficientes para describir el comportamiento de los datos funcionales observados como en el experimento (Figura 4.10). El problema de sobremuestreo no es comentado mucho en la literatura, sino más bien a partir del concepto de tasa de muestreo de Nyquist se enfatiza los problemas del submuestreo (ver Vidakovich, 1999, el concepto de “aliasing”). Sin embargo en el contexto estadístico del procedimiento de contraste propuesto por Abramovich et al, deben considerarse los problemas que el sobremuestro puede inducir en la estimación de σ , ésta siendo vital en el procedimiento.

Las 16 observaciones son equiespaciadas del conjunto de datos dado, esto es así porque el uso de la transformación discreta de *wavelet* requiere un diseño equiespaciado. De entrada nosotros esperaríamos que las conclusiones a las que llegemos sean las mismas a las que llegamos con el modelo de Ramsay y Silverman (2005).

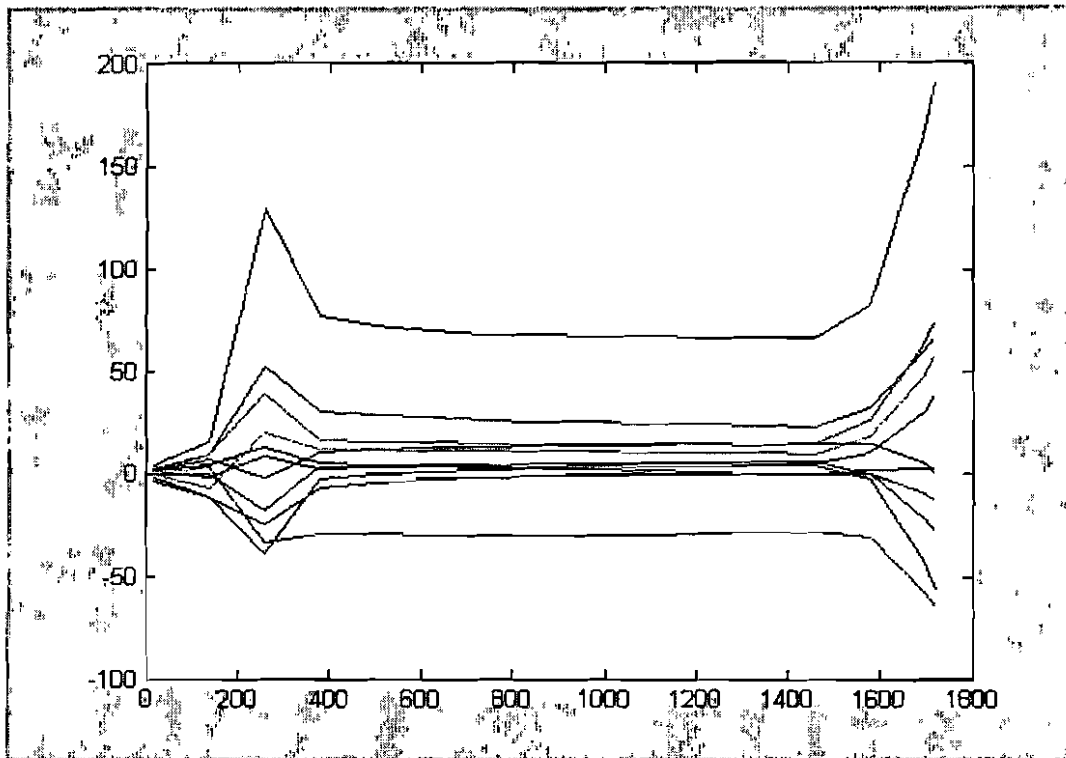


Figura 4.10: Usando sólo 16 observaciones de cada respuesta y_i .

Dadas las nuevas dimensiones de y y Z entonces B cuenta con los 11 tratamientos y cada uno de estos con 16 observaciones.

El objetivo es tomar cada uno de los B_l , $l = 1, \dots, q$ y aplicarles una transformación discreta *wavelet* y estimar $\beta_l(t)$, una vez hecho esto, lo que nos interesa es contrastar:

$$H_0 : \beta_l = 0 \text{ vs } H_1 : \beta_l \neq 0.$$

El error estándar de $\hat{\beta}_l$ está dado por

$$\eta = \sqrt{\frac{n}{2}} \hat{\sigma} \sqrt{m_{ii}},$$

donde m_{ii} es el i -ésimo elemento de la diagonal de la matriz $(\mathbf{X}\mathbf{X})^{-1}$ y $\hat{\sigma}$ es la desviación estándar del error, el cual podemos calcular con la mediana de los valores absolutos de los

valores que se encuentran en el último nivel de la descomposición *wavelet* de las $y_i(t)$, como ya se mencionó.

Por ejemplo, la Figura 4.11 muestra los coeficientes *wavelets* que corresponden a la descomposición *wavelet* de $y_1(t)$.

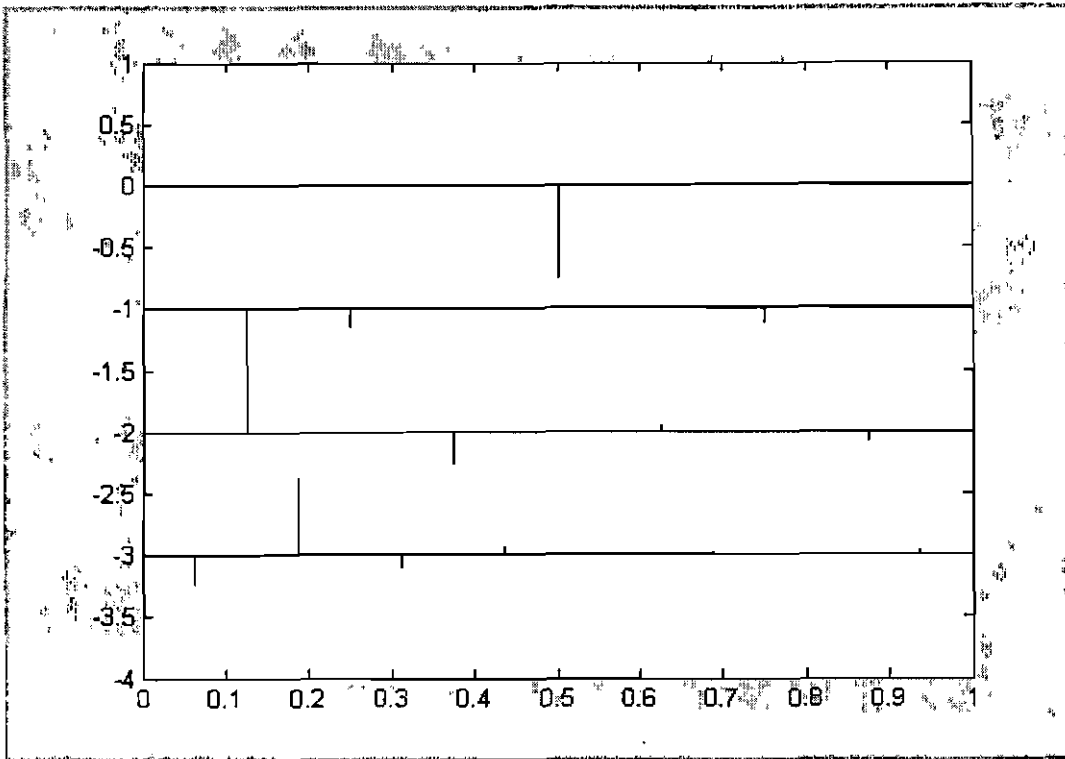


Figura 4.11: Coeficientes de la transformación discreta *wavelet* de $y_1(t)$.

El número de coeficientes *wavelets*, está determinado por

$$n = \frac{n}{2} + \frac{n}{4} + \dots + \frac{n}{2^{J-1}} + \frac{n}{2^J} + \frac{n}{2^J}$$

dado que $J = 4$, $n = 2^J = 16$ y

$$c_{J-1,\kappa} \approx \int \phi_{J-1,\kappa} y(t) dt$$

$$d_{j,\kappa} \approx \int \psi_{j,\kappa} y(t) dt \quad j = 0, 2, \dots, J-1$$

donde

$$c_{J-1} = \left(c_{J-1,1}, c_{J-1,2}, \dots, c_{J-1, \frac{n}{2^{J-1}}} \right)$$

$$d_{J-1} = \left(d_{J-1,1}, d_{J-1,2}, \dots, d_{J-1, \frac{n}{2^{J-1}}} \right)$$

$$d_{J-2} = \left(d_{J-2,1}, d_{J-2,2}, \dots, d_{J-2, \frac{n}{2^{J-2}}} \right)$$

$$\cdot \quad \quad \quad \cdot$$

$$\cdot \quad \quad \quad \cdot$$

$$\cdot \quad \quad \quad \cdot$$

$$d_1 = \left(d_{1,1}, d_{1,2}, \dots, d_{1, \frac{n}{2}} \right)$$

entonces para $y_1(t)$ en total se tiene 16 coeficientes wavelets. En la Figura 4.1.4 el último nivel de la descomposición está indicado con el -3 y cuenta con 8 coeficientes que corresponden al nivel más fino de la descomposición *wavelet* (los cuales serán considerados para la estimación de σ).

Los resultados para las pruebas

$$H_0 : \beta_l(t) = 0 \text{ vs } H_1 : \beta_l(t) \neq 0 \text{ para } l = 1, \dots, 11.$$

son los siguientes:

| | | |
|--|------------------------------------|--|
| $\beta_0 := \text{media}$ | $p - \text{value} = 0$ | $\phi^* = 1$ se rechaza $H_0 : \beta_0 = 0$ |
| $\beta_1 := \text{ph}$ | $p - \text{value} = 0,00055667$ | $\phi^* = 1$ se rechaza $H_0 : \beta_1 = 0$ |
| $\beta_2 := \%prot$ | $p - \text{value} = 0,76867$ | $\phi^* = 0$ no se rechaza $H_0 : \beta_2 = 0$ |
| $\beta_3 := \%hum$ | $p - \text{value} = 4,6981e - 010$ | $\phi^* = 1$ se rechaza $H_0 : \beta_3 = 0$ |
| $\beta_4 := \text{tempext}$ | $p - \text{value} = 0,22925$ | $\phi^* = 0$ no se rechaza $H_0 : \beta_4 = 0$ |
| $\beta_5 := \text{ph} \times \text{prot}$ | $p - \text{value} = 0,0012624$ | $\phi^* = 1$ se rechaza $H_0 : \beta_5 = 0$ |
| $\beta_6 := \text{ph} \times \text{hum}$ | $p - \text{value} = 0,8643$ | $\phi^* = 0$ no se rechaza $H_0 : \beta_6 = 0$ |
| $\beta_7 := \text{ph} \times \text{tempext}$ | $p - \text{value} = 0,93771$ | $\phi^* = 0$ no se rechaza $H_0 : \beta_7 = 0$ |
| $\beta_8 := \text{prot} \times \text{hum}$ | $p - \text{value} = 7,3578e - 010$ | $\phi^* = 1$ se rechaza $H_0 : \beta_8 = 0$ |
| $\beta_9 := \text{prot} \times \text{tempext}$ | $p - \text{value} = 0,34591$ | $\phi^* = 0$ no se rechaza $H_0 : \beta_9 = 0$ |
| $\beta_{10} := \text{hum} \times \text{tempext}$ | $p - \text{value} = 0,011404$ | $\phi^* = 1$ se rechaza $H_0 : \beta_{10} = 0$ |

Como podemos ver los tratamientos ph y $\%hum * \text{tempext}$ son importantes en la variabilidad de la respuesta, además $\%hum$, $\text{ph} * \text{prot}$ y $\text{prot} * \text{hum}$ mientras que el resto de los tratamientos y las interacciones de estos no tienen impacto en la variable de respuesta. A diferencia del análisis con el modelo de Ramsay y Silverman (2005), en éste se tienen 5 tratamientos significativos en la variabilidad de la respuesta. Podríamos pensar que esto se debe a dos razones: la primera es que un análisis estadístico funcional con bases *wavelet*, es más sensible para detectar efectos significativos en análisis estadístico, la segunda, que el número de observaciones tomado para este análisis, no sea el más representativo del conjunto de los datos dados en un principio.

4.2. Modelo lineal funcional para respuesta funcional y covariables funcionales

4.2.1. Modelo general

En este modelo la respuesta $y_i, i = 1, \dots, N$ y cada covariable z_i son funciones en el mismo argumento t . Una covariable escalar puede ser vista como una covariable funcional si es expandida en términos de una base constante. Cuando z_i solamente influye a $y_i(t)$ a través de su valor $z_i(t)$ en el tiempo t se dice que se tiene un *modelo concurrente*. Cuando se tiene q covariables funcionales z_{il} , cada una multiplicada por su coeficiente de regresión funcional β_l entonces el modelo de regresión múltiple funcional concurrente esta dado por

$$y_i(t) = \sum_{l=1}^q z_{il}(t)\beta_l(t) + \epsilon_i(t)$$

El modelo lineal funcional concurrente, en notación matricial es

$$\mathbf{y}(t) = \mathbf{Z}(t)\boldsymbol{\beta}(t) + \boldsymbol{\epsilon}(t) \quad (4.14)$$

donde \mathbf{Z} es una matriz funcional de $N \times q$ conteniendo los z_{il} y $\boldsymbol{\beta}$ es un vector de parámetros funcionales de longitud q conteniendo todas las funciones $\beta_l(t)$, y es un vector funcional de longitud N conteniendo las respuestas funcionales.

Al expandir cada una de las funciones β_l donde $l = 1, \dots, q$, en funciones base, nosotros deberíamos permitir que la base en que serán expandidos cada uno de los β_l y el coeficiente de suavizamiento varíe de un β_l a otro, pues habrá ocasiones en que algunos β_l 's necesiten más funciones base por su alta frecuencia de variabilidad, por estas razones definimos la rugosidad penalizada para cada β_l como

$$PEN_l(\beta_l) = \lambda_l \int [L_l \beta_l(t)]^2 dt. \quad (4.15)$$

Cada penalización es definida a través de un operador diferencial L_l , que es el apropiado para cada uno de los parámetros funcionales.

Cuando tenemos el modelo

$$\mathbf{y}(t) = \mathbf{Z}(t)\boldsymbol{\beta}(t) + \boldsymbol{\epsilon}(t)$$

y queremos estimar a $\boldsymbol{\beta}$ podemos usar el criterio de mínimos cuadrados

$$LMSS(\boldsymbol{\beta}) = \int [\mathbf{y}(t) - \mathbf{Z}(t)\boldsymbol{\beta}(t)]' [\mathbf{y}(t) - \mathbf{Z}(t)\boldsymbol{\beta}(t)] dt + \sum_l^q \lambda_l \int [L_l \beta_l(t)]^2 dt \quad (4.16)$$

al derivar con respecto a $\boldsymbol{\beta}$ e igualar a cero la ecuación (4.16) podemos encontrar un estimador de este.

Dada la ecuación (4.15) se reescribe el modelo $\mathbf{y}(t) = \mathbf{Z}(t)\boldsymbol{\beta}(t) + \boldsymbol{\epsilon}(t)$ para hacer algunos cambios. Como ya se dijo cada función de regresión β_l tiene una expansión de la forma

$$\beta_l(t) = \sum_k^{K_l} b_{k_l} \theta_{k_l}(t) = \boldsymbol{\theta}_l(t)' \mathbf{b}_l(t)$$

cada β_l tiene k_l funciones base θ_{k_l} , además sea

$$\mathbf{K}_\beta = \sum_l^q K_l$$

primero construimos el vector \mathbf{b} de longitud K_β y apilamos los vectores verticalmente

$$\mathbf{b} = (b'_1, b'_2, b'_3, \dots, b'_q).$$

Luego definimos una matriz Θ de $q \times K_\beta$, tal que

$$\Theta = \begin{bmatrix} \theta'_1 & 0 & \dots & 0 \\ 0 & \theta'_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \theta'_q \end{bmatrix}$$

entonces podemos expresar el modelo (4.14) como

$$\mathbf{y}(t) = \mathbf{Z}(t)\Theta(t)\mathbf{b} + \epsilon(t)$$

es decir, separamos las funciones base de la expansión en una matriz $\Theta(t)$ y a los coeficientes de esta expansión en un vector \mathbf{b} .

De hecho si $\mathbf{Z}^*(t) = \mathbf{Z}(t)\Theta(t)$ donde $\mathbf{Z}^*(t)$ es de $N \times K_\beta$ entonces se tiene

$$\mathbf{y}(t) = \mathbf{Z}^*(t)\mathbf{b} + \epsilon(t).$$

Si una de las respuestas funcionales y_i es expandida en términos de K_y funciones base, entonces el número total de grados de libertad para el error df_e en el modelo es $df_e = NK_y - K_\beta$, teniendo esto en cuenta nos permite evitar un sobreajuste de los datos.

Para penalizar la rugosidad de cada $\beta_i(t)$ definimos una matriz \mathbf{R}_i de orden K_i multiplicada por su correspondiente parámetro de suavizamiento

$$\mathbf{R}_l = \lambda_l \int \boldsymbol{\theta}_l(t) \boldsymbol{\theta}'_l(t) dt$$

se coloca cada una de estas matrices en la diagonal de una nueva matriz \mathbf{R} de orden K_β para escribir la ecuación (4.16) como

$$\begin{aligned} LMSS(\boldsymbol{\beta}) &= \int \mathbf{y}'(t) \mathbf{y}(t) - \int \mathbf{b}' \boldsymbol{\Theta}'(t) \mathbf{Z}'(t) \mathbf{y}(t) - \int \mathbf{y}(t) \mathbf{b}' \boldsymbol{\Theta}'(t) \mathbf{Z}'(t) \\ &\quad + \int \mathbf{b}' \boldsymbol{\Theta}'(t) \mathbf{Z}'(t) \mathbf{Z}(t) \boldsymbol{\Theta}(t) \mathbf{b} + \mathbf{b} \mathbf{b}' \mathbf{R} \end{aligned}$$

igualando a cero y derivando esta última expresión con respecto a \mathbf{b} se tiene

$$\left[\int \boldsymbol{\Theta}'(t) \mathbf{Z}'(t) \mathbf{Z}(t) \boldsymbol{\Theta}(t) dt + \mathbf{R} \right] \mathbf{b} = \left[\int \boldsymbol{\Theta}'(t) \mathbf{Z}'(t) \mathbf{y}(t) dt \right]. \quad (4.17)$$

4.2.2. Intervalos de confianza para los parámetros funcionales de la regresión

Se hace una expansión de la respuesta funcional $\mathbf{y}(t)$ en funciones base se tiene $\mathbf{y} = \mathbf{C}\boldsymbol{\phi}$ donde $\boldsymbol{\phi}$ el vector de funciones base de longitud K_y y \mathbf{C} es la matriz de coeficientes de la expansión de las funciones base de las respuestas funcionales

De la ecuación (4.16) se obtiene un estimador para \mathbf{b}

$$\hat{\mathbf{b}} = \left[\int \boldsymbol{\Theta}' \mathbf{Z}' \mathbf{Z} \boldsymbol{\Theta} + \mathbf{R} \right]^{-1} \left[\int \boldsymbol{\Theta}' \mathbf{Z} \mathbf{C} \boldsymbol{\phi} \right]$$

al aplicar el producto Kronecker a $\left[\int \boldsymbol{\Theta}' \mathbf{Z} \mathbf{C} \boldsymbol{\phi} \right]$ tenemos

$$\hat{\mathbf{b}} = \left[\int \Theta' \mathbf{Z}' \mathbf{Z} \Theta + \mathbf{R} \right]^{-1} \left[\int \phi' \otimes \mathbf{C} \Theta' \mathbf{Z} \right] \text{vec}(\mathbf{C}) \quad (4.18)$$

Finalmente de la ecuación (4.18) podemos obtener la segunda matriz de mapeo

$$\text{Map2} = \left[\int \Theta' \mathbf{Z}' \mathbf{Z} \Theta + \mathbf{R} \right]^{-1} \left[\int \phi' \otimes (\Theta' \mathbf{Z}') \right]$$

esta matriz de mapeo convierte el vector de coeficientes \mathbf{C} a un vector \mathbf{b} de longitud K_β .

Sabemos que *Map2* es necesaria para obtener una tercera matriz *Map3* la cual sirve para construir los intervalos de confianza y que resulta de la composición de *Map1* y *Map2* como $\text{Map1} = \phi^{-1}$ entonces

$$\text{Map3} = \text{Map2} \circ \text{Map1}$$

$$\mathbf{b} = \left[\int \Theta' \mathbf{Z}' \mathbf{Z} \Theta + \mathbf{R} \right]^{-1} \left[\int \phi' \otimes (\Theta' \mathbf{Z}') \right] \phi^{-1} \mathbf{y}$$

con \mathbf{b} podemos calcular los intervalos de confianza para $\beta_i(t)$, tal y como se hizo con el modelo lineal funcional anterior.

4.2.3. Modelo funcional lineal completo

Un modelo funcional completo es aquél en donde la respuesta y las covariables z_i son funcionales. Un modelo funcional completo tiene la siguiente forma

$$y_i(t) = \alpha(t) + \int_0^n z_i(s) \beta(s, t) ds + \varepsilon_i(t).$$

En este modelo β está en función de dos parámetros de tiempo s y t , cabe mencionar que podemos hacer $s = t$.

$z_i(s)$ está definida en el intervalo $[0, n]$ y β se expresa funcionalmente con K_1 funciones base $\{\eta_k\}$ y K_2 funciones base $\{\theta_l\}$, es decir

$$\beta(s, t) = \sum_{k=1}^{K_1} \sum_{l=1}^{K_2} b_{kl} \eta_k(s) \theta_l(t) = \eta'(s) \mathbf{B} \theta(t),$$

donde \mathbf{B} es una matriz de coeficientes b_{kl} de $K_1 \times K_2$, o más simple

$$\beta = \eta' \mathbf{B} \theta$$

y $\alpha(t) = \sum_{l=1}^{K_2} a_l \theta_l(t) = \theta'(t) \mathbf{a}$ con θ un vector de funciones base para expandir la función α .

En el modelo funcional lineal

$$\mathbf{y}(t) = \int \mathbf{z}(s) \beta(s, t) ds + \varepsilon(t)$$

y es un vector de longitud N que contiene las funciones $y_1(t), y_2(t), \dots, y_N(t)$ y $\mathbf{z}(s)$ es un vector de longitud N que contiene a las funciones $z_1(s), z_2(s), \dots, z_N(s)$, el parámetro $\beta(s, t)$ depende de dos argumentos s y t , y por consiguiente podemos penalizar a β con respecto a cada uno de ellos para obtener un mejor modelo. Además la función covariable \mathbf{z}_i también pueda estar en función del argumento t de forma que se tenga $z(s, t)$. De hecho, el modelo funcional completo es:

$$\begin{aligned} y_i(t) &= \int_{\Omega_t} z_i(s, t) \beta(s, t) ds + \varepsilon_i(t) \\ &= \int_{\Omega_t} z_i(s, t) \theta'(s, t) \mathbf{b} ds + \varepsilon_i(t). \end{aligned} \quad (4.19)$$

donde al intervalo de integración para el argumento s se le permitirá variar sobre t y se usa la notación Ω_t para indicar que el intervalo está asociado a t . Entonces la función coeficiente está dado ahora por

$$\beta(s, t) = \sum_k^{K_\beta} b_k \theta_k(s, t) = \theta'(s, t) \mathbf{b}.$$

Debemos tener presente que para poder obtener un estimador de la función coeficiente $\beta(s, t)$ es necesario aplicar el criterio de mínimos cuadrados a la ecuación (4.19); si se quiere penalizar la rugosidad en el criterio de los mínimos cuadrados para el caso del modelo funcional completo hace falta definir dos matrices que penalizan la rugosidad de β , una con respecto a s y la otra con respecto a t .

4.2.4. Aplicación al conjunto de datos de la temperatura global de la tierra

Para esta aplicación se usa el modelo

$$\mathbf{y}(t) = \mathbf{Z}(t)\beta(t) + \epsilon(t)$$

el objetivo es predecir el comportamiento de la temperatura global anual por medio de la cantidad de dióxido de carbono global anual emitido desde 1988 al 2003.

Para este ejemplo se tiene una respuesta funcional y una covariable funcional. La respuesta funcional es un vector \mathbf{y} que contiene 16 valores que representan las temperaturas globales anuales durante los años 1988 al 2003, y la covariable funcional es representada por un vector \mathbf{z} el cual contiene 16 los valores escalares que representan las emisiones de

dióxido de carbono global anual durante los años 1998 al 2003.

El análisis estadístico funcional es el siguiente:

La Figura 4.12 muestra los 16 valores de la temperatura global anual a través del tiempo dado.

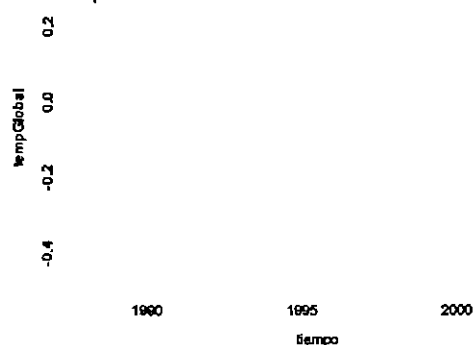


Figura 4.12: Valores de la temperatura global anual de 1998 al 2003.

La Figura 4.13 muestra los 16 reportes discretos del CO_2 durante el tiempo dado.

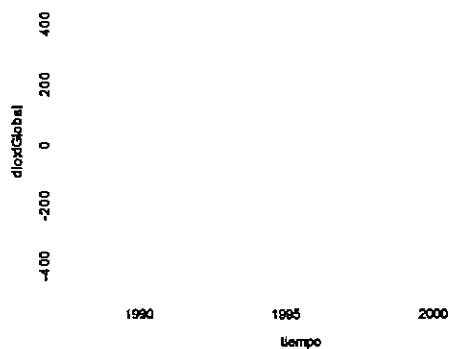


Figura 4.13: Dióxido de carbono global anual del 1998 al 2003.

La Figura 4.14 muestra la función $\hat{\beta}(t)$ en la cual se puede apreciar que el CO_2 tiene un efecto positivo sobre la temperatura en la mayoría del tiempo y sólo en tres pequeños intervalos de tiempo el CO_2 tiene un efecto negativo en la temperatura.

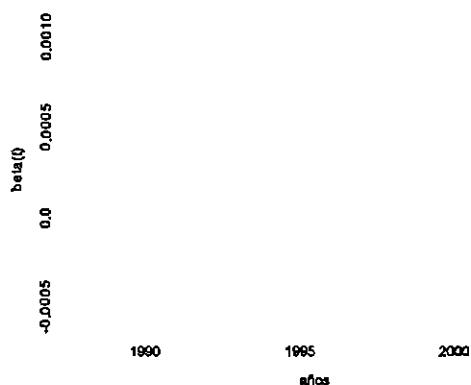


Figura 4.14: Coeficiente funcional de la regresión.

La Figura 4.15 muestra la función $\hat{y}(t)$, que es la respuesta funcional estimada en la cual se aprecia que se ajusta bien a los 16 datos discretos de la temperatura global anual.

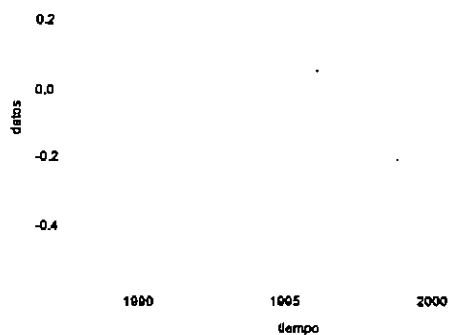


Figura 4.15: Gráfica de $\hat{y}(t)$ estimada de los 16 observaciones.

En la Figura 4.16 se aprecia claramente que los residuales no son independientes, lo cual implica tener cuidado en la interpretación del análisis funcional.

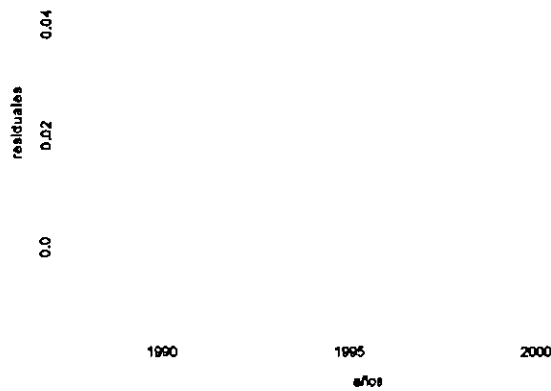


Figura 4.16: Gráfica de los residuales.

La gráfica de la Figura 4.17 muestra el error estándar de la función $\hat{\beta}(t)$, los valores que toma esta función son muy pequeños.

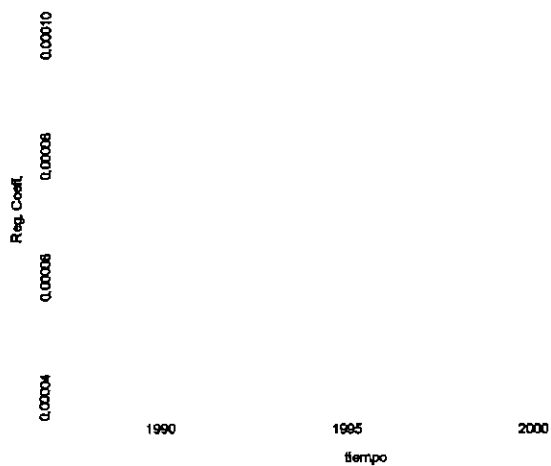


Figura 4.17: Error estándar del parámetro funcional $\hat{\beta}(t)$.

Por último generamos un intervalo puntual para la función $\hat{\beta}(t)$ con un 95 % de confianza. Observando la Figura 4.18 concluimos que el efecto del CO_2 en la temperatura es positiva, es decir al aumentar la concentración de CO_2 en la atmósfera de la tierra aumentará la temperatura; esta conclusión se deriva del hecho de que el intervalo de confianza puntual no contiene el cero.

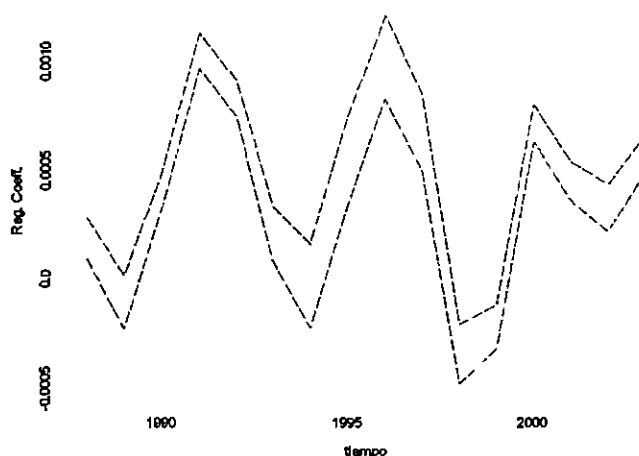


Figura 4.18: Intervalo del parámetro funcional $\hat{\beta}(t)$.

El análisis anterior corresponde al modelo donde β depende de un sólo parámetro t , analicemos ahora un conjunto de datos donde β depende dos veces del mismo parámetro t , usando así el modelo lineal funcional completo

$$y_i(t) = \alpha(t) + \int_0^n z_i(s)\beta(s, t)ds + \varepsilon_i(t).$$

El primero conjunto de datos corresponde a una matriz de 12×16 que contiene las temperaturas globales mensuales de la tierra y océano, durante los años 1988 al 2003. El segundo conjunto de datos es la matriz de 12×16 que contiene el reporte mensual de

CO_2 durante 16 años. Como ya comentamos las respuestas y_i serán funciones, al igual que las covariables z_i .

La Figura 4.19 muestra 16 funciones estimadas $z_i(t)$ y suavizadas.

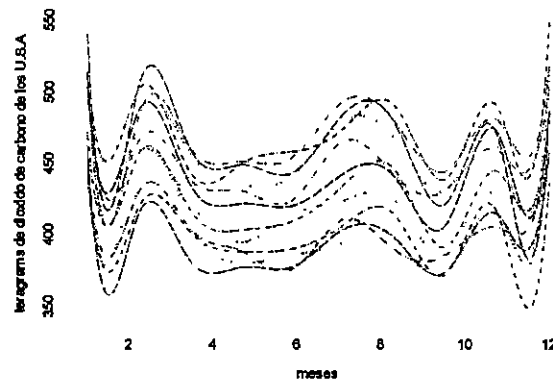


Figura 4.19: Funciones estimadas del CO_2 .

Los valores de la temperatura global mensual de la tierra y el océano correspondientes a cada año se muestran en la Figura 4.20. Estos son los valores que deseamos estimar dado los valores del CO_2 .

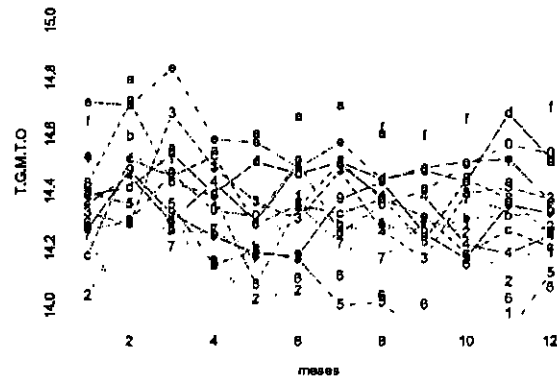


Figura 4.20: Datos escalares de las temperaturas globales mensuales de la tierra y el océano.

En la Figura 4.21 se muestran las 16 funciones $\hat{y}_i(t)$ de los datos de la temperatura. Se puede apreciar de estas gráficas que la temperatura aumenta a principio de año y a finales del mismo. En general la temperatura se eleva en la época de invierno.

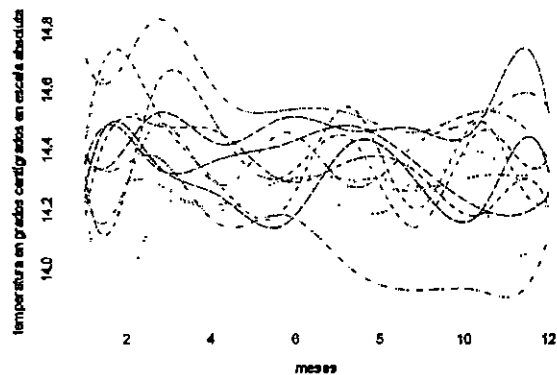


Figura 4.21: 16 funciones $\hat{y}_i(t)$ de las temperaturas globales mensuales de la tierra y el

océano.

Dado que el modelo en uso es

$$y_i(t) = \alpha(t) + \int_0^n z_i(s)\beta(s,t)ds + \varepsilon_i(t);$$

entonces la gráfica de la función $\alpha(t)$, se muestra en la Figura 4.22.

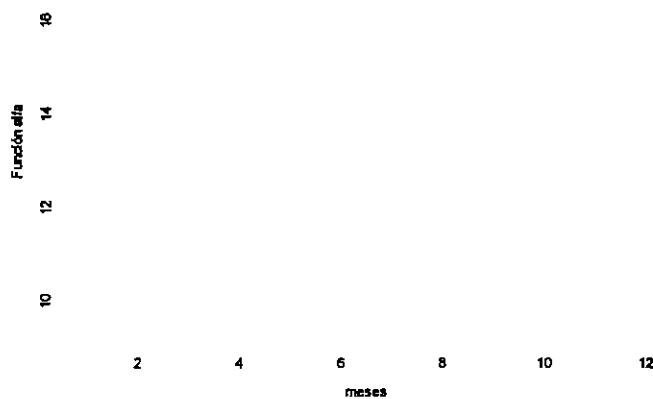


Figura 4.22: Término funcional independiente del modelo de regresión.

En la Figura 4.23 se muestra la gráfica tridimensional de la función $\beta(s, t)$. Nuevamente esta función nos muestra que el CO_2 tiene mayor impacto en la temperatura a finales y principios de cada año.

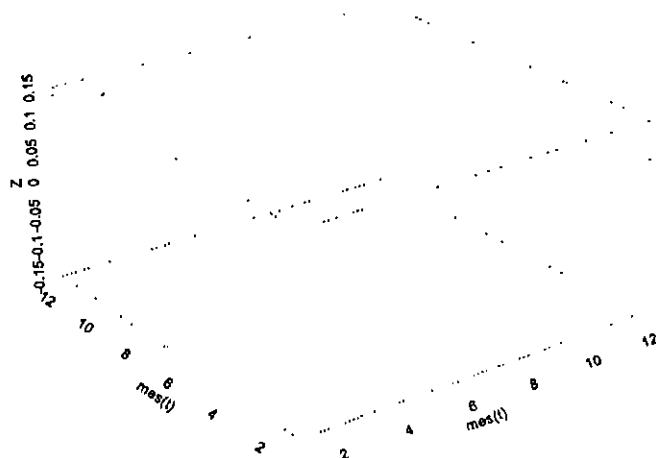


Figura 4.23: Función bivariada de la regresión.

Al comparar gráficamente las 16 funciones $\hat{y}_i(t)$ de la Figura 4.24 con las 16 funciones $y_i(t)$ suavizadas que se muestran en la Figura 4.20, se observa que existen diferencias entre ellas.

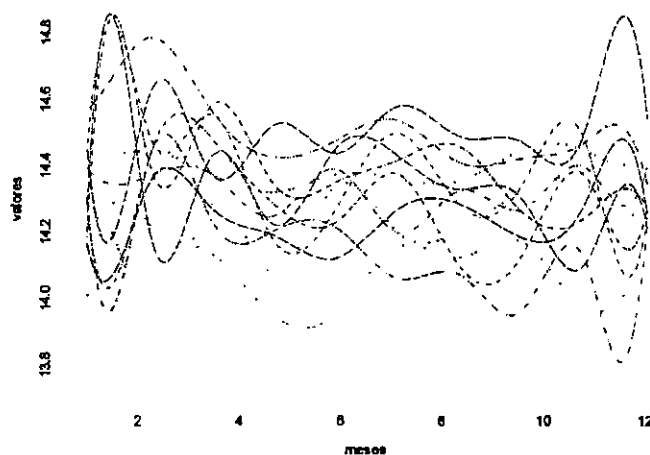


Figura 4.24: Temperatura globales mensuales funcionales estimadas.

Comparando los coeficientes de determinación mostrados en las Figuras 4.25 y 4.26

para cada una de las funciones estimadas, podemos concluir, salvo para los años 1990, 1996, 1997 y 1998, que la cantidad de CO_2 en el medio ambiente tiene impacto en la temperatura del planeta.

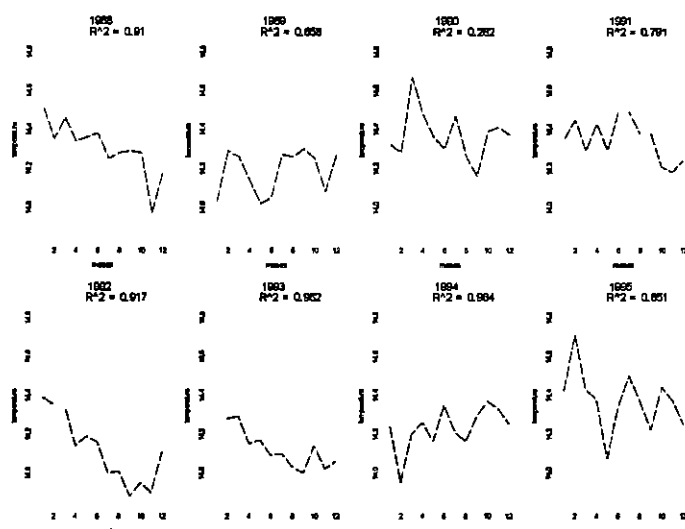


Figura 4.25: Funciones \hat{y}_i vs sus promedios funcionales y el valor del R^2 .

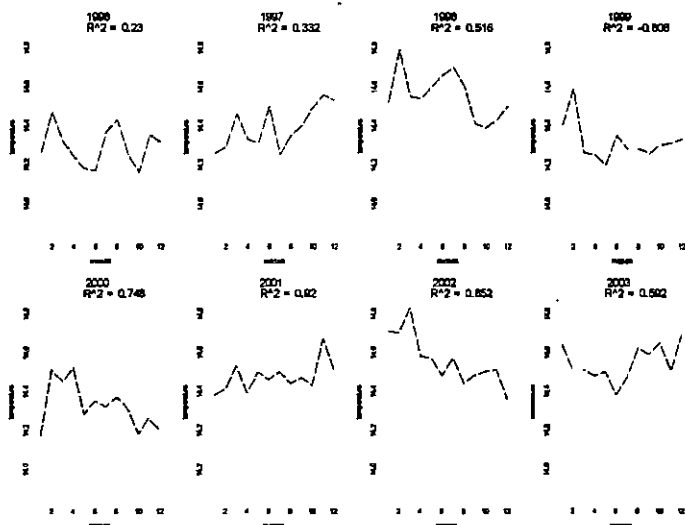


Figura 4.26: Funciones \hat{y}_i vs sus promedios funcionales y el valor del R^2 .

4.3. Modelo lineal funcional para respuesta escalar y covariables funcionales

4.3.1. Modelo general

En este capítulo consideraremos los modelos lineales funcionales con respuesta escalar, definidos para conjuntos de datos, en donde la variable de respuesta es un valor escalar y las covariables predictoras serán representadas a través de funciones. También los coeficientes de este modelo, serán funciones.

En el caso del modelo funcional con respuesta escalar, se tiene por estimar $\beta(t)$ y los escalares \hat{y}_j dados los y_j , $j = 1, \dots, n$ y la matriz \mathbf{Z} de $n \times p$. El vector \mathbf{y} está representado por:

$$\mathbf{y} = \alpha + \int_0^n \mathbf{Z}(t)\beta(t)dt + \epsilon$$

donde $\alpha = \beta_0$, cuando $l = 1$, α será representada por medio una función constante.

Los elementos del vector β , pueden ser expandidos funcionalmente en términos de un conjunto de funciones base del tipo *fourier*, *spline* o cualquier otra base. Definimos el estimador de y_j como:

$$\hat{y}_j = \int_0^n \mathbf{Z}(t)\beta(t)dt.$$

La expansión funcional de los elementos del vector β esta dado en términos del conjunto de funciones base $\theta_k(t)$. Definimos el vector de funciones base *spline* θ de longitud k_β tal que β sea

$$\beta(t) = \sum_k^{K_\beta} b_k \theta_k(t) = \theta' \mathbf{b}$$

escogemos a K_β lo suficientemente grande, de tal forma que no se pierda información significativa del conjunto de datos a estimar, definimos las covariables funcionales $z_i(t)$ como

$$z_i(t) = \sum_k^{k_z} c_{ik} \psi_k(t) \text{ o } \mathbf{Z}(t) = \mathbf{C}\psi(t)$$

donde ψ es un vector de longitud k_z que contiene las funciones base *spline* y \mathbf{C} es la matriz de coeficientes de $n \times k_z$.

4.3.2. Intervalos de confianza para los parámetros funcionales de la regresión

En este capítulo no es necesario suavizar la variable de la respuesta, por lo tanto podemos olvidarnos del primer mapeo *Map1*. Se define

$$\zeta = (\alpha, b_1, \dots, b_K)' \text{ y } \mathbf{J}_{\psi\theta} = \int \psi(t)\theta'(t)dt,$$

y

$$\mathbf{Z} = [1 \ \mathbf{C}\mathbf{J}_{\psi\theta}]$$

de $N \times (k_z + 1)$, así

$$\hat{\mathbf{y}} = \mathbf{Z}\hat{\zeta},$$

sea

$$\mathbf{R} = \int [D^2\phi(t)] [D^2\phi'(t)] dt$$

luego agregando columnas y renglones de longitud $k_x + 1$ de ceros a la matriz \mathbf{R} se obtiene \mathbf{R}_0 . Así Ramsay y Silverman (2005) definen la matriz correspondiente a *Map2* que puede ser simplemente tomada de la expresión $(\mathbf{Z}'\mathbf{Z} + \lambda\mathbf{R}_0)\hat{\zeta} = \mathbf{Z}'\mathbf{y}$, de modo que

$$Map2 = (\mathbf{Z}'\mathbf{Z} + \lambda\mathbf{R}_0)^{-1}\mathbf{Z}'.$$

La matriz de varianza y covarianza \sum_e calculada de los residuales es ahora un estimador escalar σ_e^2 del cuadrado del promedio de los residuales y la varianza de $\hat{\zeta}$ esta dada por

$$Var[\hat{\zeta}] = \sigma_e^2(\mathbf{Z}'\mathbf{Z} + \lambda\mathbf{R}_0)^{-1}\mathbf{Z}'\mathbf{Z}(\mathbf{Z}'\mathbf{Z} + \lambda\mathbf{R}_0)^{-1}.$$

Con todas estas herramientas ya podemos calcular los intervalos de confianza para los parámetros funcionales del modelo $\mathbf{y} = \alpha + \int_0^n \mathbf{Z}(t)\beta(t)dt + \epsilon$.

4.3.3. Aplicación al conjunto de datos de la temperatura global de la tierra

En la época actual la preocupación por el calentamiento global de la tierra es una constante en los temas ambientales. Como todos sabemos existen varios contaminantes que provocan el aumento de la temperatura, pero los más destacados son los gases de efecto invernadero, tales como el dióxido de carbono (CO_2), el metano (CH_4), óxido nitroso (N_2O), los hidrofluorocarbonos (*HFCS*), los perfluorocarbonos (*PFCS*) y el hexafluoruro de azufre (SF_6). Cada uno de ellos contribuye en mayor o menor medida en el calentamiento de la tierra, pero el contaminante con un 60 % de impacto en el calentamiento de la tierra

es el CO_2 .

Con la finalidad de corroborar esta afirmación obtenida de la literatura, se tomaron dos conjuntos de datos, el primer conjunto corresponde a la variable de respuesta, el cual es un vector de longitud 16, éste contiene los promedios de las temperaturas globales anuales del año 1988 al 2003. El segundo conjunto de datos está formado por una matriz de 12×16 , la cual contiene el reporte mensual de CO_2 para cada uno de los tiempos marcados.

El análisis estadístico funcional es el siguiente:

La Figura 4.27 muestra los promedios de las temperaturas globales durante los años 1988 al 2003, cada uno de estos promedios está dado en grados centígrados, y como ponemos apreciar la temperatura en los últimos años ha ido en aumento.

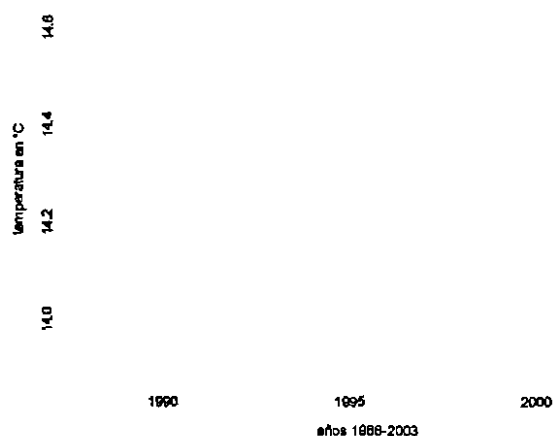


Figura 4.27: Promedios de las temperaturas globales anuales.

En la Figura 4.28 se muestran las 16 curvas de CO_2 para cada uno de los años. En general estas curvas tienen un comportamiento bastante parecido entre ellas.

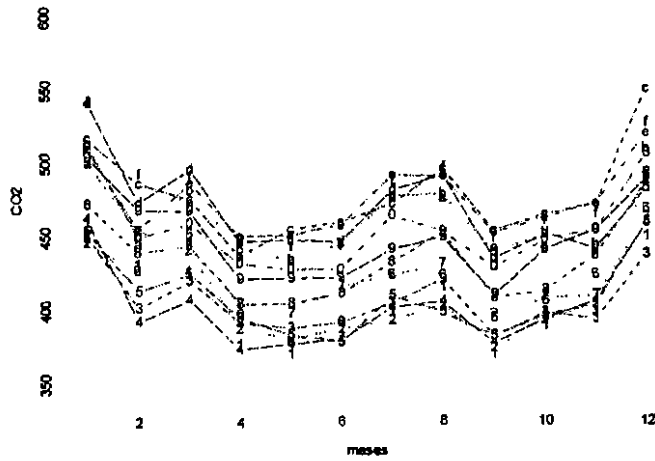


Figura 4.28: Dióxido de carbono emitido durante los años 1988 – 2003.

Lo que sigue es representar a la matriz \mathbf{Z} en funciones base *spline*. El orden de las funciones base será 3 y el número de funciones base se elige de acuerdo al comportamiento de los datos. Juzgando por las gráficas de la Figura 4.29 el ajuste funcional es bueno para los datos del CO_2 .

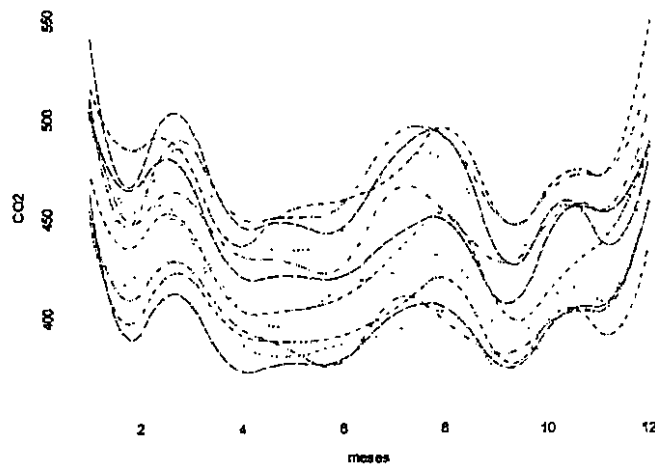


Figura 4.29: Representación funcional del CO_2 .

La Figura 4.30 muestra que el parámetro funcional $\hat{\beta}(t)$ es positivo en el intervalo de enero a mayo mientras que de mayo a julio la función tiene una caída bastante fuerte siendo negativa y en el resto del año empieza nuevamente a tener una pendiente positiva. Con esto estaríamos diciendo que el CO_2 aumenta en invierno y disminuye en primavera-verano.

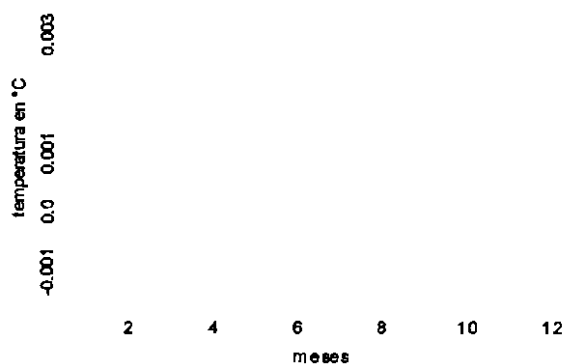


Figura 4.30: Coeficiente funcional estimado $\beta(t)$.

En la Figura 4.31 se observa que la recta de regresión estimada, genera residuales pequeños para la mayoría de los datos escalares y_j .

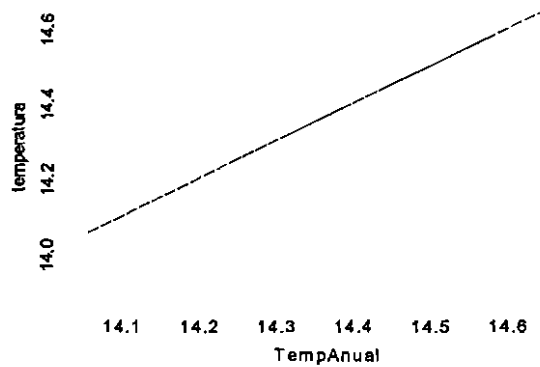


Figura 4.31: Recta de regresión ajustada a los datos y_j .

Para validar el modelo estimado, calculamos coeficiente de determinación que está dado por:

$$R = +\sqrt{\frac{1 - SSE}{SSy}}$$

donde $SSE = \sum_j^n \varepsilon_j^2(t) = \sum_j^n \{\hat{y}_j(t) - y_j(t)\}^2$ y $SSy = \sum_j^n y_j^2 - n\bar{y}^2(t) = \sum_j^n \{y_j(t) - \bar{y}(t)\}^2$ tal que SSE mide la variación residual que queda sin explicar por el modelo ajustado y SSy es la suma total corregida de los cuadrados de y_j . El coeficiente de determinación es de 0,80387.

Finalmente calculamos un intervalo de confianza para el parámetro funcional $\hat{\beta}(t)$, que se muestra en la Figura 4.32. De este análisis funcional se concluye que la temperatura del planeta aumenta al principio y al final del año y el aumento de la temperatura del planeta depende efectivamente en gran medida del CO_2 .

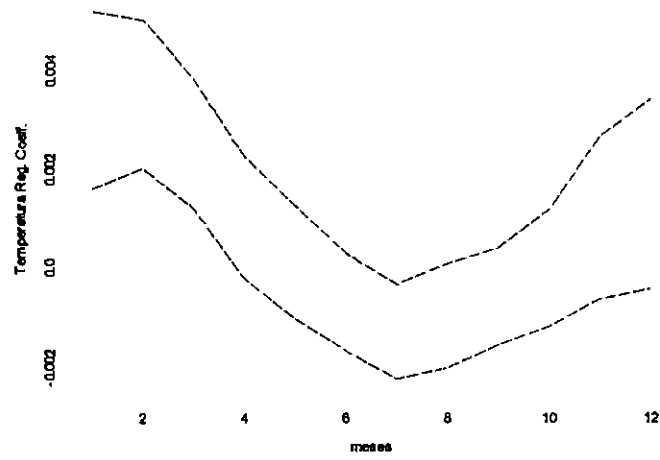


Figura 4.32: Intervalo puntual del parámetro funcional $\beta(t)$.

Capítulo 5

Ecuaciones diferenciales ordinarias en el análisis de datos funcionales

El uso de las ecuaciones diferenciales lineales ordinarias como modelos en el análisis de datos funcionales supera a los modelos de regresión funcional lineal, una ecuación diferencial ayuda a explicar mejor el comportamiento de los datos ya que se cuenta con las derivadas de la función $y(t)$ que se ajusta a los datos. Este tipo de análisis recibe el nombre de Análisis Diferencial Principal (*ADP*) de datos. Con el *ADP* se puede hacer el análisis de datos funcionales ruidosos, ya que se puede capturar cualquier rasgo de la curva $y(t)$ o la variación de los rasgos entre varias curvas $y_i(t)$.

Una ecuación diferencial puede capturar la forma de la curva para un dato funcional, también puede modelar la variación entre las observaciones cuando $N > 1$. En el análisis diferencial principal, se tienen dos objetivos, el primero es poder modelar un conjunto de datos por medio de una ecuación diferencial y luego buscar en el espacio m dimensional

de sus soluciones la que mejor nos estime el comportamiento del conjunto de los datos dados. El segundo objetivo es poder ver cuánta variación funcional puede ser explicada entre los múltiples procesos realizados. Así los modelos lineales y la descomposición de varianza surgen en este tipo de análisis.

Antes de realizar cualquier análisis diferencial principal, se recuerdan algunos conceptos de las ecuaciones diferenciales ordinarias que nos serán útiles.

5.1. Ecuaciones diferenciales con coeficientes constantes para datos funcionales

Una ecuación diferencial lineal de orden m homogénea con coeficientes constantes es como sigue

$$\frac{d^m y(t)}{dt^m} + \beta_{m-1} \frac{d^{m-1} y(t)}{dt^{m-1}} + \dots + \beta_1 \frac{dy(t)}{dt} + \beta_0 y(t) = 0$$

por ser de orden m cuenta con m soluciones linealmente independientes, las cuales forman una base, es decir, forman el espacio m dimensional de soluciones para esta ecuación.

Por conocimientos previos, se sabe que es más fácil trabajar con ecuaciones diferenciales donde β es constante, de igual forma en el *ADP* es más fácil trabajar con este tipo de ecuaciones diferenciales.

Un ejemplo de una ecuación diferencial con coeficientes constantes se tiene en un sistema de entrada-salida, como el que se muestra a continuación. El sistema de entrada-salida (5.1) tiene una función de entrada $u(t)$ que modifica de alguna manera la función de

salida $y(t)$

$$Dy(t) = -\beta y(t) + \alpha u(t) + \epsilon(t) \quad (5.1)$$

que es un modelo funcional lineal, en el cual la variable dependiente es la derivada de la función de salida $y(t)$, y las dos variables independientes son $y(t)$ y la función de entrada $u(t)$. En esta ecuación se consideran los coeficientes β y α constantes. La función ϵ es el ruido asumido en el modelo, es decir la variación ignorada en los datos funcionales. Este sistema se puede hacer más simple si quitamos la función $u(t)$, y generalmente se hace cuando se quiere modelar el comportamiento de la función $y(t)$ y sus derivadas sin considerar cualquier influencia externa

$$Dy(t) = -\beta y(t) + \epsilon(t)$$

de esta forma se tiene una ecuación homogénea. Cuando la función $u(t)$ esta presente en el sistema, se le da el nombre de función de forzamiento. Una solución de esta ecuación homogénea con $\epsilon = 0$ está dada por

$$y_0(t) = Ce^{-\beta t}$$

para alguna constante $C \neq 0$. Mientras que la solución para esta ecuación con $u(t)$ presente y $\epsilon = 0$, está dada por

$$y(t) = Ce^{-\beta t} + \alpha \int e^{-\beta(t-s)} u(s) ds.$$

el sistema (5.1) puede ser escrito como un operador diferencial de orden uno

$$Ly(t) = \beta y(t) + Dy(t) - \alpha u(t) - \epsilon(t)$$

luego la solución $y(t)$ encontrada cuando $\epsilon(t) = 0$, es solución del operador diferencial si y solo si $Ly(t) = 0$.

5.2. Ecuaciones diferenciales con coeficientes variables para datos funcionales

Cuando los coeficientes son variables lo que se tiene es la siguiente expresión

$$\frac{d^m y(t)}{dt^m} + \beta_{m-1}(t) \frac{d^{m-1} y(t)}{dt^{m-1}} + \dots + \beta_1(t) \frac{dy(t)}{dt} + \beta_0(t) y(t) = 0$$

donde $\beta_0(t), \beta_1(t), \dots, \beta_{m-1}(t)$ son funciones analíticas. Recordemos que una función $f(t)$ es analítica para t_0 si su serie de Taylor alrededor de t_0

$$\sum_{m=0}^{\infty} \frac{f^{(m)}(t_0)(t-t_0)^m}{m!}$$

converge a $f(t)$ en los alrededores de t_0 .

Un ejemplo de una ecuación diferencial con coeficientes variables es

$$Dy(t) = -\beta(t)y(t) + \alpha(t)u(t) + \epsilon(t)$$

para encontrar la solución de esta ecuación suponemos que $\epsilon(t) = 0$

$$Dy(t) + \beta(t)y(t) = \alpha(t)u(t)$$

la cual tiene por solución a

$$\begin{aligned} x(t) &= ce^{-\int \beta(t)dt} + e^{-\int \beta(t)dt} \int e^{\int \beta(t)dt} \alpha(t)u(t)dt \\ &= ce^{-\int_0^t \beta(s)ds} + \int_0^t \alpha(s)u(s) \frac{e^{\int_0^t \beta(s)ds}}{e^{\int_0^s \beta(t)dt}} ds \end{aligned}$$

En el análisis diferencial principal los parámetros funcionales $\beta_0(t), \beta_1(t), \dots, \beta_{m-1}(t)$ serán generados por medio de funciones base *spline*, más adelante explicaremos la metodología para su generación.

5.3. Estimación de operadores diferenciales en el análisis de datos funcionales

Toda ecuación diferencial ordinaria

$$\frac{d^m y(t)}{dt^m} + \beta_{m-1}(t) \frac{d^{m-1} y(t)}{dt^{m-1}} + \cdots + \beta_1(t) \frac{dy(t)}{dt} + \beta_0(t) y(t) = 0 \quad (5.2)$$

puede ser expresada de la forma

$$D^m y(t) + \beta_{m-1}(t) D^{m-1} y(t) + \cdots + \beta_1 D y(t) + \beta_0 y(t) = 0$$

simplificando

$$(D^m + \beta_{m-1}(t) D^{m-1} + \cdots + \beta_1 D + \beta_0) y(t) = 0 \quad (5.3)$$

denotando $L = D^m + \beta_{m-1}(t) D^{m-1} + \cdots + \beta_1 D + \beta_0$, la ecuación (5.2) se reduce a

$$Ly(t) = 0.$$

Por definición sabemos que si $\xi_1, \xi_2, \dots, \xi_m$ son m soluciones de $Ly(t) = 0$, entonces $c_1 \xi_1 + c_2 \xi_2 + \cdots + c_m \xi_m$ es también una solución de $Ly(t) = 0$.

Resumiendo, un operador diferencial lineal de orden m es de la forma

$$Ly(t) = \sum_{l=0}^{m-1} \beta_l D^l y(t) + D^m y(t)$$

que no es más que una ecuación diferencial homogénea de orden m reacomodada.

En el análisis diferencial principal, nuestra tarea es identificar un operador diferencial lineal de orden m

$$L = \beta_0 I + \dots + \beta_{m-1} D^{m-1} + D^m$$

y su ecuación diferencial homogénea asociada

$$D^m y(t) = -\beta_0 x(t) + \dots + \beta_{m-1} D^{m-1} y(t)$$

cuando se tiene un conjunto de datos cuyo comportamiento puede ser modelado por una función $y(t)$, entonces a dicha función podemos aplicarle L , obteniendo así $Ly(t) = f(t)$; por lo general lo más conveniente es tener $f(t) = 0$, pero no siempre es así, dicha función $f(t)$ recibe el nombre de *función forzada* o también conocida como *residual funcional*. Cuando se tiene un conjunto de N observaciones funcionales $y_i(t)$ entonces se tiene un posible conjunto de covariables funcionales asociadas f_i , $i = 1, \dots, N$. Debemos tener en mente que cuando modelamos un conjunto de datos funcionales, estos pueden ser modelados por medio de un sistema de ecuaciones diferenciales, el cual puede contener múltiples funcionales forzadas y además las ecuaciones diferenciales pueden no ser lineales.

En el *ADP*, como ya comentábamos, primero consideremos el caso donde la ecuación diferencial es homogénea, es decir, donde la función forzada $f(t)$ no está presente. Al empezar con el *ADP* sólo se cuenta con la función $y(t)$ estimada directamente de los datos observados, entonces para poder plantear el operador L necesitamos estimar los $m - 1$ parámetros funcionales β_l $l = 0, \dots, m - 1$ claro, que algunos de estos parámetros pueden ser fijos, es decir que en el intervalo τ de la función $y(t)$, el coeficiente β_l sea una constante funcional para toda t de τ .

En el *ADP* lo que buscamos finalmente es que el operador L aniquile tanto como sea posible a la función $y(t)$, y esto se logra cuando se determina con exactitud el orden del operador y definir adecuadamente los coeficientes funcionales, ya que estos pueden ser

o todos funcionales variables o funcionales constantes o de ambas clases.

La ecuación (5.3) también puede ser escrita como

$$D^m y(t) = - \sum_{l=0}^{m-1} \beta_l(t) D^l y(t)$$

y se obtiene cuando hacemos $\alpha(t)u(t) = 0$ y $\epsilon(t) = 0$.

Las funciones que satisfacen la ecuación homogénea $Ly(t) = 0$, son consideradas extremadamente suaves en el sentido de que las escogemos para ignorar cualquier componente de variabilidad.

Generalmente cuando se está trabajando con datos funcionales, es necesario cuidar ciertos aspectos que se cuidan en todo análisis de regresión múltiple. Por ejemplo, supongamos que tenemos un conjunto de datos discretos $y_{ij}, i = 1, \dots, N, j = 1, \dots, n$ generados por el modelo

$$y_j = y(t_j) + \epsilon_j,$$

donde, como en previos capítulos, $y(t)$ es alguna función suavizada que deseamos estimar por medio de una regresión no paramétrica, y ϵ_j es el error asumido con distribución independiente para todo j con media cero y varianza finita.

Usando el criterio de suavizamiento general, tenemos

$$PENSSE_\lambda(\hat{y}) = n^{-1} \sum_j^n [y_j - \hat{y}(t_j)]^2 + \lambda \int [\beta \hat{y}(t) + D\hat{y}(t)]^2 dt$$

para algún operador diferencial L . Si escogemos \hat{y} para que minimice $PENSSE_\lambda$, entonces la integral del sesgo al cuadrado es

$$(\text{sesgo})^2(\hat{y}) = \left\{ \int E[\hat{y}(t) - y(t)] dt \right\}^2$$

que no puede exceder la $\int (Ly)^2(t)dt$. Esto es útil, por que si escogemos a L tal que aproxime $Ly(t) = 0$, entonces el sesgo probablemente sea pequeño. Entonces esto nos permite darle valores grandes al parámetro de suavizamiento λ , dejando la varianza pequeña sin introducir demasiado sesgo. También podemos calcular valores pequeños para la integral del error cuadrado medio

$$IMSE(\hat{y}) = \int E[\hat{y}(t) - y(t)]^2 dt$$

entonces

$$IMSE(\hat{y}) = Bias^2(\hat{y}) + var(\hat{y})$$

donde

$$Var(\hat{y}) = \int E \{ \hat{y}(t) - E[\hat{y}(t)] \}^2 dt.$$

Por definición sabemos que un operador diferencial lineal L de grado m , tiene m soluciones ξ_r linealmente, donde $r = 1, \dots, m$ independientes, las cuales hacen $L\xi_r = 0$. El conjunto de todas las funciones ξ_r para el cual $L\xi_r = 0$, es llamado el espacio nulo de L y se denota con $\ker L$, las funciones ξ_r forman una base para este espacio. Esto significa, que podemos usar el operador diferencial lineal L para hacer una partición de la variabilidad funcional, en el sentido que $Ly(t)$ divide a $y(t)$ en dos partes, la primer parte consiste en que $y(t)$ puede ser expresado en términos de una combinación lineal del espacio nulo de funciones ξ_r y el segundo es ortogonal a esas funciones. Dado que existe información importante en las derivadas de la función $y(t)$ es importante explorarla en la variación particionada. Cabe mencionar que trabajar con derivadas, en general es más difícil pues $Dy(t)$ tiene en general

más curvatura y variabilidad que $y(t)$ y además en ocasiones las derivadas tienen que ser más suavizadas que la función original, pero aun y con todo esto el *ADP* es más completo.

Dada la ecuación

$$y(t) = \sum_{k=1}^K a_k \phi_k(t) \quad (5.4)$$

la cual modela algún fenómeno descrito por algún conjunto de datos discretos, lo que se busca es un operador diferencial L de orden m tal que

$$L(y(t)) = 0,$$

la tarea es calcular las funciones de peso $\beta_0, \beta_1, \dots, \beta_{m-1}$. Una forma de encontrarlos es por medio de la expresión

$$\beta = -W^{-1} D^m \alpha$$

donde

$$\alpha = \begin{pmatrix} a_k(t)\phi_k \\ a_{k-1}(t)\phi_{k-1} \\ \cdot \\ \cdot \\ a_1(t)\phi_1 \end{pmatrix}$$

y

$$W(t) = \begin{pmatrix} \alpha(t) & D\alpha(t) & \dots & D^{m-1}\alpha(t) \end{pmatrix} \cdot$$

La matriz $W(t)$ tiene que ser invertible, es decir, su determinante distinto de cero sobre τ . Entonces encontrar el vector de funciones de peso $\beta = (\beta_0(t), \beta_1(t), \dots, \beta_{m-1}(t))^t$ tiene que ver con resolver el siguiente sistema de m ecuaciones lineales

$$W(t)\beta(t) = -D^{-m}\alpha(t).$$

Después de estimar el vector β necesitamos estimar las funciones ξ_r que satisfagan $L\xi_r = 0$. Para poder estimar dichas funciones, se define de forma general la siguiente ecuación diferencial homogénea generada con el vector β y la función $y(t)$

$$\begin{aligned} D^m y(t) &= -\beta_0(t)y(t) - \beta_1(t)Dy(t) - \dots - \beta_{m-1}(t)D^{m-1}y(t) \\ &= -\sum_{l=0}^{m-1} \beta_l(t)D^l y(t) \end{aligned} \quad (5.5)$$

luego usando el algoritmo de Runge-Kutta y estableciendo las condiciones iniciales se resuelve la ecuación (5.5). Generalmente las condiciones iniciales están dados en un vector de longitud $B_0 y(t)$ de dimensión m , dicho vector recibe el nombre de *operador inicial* u *operador de restricciones*.

$$B_0 y(t) = \begin{bmatrix} y(t_0) \\ Dy(t_0) \\ \cdot \\ \cdot \\ \cdot \\ D^{m-1}y(t_0) \end{bmatrix}$$

El operador de restricciones $\mathbf{B}_0 y(t)$ es para especificar las m piezas de información acerca de $y(t)$ que necesitamos para identificar una función específica u como la solución única tal que $Lu = 0$. Al definir $\mathbf{B}_0 y(t)$ podemos encontrar m soluciones linealmente independientes ξ_i que pueden ser usadas como una base para obtener todas las posibles soluciones del operador L .

Generalmente toda ecuación diferencial cuenta con ciertas restricciones iniciales antes de ser resuelta, este conjunto de restricciones, forma el espacio $\ker B$. Las siguientes definiciones y teoremas nos explican este concepto.

Definition 4 $V = \ker L \oplus \ker B$, donde V es el espacio de las soluciones de una ecuación diferencial, si y solo si $v \in V$ se expresa de manera única como $v = \xi + e$ tal que $\xi \in \ker L$ y $e \in \ker B$, donde $\ker L$ y $\ker B$ son subespacios de V y $L\xi = 0$, $Be = 0$. Además $v = \xi + e$ se puede expresar de manera única si y solo si $\ker L \cap \ker B = \{0\}$ (Ramsay y Silverman, 2005).

Definition 5 Se dice que un espacio vectorial V es la suma directa de W_1 y W_2 expresada como $V = W_1 \oplus W_2$, si W_1 y W_2 son subespacios de V tal que $W_1 \cap W_2 = \{0\}$ y $V = W_1 + W_2$ (Friedberg et al. 1982).

Theorem 6 Sean W_1 y W_2 subespacios de un espacio vectorial V . Entonces V es la suma directa de W_1 y W_2 si y solo si cada elemento de V puede ser escrito de manera única como $y_1(t) + y_2(t)$ donde $y_1(t) \in W_1$ y $y_2(t) \in W_2$ (Friedberg et al. 1982).

Theorem 7 Sean V y W espacios vectoriales y sea $T : V \rightarrow W$ lineal. Entonces T es uno-a-uno si y solo si $N(T) = 0$ (Friedberg et al. 1982).

Asumiendo que el operador de restricciones cumple la condición de ortogonalidad $\ker L \cap \ker B = \{0\}$, se puede definir una familia de producto interno como sigue

$$(y, y)_{B,L} = (By(t))'(By(t)) + \int (Ly)(t)(Ly)(t)dt$$

con la correspondiente norma

$$\|y\|_{B,L}^2 = (By(t))'(By(t)) + \int (Ly)^2(t)dt,$$

la condición $\ker L \cap \ker B = \{0\}$ asegura que esta expresión sea realmente una norma.

Suponiendo que tenemos y_{i1}, \dots, y_{in} valores discretos con $i = 1, \dots, N$ entonces se generan varias funciones $y_i(t)$, si es así el método de mínimos cuadrados en el *ADP* se define como el criterio de ajuste de la suma de las normas al cuadrado del *residual función* $Ly_i(t)$

$$SSE_{ADP}(L | y(t)) = \sum_{i=1}^N \int [Ly_i(t)]^2 dt = \sum_{i=1}^N \|Ly_i(t)\|^2.$$

Cuando una función forzada es una función de entrada en el modelo, entonces tenemos la ecuación no homogénea

$$Ly_i(t) = f_i, \quad i = 1, \dots, N,$$

y el criterio de ajuste de mínimos cuadrados está dado por

$$SSE_{ADP}(L | y, f) = \sum_{i=1}^N \int [Ly_i(t) - f_i(t)]^2 dt = \sum_{i=1}^N \|Ly_i - f_i\|^2.$$

Para juzgar la efectividad del operador diferencial L en aniquilar la variación en los y_i , se verifica las gráficas de las funciones empíricas forzadas $Ly_i(t)$, si estas funciones son

pequeñas y principalmente ruidosas, entonces podemos tener confianza de que la ecuación diferencial está haciendo un buen trabajo con la representación funcional de los datos. En este tipo de análisis como en todos los demás, tenemos hipótesis a cumplir, una hipótesis para el modelo diferencial sería suponer que $\beta_0 = \dots = \beta_{m-1} = 0$, los $D^m y_i(t)$ son las funciones forzadas empíricas correspondientes a esta hipótesis nula, entonces podemos comparar el tamaño de las $Ly_i(t)$ con estas derivadas y poder juzgar la calidad del modelo. Aunque estrictamente hablando la tarea de *ADP* no es aproximar las curvas originales (esto es una tarea para análisis de componentes principales), no obstante podemos preguntarnos si las soluciones del operador sirven para este propósito, es decir, hacer una aproximación de mínimos cuadrados en términos de las soluciones de la ecuación $Ly_i(t) = 0$.

Otra manera de estimar las funciones de peso β_l de la ecuación diferencial generada con las funciones $y_i(t)$ es por el siguiente criterio: criterio de ajuste de *puntos apropiados*

$$PSSE_L(t) = \sum_i [Ly_i(t) - f_i(t)]^2 = \sum_i \left[\sum_{l=0}^m \beta_l(t) D^l y_i(t) - f_i(t) \right]^2 \quad (5.6)$$

donde $\beta_m(t) = 1$ para todo t . Si t se fija entonces lo anterior es un simple ajuste de mínimos cuadrados, y se define un vector de coeficientes de dimensión m como

$$\beta(t) = (\beta_0(t), \dots, \beta_{m-1}(t))'$$

también se define la matriz de diseño \mathbf{Z} de *puntos apropiados* de $N \times (m + 1)$ con renglones de la forma

$$\mathbf{z}_i(t) = \{-y_i(t), \dots, -D^{m-1}y_i(t), f_i(t)\}$$

y por último se define la variable dependiente con el vector \mathbf{y} de dimensión N con elementos

$$y_i(t) = D^m y_i(t).$$

Con estas definiciones podemos expresar el criterio de ajuste (5.6) en términos matriciales como

$$PSSE_L(t) = [\mathbf{y}(t) - \mathbf{Z}(t)\boldsymbol{\beta}(t)]'[\mathbf{y}(t) - \mathbf{Z}(t)\boldsymbol{\beta}(t)].$$

Manteniendo a t fija, la solución al minimizar el criterio de mínimos cuadrados $PSSE_L(t)$ con respecto a los valores de $\beta_i(t)$ es

$$\boldsymbol{\beta}(t) = [\mathbf{Z}(t)'\mathbf{Z}(t)]^{-1}\mathbf{Z}(t)'\mathbf{y}(t).$$

La existencia de los valores de $\boldsymbol{\beta}(t)$ depende si el determinante de $\mathbf{Z}(t)'\mathbf{Z}(t)$ es distinto de cero para todo t . Asumir que el determinante es distinto de cero, es equivalente a suponer que $\mathbf{Z}(t)$ es de rango completo para todo t . Si $m = 1$ podemos escribir a β_0 como

$$\beta_0 = - \sum_i y_i(t)(Dy_i)(t) / \sum_i y_i^2(t)$$

y las condiciones de rango completo requieren que para cada valor de t alguna $y_i(t)$ sea distinta de cero.

Aislar las singularidades de $\mathbf{Z}(t)'\mathbf{Z}(t)$ implica aislar las singularidades de una o más de las funciones de peso β_i , estas singularidades pueden ser evitadas si usamos funciones de peso β_i lo suficientemente suaves. El método anterior sólo sirve si el número de observaciones funcionales N excede el número de columnas de la matriz de diseño.

Cuando necesitamos ajustar una ecuación diferencial a una observación funcional se usa el modelo concurrente.

El objetivo del *ADP* es minimizar la norma $\|Ly\|$ de las funciones forzadas asociadas con el operador diferencial estimado, entonces la calidad del modelo puede variar sobre el dominio de τ , es necesario evaluar el modelo en términos de los puntos apropiados de la suma de los errores al cuadrado $PSSE_L(t)$.

En el modelo lineal debemos comparar $PSSE_L(t)$ con el modelo teórico y sus funciones de peso asociadas ω_j

$$PSSE_0(t) = \sum_i \left[\sum_{l=0}^{m-1} \omega_l(t)(D^l y_i)(t) + (D^m y_i)(t) \right]^2$$

cuando no hay un modelo teórico a la mano, podemos usar $\omega_l = 0$, para que la comparación sea simple con la suma de cuadrados de $D^m y_i$. De aquí podemos examinar el cuadrado de la función de correlación múltiple del criterio de ajuste de *puntos apropiados*

$$RSQ(t) = \frac{PSSE_0(t) - PSSE_L(t)}{PSSE_0(t)}$$

entonces el cociente F de este criterio es

$$FRATIO(t) = \frac{(PSSE_0(t) - PSSE_L(t))/m}{PSSE_0(t)/(N - m)}.$$

Una vez que se encuentra el operador L podemos definir m funciones linealmente independientes ξ_1, \dots, ξ_m que expandan el espacio nulo de L , tal que cualquier función $y(t)$ que satisfaga $Ly(t) = 0$ puede ser expresada como combinación lineal de estas funciones ξ_r . Entonces L puede ser escogido para hacer $Ly_i(t)$ lo más pequeño posible, podemos esperar obtener una buena aproximación de $y_i(t)$ expandiéndola en términos de las ξ_r . Por lo tanto el análisis diferencial principal está definido como la identificación del operador diferencial L que minimice el criterio de mínimos cuadrados SSE_{ADP} .

5.4. Aplicaciones

Las aplicaciones las hacemos a dos conjuntos de datos, el primero de ellos corresponde a datos de tecnología de conservación de alimentos: datos que representan la pérdida de peso en ajos mexicanos, el segundo de ellos al Indicador Global de la Actividad Económica Mexicana (1993 – 2005).

Para ambos conjunto el procedimiento del *ADP* es el siguiente:

- 1 Se ajusta una función $y(t)$ a los datos discretos.
- 2 Como lo que se busca es una ecuación diferencial lineal de la forma $\beta_0 y(t) + \dots + \beta_{m-1} D^{m-1} y(t) + D^m y(t)$ debemos estimar los coeficientes β_l donde $l = 0, \dots, m - 1$, a partir de la función $y(t)$. A los coeficientes estimados los denotamos con $\hat{\beta}_l$ los cuales pueden ser variables o constantes.
- 3 Con los $\hat{\beta}_l$ se formular un operador diferencial de la forma $L = \hat{\beta}_0 I + \dots + \hat{\beta}_{m-1} D^{m-1} + D^m$, si este operador es el óptimo, entonces las $m - 1$ soluciones de este operador diferencial superan en estimación de variabilidad a la función $y(t)$ previamente ajustada y suavizada de los datos.

5.4.1. Conjunto de datos que representan la pérdida de peso en ajos mexicanos

A este conjunto de datos que representan la pérdida de peso de ajos mexicanos se le aplican 6 tratamientos:

- a) tratamiento 1 : $0^\circ C$

- b) tratamiento 2 : $20^{\circ}C$
- c) tratamiento 3 : $30^{\circ}C$
- d) tratamiento 4 : $5^{\circ}C$
- e) tratamiento 5 : $0^{\circ}C/70RH$
- f) tratamiento 6 : No control (temperatura no controlada)

En total se tienen tres réplicas para cada tratamiento denotamos con i a las réplicas y con q los tratamientos. Entonces tenemos $y_i(t)$, $i = 0, \dots, 18$ cada una de estas funciones.

A continuación se muestra el análisis diferencial estadístico, para este conjunto de datos:

La Figura 5.1 muestra las curvas de los datos escalares, se aprecia que la pérdida de peso de los ajos ocurre para los seis tratamientos, para la mayoría de los tratamientos la pérdida de peso es rápida.

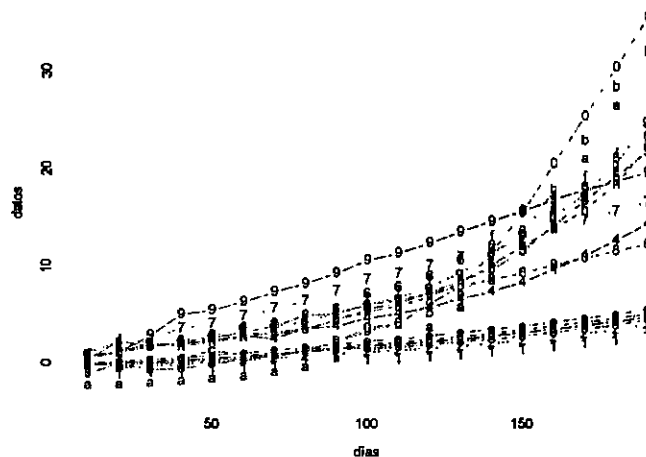


Figura 5.1: 18 curvas de los valores escalares.

En la Figura 5.2 tenemos las 18 representaciones funcionales $y_i(t)$ de los seis tratamientos, estas funciones, son las que mejor representan al conjunto de datos dado, obviamente estas funciones, ya están suavizadas. Los detalles de la estimación de estas funciones se dan en el código en *Splus*, que se muestra en el apéndice A.

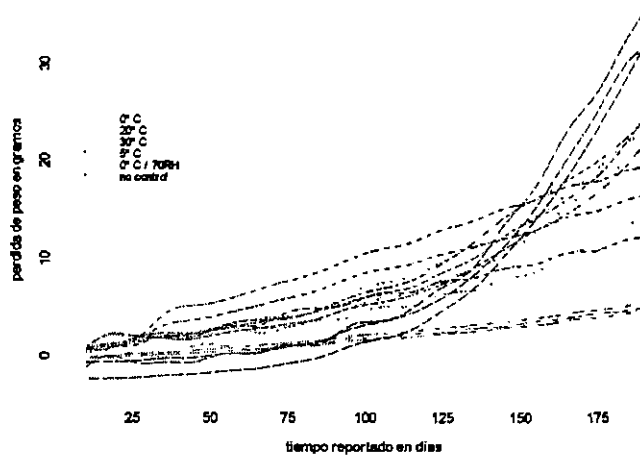


Figura 5.2: Los seis tratamientos con sus tres réplicas.

De los seis tratamientos que se tiene, vamos a considerar, a manera de ejemplo, sólo el análisis estadístico para el tratamiento seis. En la Figura 5.3 se muestra las tres réplicas funcionales del tratamiento seis, donde la temperatura para los ajos no es controlada.

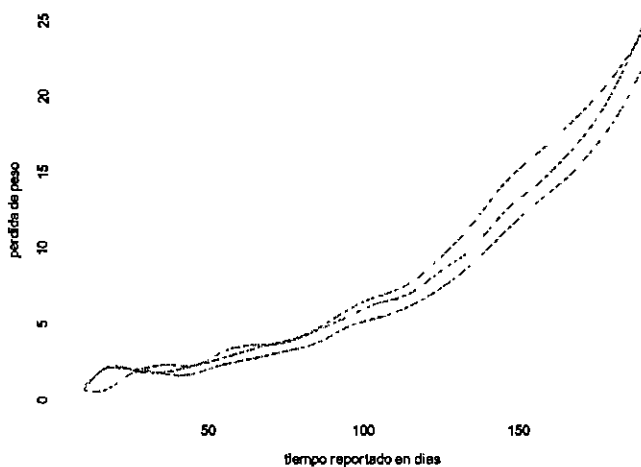


Figura 5.3: Réplicas funcionales del tratamiento seis.

En la Figura 5.4 se encuentran las gráficas de los parámetros funcionales estimados para las tres réplicas del tratamiento seis, el primer coeficiente es variable y los otros dos son funciones constantes.

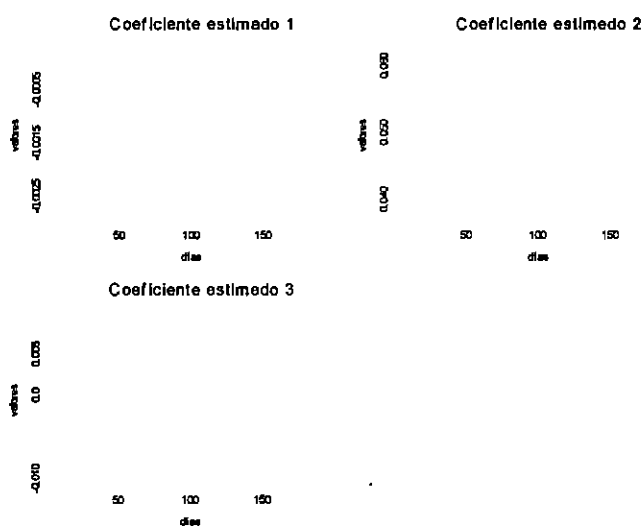


Figura 5.4: Parámetros funcionales estimados para el tratamiento seis.

Entonces la ecuación diferencial que se propone es de tercer orden. Es decir, dados $\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2$, definimos el operador

$$L = \hat{\beta}_0 I + \hat{\beta}_1 D + \hat{\beta}_2 D^2 + D^3$$

el cual aplicamos a cada una de las réplicas funcionales $y_i(t)$ del tratamiento seis, obteniendo así tres funciones forzadas o *residual funcional*

$$Lx_i(t) = f_i(t).$$

Ahora que ya se ha definido el operador diferencial, procedemos a hacer una estimación de las funciones $y_i(t)$ $i = 1, 2, 3$, es decir a encontrar las tres soluciones ξ_r linealmente independientes de este operador diferencial.

En la Figura 5.5 se muestran las tres soluciones estimadas ξ_r del operador diferencial $L = \hat{\beta}_0 I + \hat{\beta}_1 D + \hat{\beta}_2 D^2 + D^3$. Obsérvese cómo estas soluciones son parecidas a las funciones $y_i(t)$ del tratamiento seis.

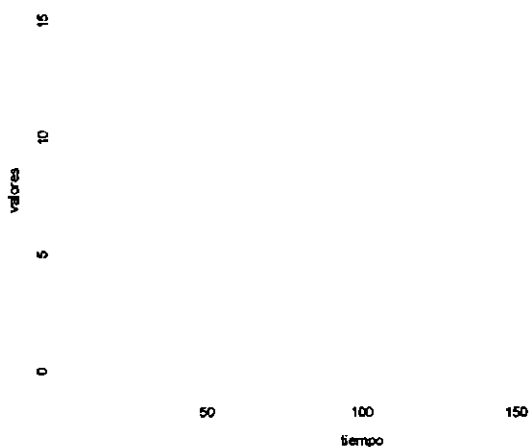


Figura 5.5: Soluciones estimadas para el tratamiento seis.

En la gráfica del lado derecho de la Figura 5.6 la línea color naranja es la función $y_1(t)$ y la línea de color verde es una combinación lineal de las tres soluciones estimadas ξ_r linealmente independientes; y la gráfica del lado izquierdo de esta Figura muestra el *residual funcional* $f_1(t)$. Por el comportamiento de la función $f_1(t)$ podemos decir que la ecuación diferencial de orden tres hace un buen trabajo con la estimación de la función $y_1(t)$.

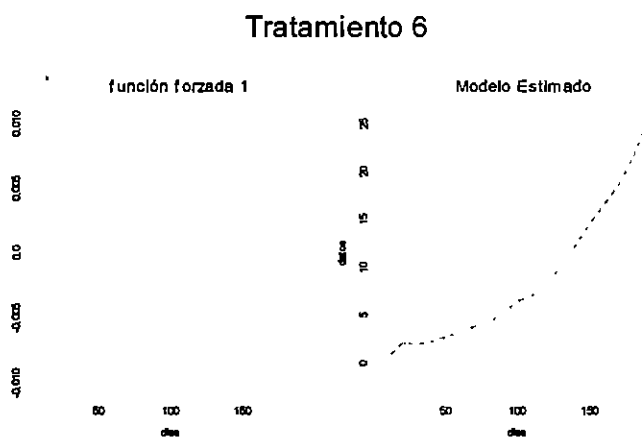


Figura 5.6: Ajuste de la función $\hat{y}_1(t)$ para la primer réplica.

Las Figuras 5.7 y 5.8 muestran las gráficas de las otras dos réplicas del tratamiento seis, y de igual forma se observa que el operador de orden tres

$$L = \hat{\beta}_0 I + \hat{\beta} D + \hat{\beta}_2 D^2 + D^3$$

genera las funciones $\hat{y}_2(t)$ y $\hat{y}_3(t)$ que mejoran las funciones $y_1(t)$ y $y_2(t)$ previamente estimadas de los datos, respectivamente.

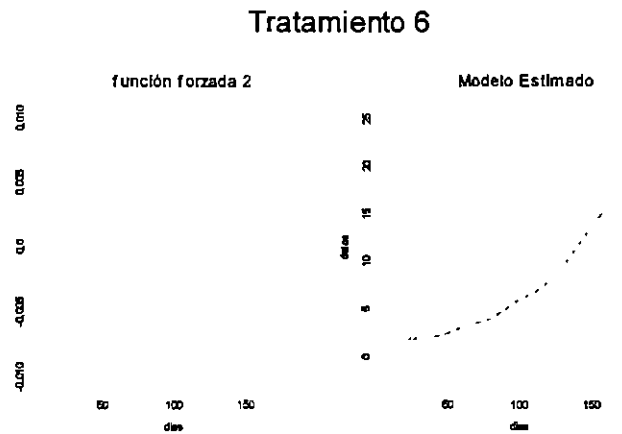


Figura 5.7: Ajuste de la función $\hat{y}_2(t)$ para la segunda réplica.

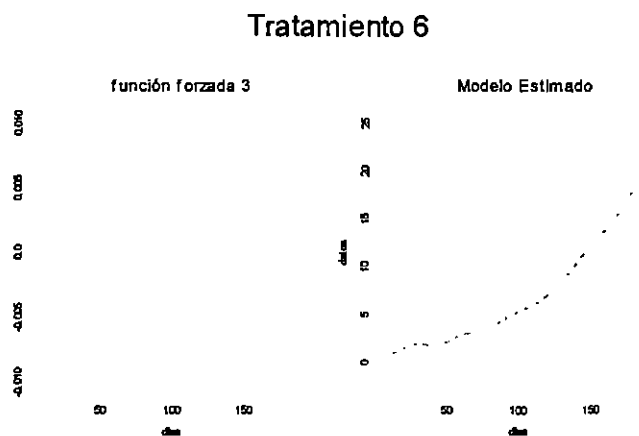


Figura 5.8: Ajuste de la función $\hat{y}_3(t)$ para la tercer réplica.

Para los tratamientos 1, 2, 4 sólo la estimación de β_2 es una constante funcional, mientras que para tratamiento 5 sólo β_0 es variable funcional. Para el tratamiento 3 tenemos que usar un operador diferencial de orden 2 y hacer a β_0 variable. Lo interesante de estos seis tratamientos con el *ADP* es el análisis estadístico que podemos hacer entre los tratamientos.

Por ejemplo para los operadores diferenciales de las réplicas del tratamiento 1 y las réplicas del tratamiento 6

$$L\hat{y}_{i1}(t) = \hat{\beta}_0(t)\hat{y}_{i1}(t) + \hat{\beta}_1(t)D\hat{y}_{i1}(t) + 0,019D^2\hat{y}_{i1}(t) + D^3\hat{y}_{i1}(t)$$

$$L\hat{y}_{i6}(t) = \hat{\beta}_0\hat{y}_{i6}(t) + 0,05D\hat{y}_{i6}(t) - 0,001D^2\hat{y}_{i6}(t) + D^3\hat{y}_{i6}(t)$$

se puede ver que la pérdida de peso para el primer tratamiento, es mínima, comparada con la pérdida de peso del sexto tratamiento. Dadas estas particularidades, los coeficientes de estos respectivos operadores tienen sus diferencias, por ejemplo la velocidad de la pérdida de peso para ambos tratamientos, está ponderada de manera diferente indicando que para el sexto tratamiento, la velocidad de pérdida de peso siempre será constante y positiva, mientras que para el primer tratamiento, la velocidad de pérdida de peso, será más controlable. En cuanto a la ponderación de la aceleración de las funciones estimadas, para el sexto tratamiento la aceleración será ponderada negativa y constante, indicando, que en un momento dado, ya no se seguirá perdiendo peso dado que los ajos se secaran. Mientras que para el primer tratamiento la ponderación de la aceleración de la función estimada será positiva y constante, indicando que la aceleración de la pérdida de peso seguirá así por más tiempo.

5.4.2. Datos del Indicador Global de la Actividad Económica Mexicana (1993-2005)

El conjunto de datos del *IGAEM*, está formado por 157 datos pertenecientes a los años 1993 al 2005.

La función $x(t)$ como ya sabemos es una expansión de funciones base ϕ_k y esta expansión está acompañada de coeficientes c_k . Para este ejemplo los ϕ_k son segmentos de polinomios de grado siete. Haciendo el *ADP* encontramos que el comportamiento de este conjunto de datos puede ser modelado por la ecuación diferencial lineal homogénea

$$\beta_0 + \beta_1 Dy(t) + \beta_2 D^2 y(t) + D^3 y(t) = 0.$$

para hacer una estimación de los parámetros funcionales β_l donde $l = 0, 1, 2$ se usa nuevamente la función *pda.fd*. Si algún coeficiente funcional β_l estimado es variable, entonces este es una función analítica que depende del tiempo t donde $t \in [1993, 2005]$ y se define como

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

La Figura 5.9 muestra los valores discretos de la variable que indica el *IGAEM*, en total son 156 como ya se comentaba.

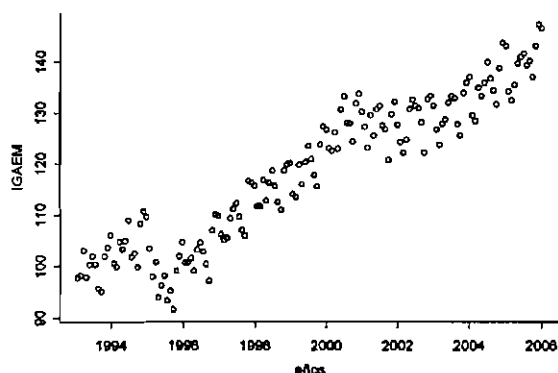


Figura 5.9: 156 datos del *IGAEM*.

La Figura 5.10, muestra a la función $y(t)$, la cual ya esta previamente suavizada (sí convienen a los objetivos del investigador), con esta función se estiman las funciones $\beta_i(t)$.

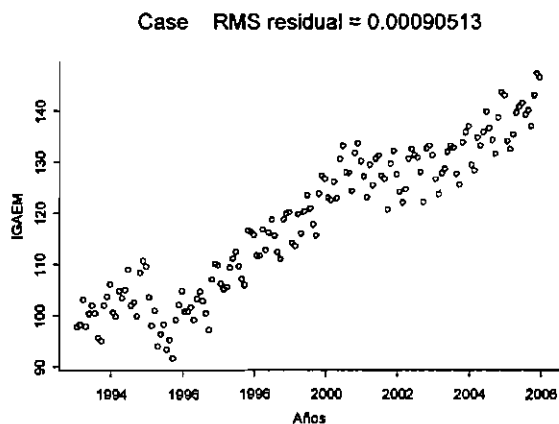


Figura 5.10: Función $y(t)$ estimada del *IGAEM*.

En la Figura 5.11 se muestran las tres gráficas de los parámetros funcionales $\hat{\beta}_i$ para la ecuación diferencial $\beta_0 + \beta_1 Dy(t) + \beta_2 D^2y(t) + D^3y(t) = 0$, sólo el coeficiente $\hat{\beta}_3$ es una función constante.

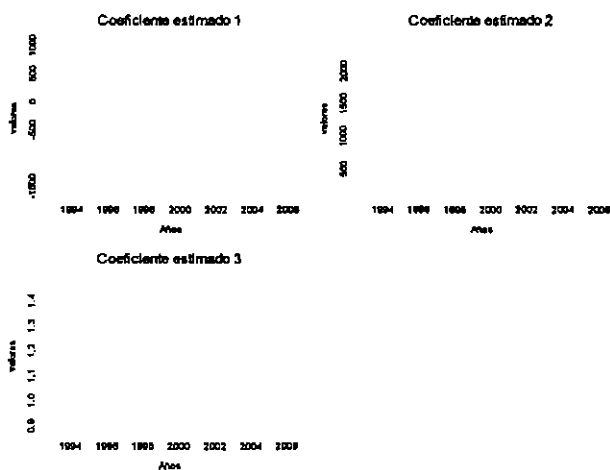


Figura 5.11: Parámetros funcionales del *IGAEM*.

En la Figura 5.12 se encuentran las tres soluciones ξ_r , una de ellas es la que más explica el comportamientos de los datos, mientras que las otras dos se pegan casi al cero.

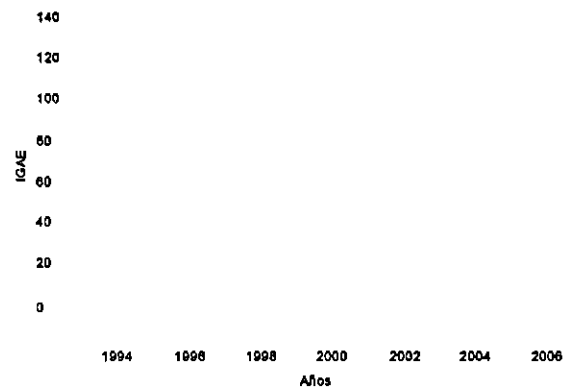


Figura 5.12: Las tres soluciones $\hat{\beta}_i(t)$ del operador diferencial lineal planteado.

La Figura 5.13 muestra la gráfica de la función $f(t) = Ly(t)$. Dado que la gráfica de la función $f(t)$ oscila alrededor del cero y de manera estable, entonces el operador $L = \hat{\beta}_0 I + \hat{\beta}_1 D + \hat{\beta}_2 D^2 + D^3$ está haciendo un buen trabajo con el análisis estadístico de los datos en cuestión.

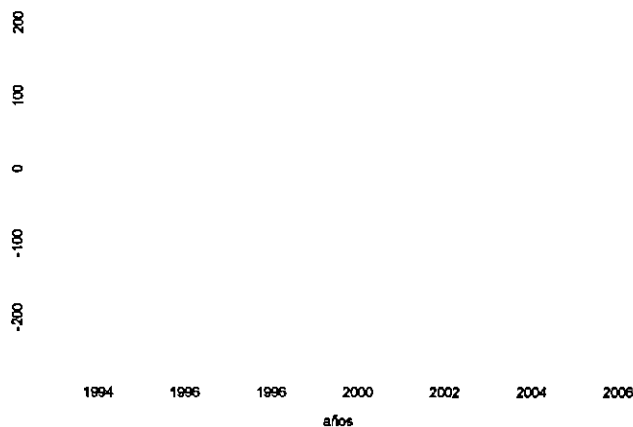


Figura 5.13: Residual funcional $f(t)$.

En la Figura 5.14 se muestra la función $\hat{y}(t)$ generada por las soluciones ξ_r , no se puede negar que la estimación es buena, con lo cual concluimos que el Indicador Global de la Actividad Económica Mexicana puede ser modelado por la ecuación $\hat{y}(t)$. $L = \hat{\beta}_0 I + \hat{\beta}_1 D + \hat{\beta}_2 D^2 + D^3$.

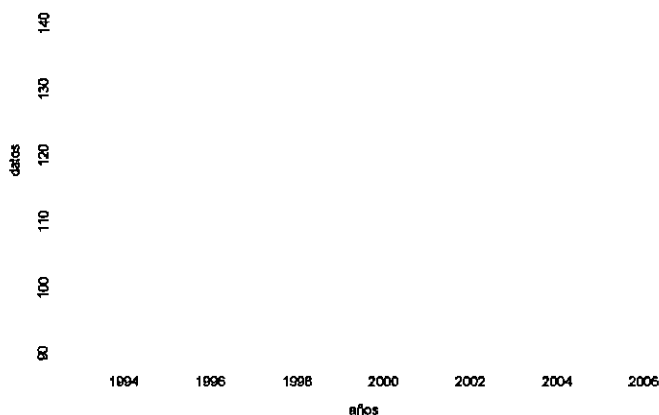


Figura 5.14: Modelo diferencial funcional estimado vs los datos del *IGAEM*.

Capítulo 6

Conclusiones

Son amplios los horizontes y muy prometedores cuando se visualiza el análisis estadístico desde el punto de vista funcional. Claro está que el estudio de los datos en su contexto y de tipo de funciones que existen en la teoría son muy necesarios para obtener resultados coherentes con la realidad, y evitar de esta forma caer en interpretaciones erróneas. La teoría y la realidad fue una constante en este trabajo ya que siempre se buscó que la teoría estadística y funcional ayudara con el análisis de datos que encontramos en nuestra vida cotidiana.

Como en este trabajo lo que se buscó siempre fue aterrizar la teoría con algunas aplicaciones podemos concluir de éstas, que el análisis estadístico funcional obtenido fue más completo. Pues se logró determinar el comportamiento de los datos a través de la función subyacente de éstos. Y como bien sabemos: si se determina el comportamiento funcional de algún fenómeno, entonces se conocemos mejor el fenómeno. Así pues las aplicaciones propuestas en este trabajo fueron elegidas con la intención de aplicar la teoría que proponen

Ramsay y Silverman (2005), y podemos concluir felizmente, que siempre se obtuvieron resultados coherentes con la realidad.

El trabajar con teoría de análisis de datos funcionales nos dejó experiencia para interpretar resultados funcionales sobre el contexto de la realidad.

El análisis estadístico con teoría funcional es muy prometedor y contundente, y más que generar de alguna manera algún conflicto, genera herramienta poderosa para analizar datos extraídos de la realidad que sería difícil decir cualquier cosa de ellos. Por lo tanto podemos concluir que la teoría funcional es una herramienta poderosa en la rama de la estadística.

Apéndice A

Código base en Splus para las aplicaciones

A continuación, se muestra cada uno de los códigos en *Splus* para las aplicaciones de los temas expuestos. Los siguientes códigos se basan en los códigos desarrollados por Ramsay y Silverman, que se encuentran en la dirección electrónica <http://www.functionaldata.org>.

Código base en Splus correspondiente a la aplicación para *el Modelo lineal funcional para respuesta funcional y covariables escalares* de Ramsay y Silverman (2005)

El conjunto de datos, para este modelo fueron proporcionados por la Doctora Silvia Amaya.

```
# *****
```

```
# Análisis funcional con muestras de quesos
```

```

# *****

#::::::::::::::::::::: Datos de entrada

respY<-as.matrix(quesos,173,16,byrow=T)

matrizdise<-(

1.00,1.13,1.01,1.17,-0.89,1.14,1.32,-1.01,1.18,-0.90,-1.04

1.00,-1.11,1.01,-0.93,-0.89,-1.12,1.03,0.99,-0.94,-0.90,0.83

1.00,-0.94,-1.14,1.17,-0.89,1.08,-1.10,0.84,-1.34,1.02,-1.04

1.00,0.10,-0.07,-0.05,-0.89,-0.01,-0.01,-0.09,0.00,0.06,0.05

1.00,-1.27,-1.14,-0.93,-0.89,1.46,1.18,1.14,1.06,1.02,0.83

1.00,-0.94,1.01,1.17,-0.89,-0.95,-1.10,0.84,1.18,-0.90,-1.04

1.00,1.13,-1.14,-0.93,-0.89,-1.30,-1.05,-1.01,1.06,1.02,0.83

1.00,1.13,1.01,-0.93,-0.89,1.14,-1.05,-1.01,-0.94,-0.90,0.83

1.00,-0.94,-0.07,1.17,0.21,0.06,-1.10,-0.19,-0.08,-0.01,0.24

1.00,0.10,-1.14,-1.28,0.21,-0.11,-0.12,0.02,1.46,-0.24,-0.26

1.00,1.13,1.01,-0.05,0.21,1.14,-0.06,0.23,-0.06,0.21,-0.01

1.00,1.13,-0.07,-0.93,1.30,-0.08,-1.05,1.48,0.06,-0.09,-1.21

1.00,-0.94,1.01,-0.93,1.30,-0.95,0.87,-1.23,-0.94,1.32,-1.21

1.00,1.13,-1.14,1.17,1.30,-1.30,1.32,1.48,-1.34,-1.49,1.52

1.00,-0.94,-1.14,-0.05,1.30,1.08,0.05,-1.23,0.06,-1.49,-0.07

1.00,0.10,1.01,1.17,1.30,0.10,0.11,0.13,1.18,1.32,1.52)

tiempo<-c(9,18,28,38,48,58,68,78,90,98,109,118,130,138,149,158,168,

178,188,198,209,218,228,238,248,258,268,278,288,298,308,318,328,

```

```

338,348,358,368,378,388,398,408,418,428,438,448,458,468,478,488,
498,508,518,528,538,548,558,568,578,588,598,608,618,628,638,648,
658,668,678,688,698,708,718,728,738,748,758,768,778,788,798,808,
818,828,838,848,858,868,878,888,898,908,918,928,938,948,958,968,
978,988,998,1008,1018,1028,1038,1048,1058,1068,1078,1088,1098,
1108,1118,1128,1138,1148,1158,1168,1178,1188,1198,1208,1218,
1228,1238,1248,1258,1268,1278,1288,1298,1308,1318,1328,1338,
1348,1358,1368,1378,1388,1398,1408,1418,1428,1438,1448,1458,
1468,1478,1488,1498,1508,1518,1528,1538,1548,1558,1568,1578,
1588,1598,1608,1618,1628,1638,1648,1658,1668,1678,1688,1698,
1708,1718,1728)

rango<-c(9,1728)

efecto<-c("media", "pH", "%prot", "%hum", "tempext", "pH*prot",
"pH*hum", "pH*tempext", "prot*hum", "prot*tempext", "hum*tempext")

#:::::::::::::::::::::::::: Gráfica de las respuestas yi

par(mfrow=c(2,4),mar=c(5,5,4,2)+1.2+2,pty="m")

for(i in 1:16){

plot(tiempo,respY[,i],type="p",col=3,cex=1,

xlab="tiempo",ylab=paste(Respuesta",i))}

#:::::::::::::::::::::::::: Representación funcional de las respuestas escalares

norden<-3

numYbasis<-171

```

```
Ybasis<-create.bspline.basis(rango,numYbasis,norden)
```

```
respYfd<-data2fd(respY,tiempo,Ybasis)
```

```
plot.fd(respYfd,main=Respuestas funcionales)
```

```
smallbas<-eval.basis(tiempo,Ybasis)
```

```
matrizMap1<-solve(crossprod(smallbas))%*%t(smallbas)
```

matrizMap1 es una matriz de 171×173 que contiene la matriz de mapeo $\mathbf{S}_{\phi, \lambda_y} = \Phi(\Phi' \Phi + \lambda_y \mathbf{R}_y)^{-1} \Phi'$, la cual convierte los datos de la matriz $Y_{N \times n}$ a coeficientes de la matriz $\mathbf{C}_{N \times k_y}$, la matriz de mapeo $\mathbf{S}_{\phi, \lambda_y}$ es de $n \times k_y$.

```
coef <- respYfd$coefs
```

```
coef17 <- cbind(coef,matrix(0,numYbasis,1))
```

```
respYfd$coefs <- coef17
```

```
matX <-as.matrix(matrizdise,16,11,byrow=T)
```

matrizdese es la matriz de diseño

```
z17 <- matrix(1,1,11)
```

```
z17[1] <- 0
```

```
matX <- rbind(matX , z17)
```

```
p<-11
```

```
Xfdlist<-vector("list",p)
```

```
for (j in 1:p)Xfdlist[[j]]<-matX[j]
```

```
#:::::::::::::::::::::::::: Representación funcional del vector  $\beta$ 
```

```
norder <- 4
```

```
nbbasis <- 40
```



```

bbasis <- create.bspline.basis(rango, nbbasis, norder)

betafd <- fd(matrix(0,nbbasis,1),bbasis)

estimate <- T

lambda <-1

betafdPar <- fdPar(betafd,2,lambda, estimate)

betalist <- vector("list",p)

for(j in 1:p) betalist[[j]] <-betafdPar

#::::::::::::::::::::: Cálculo de los parámetros funcionales y de las respuestas
funcionales estimadas

fRegressList <- fRegress(respYfd, Xfdlist, betalist)

#:::::::::::::::::::::Gráfica de los parámetros funcionales estimados

betaestlist<-fRegressList$betaestlist

par(mfrow=c(2,4))

for(jin1:p)

{

betaestParfdj<-betaestlist[[j]]

plot(betaestParfdj$fd,xab="tiempo",ylab="diámetro",

main=efecto[j],xlim=c(0,1728),ylim=c(-100,150))

}

#::::::::::::::::::::: Gráfica de las respuestas funcionales estimadas

yhatfdoj <- fRegressList$yhatfdoj

par(mfrow=c(1,1))

```

```
plot(yhatfdobj,main="Respuestas funcionales estimadas")
```

#:::::::::::::::::::::::::: Calculando algunas funciones estadísticas para valorar el ajuste de las funciones $\hat{y}_i(t)$

```
yhatmat <- eval.fd(tiempo, yhatfdobj)
```

La variable *yhatmat* almacena los valores de la matriz y_i .

```
yamat <- eval.fd(tiempo, respYfd)
```

La variables *yhatmat* almacena los valores de la matriz \hat{y}_i .

```
temprmat <- yamat[,1:16] - yhatmat[,1:16]
```

La variable *temprmat* contiene los valores $e_i = y_i - \hat{y}_i$.

```
SigmaE <- var(t(temprmat))
```

La variable *SigmaE* almacena los valores

$$s^2 = (N - K)^{-1} \sum_{i=1}^N (y_i - \hat{y}_i)^2 = (N - K)^{-1} \sum_{i=1}^N (e_i)^2$$

donde s^2 es un estimador insesgado de σ^2 y K , es el número de funciones bases seleccionado, para la estimación de las funciones $y_i(t)$, es decir

$$\text{Var}(\epsilon) = \widehat{\sum_e} = \hat{\sigma}^2 \mathbf{I}.$$

Cuando se tiene una N , lo suficiente mente grande, entonces

$$\text{Var}(\epsilon) = \widehat{\sum_e} = (N - 1)^{-1} \mathbf{E}/\mathbf{E}$$

donde \mathbf{E} , es la matriz de residuales de $N \times n$

#:::::::::::::::::::::::::: Gráfica de las desviaciones estándar de los errores.

```
par(mfrow=c(1,1), mar=c(5,5,4,2)+cexval+2, pty="m")
```

```
stddevE <- sqrt(diag(SigmaE))
```

La variable *stddevE* es un vector con 173 valores obtenidos de la raíz cuadrada, de los valores que se encuentran en la diagonal de la matriz $\hat{\sigma}^2\mathbf{I}$.

```
plot(tiempo, stddevE, type="l", cex=1.2,xlab="tiempo",
```

```
ylab="valores",main="Desviación estándar de los residuales")
```

```
#:::::::::::::::::::::::::::: Repetimos la regresión para calcular los intervalos de confi-
```

anza de los coeficientes funcionales

```
stderrList <- fRegress.stderr(fRegressList, matrizMap1, SigmaE)
```

```
betastderrlist <- stderrList$betastderrlist
```

La variable *betastderrlist* contiene los errores estándar de las funciones $\hat{\beta}_t(t)$ estimados de la expresión:

$$\text{Var}(\hat{\beta}(t)) = \theta(t)\text{Var}(\hat{\mathbf{B}})\theta'(t)$$

```
#:::::::::::::::::::::::::::: Gráficas de los errores estándar de los parámetros funcionales  $\hat{\beta}_t(t)$ .
```

```
par(mfrow=c(2,4), mar=c(5,5,4,2)+cexval+2, pty="m")
```

```
for(j in 1:p) {
```

```
betastderrj <- eval.fd(tiempo, betastderrlist[[j]])
```

```
plot(tiempo, betastderrj, cex=1.2,
```

```
type="l",lty=1, xlab="tiempo", ylab=Reg. Coeff.",main=efecto[j])}
```

```
#:::::::::::::::::::::::::::: Gráficas de los parámetros funcionales con sus intervalos de
```

confianza

Los intervalos de confianza son calculados al agregar y sustraer un múltiplo de los

errores estándar, es decir la raíz cuadrada de la varianza muestral para el ajuste actual. Un intervalo puntual con el 95 % de confianza para una función $\beta_l(t)$, está dado por

$$\left(\hat{\beta}_l(t) - 2 \times \sqrt{\text{Var}(\hat{\beta}_l(t))}, \hat{\beta}_l(t) + 2 \times \sqrt{\text{Var}(\hat{\beta}_l(t))} \right).$$

donde $\text{Var}(\hat{\beta}_l(t)) = \theta_l(t)' \text{Var}(\hat{\mathbf{B}}_l) \theta_l(t)$. Estos intervalos son puntuales porque reflejan la región de confianza para valores fijos de t , y no para toda la curva $\beta_l(t)$.

```
par(mfrow=c(2,4))
```

```
for (j in 1:p){
```

```
  betafdParj <- betaestlist[[j]]
```

```
  betafdj <- betafdParj$fd
```

```
  betaj <- eval.fd(tiempo, betafdj)
```

La variables *betaj* almacena los valores de cada $\hat{\beta}_l(t)$, para $l = 1, \dots, q$

```
  betastderrj <- eval.fd(tiempo, betastderrlist[[j]])
```

La variable *betastderrj* almacena los valores de los errores estándar $e_{\hat{\beta}_l}$ para cada $\hat{\beta}_l(t)$.

```
  matplot(tiempo, cbind(betaj, betaj+2*betastderrj, betaj-2*betastderrj),
```

```
  type="l",lty=c(1,4,4), xlab="tiempo", ylab="Reg. Coeff.",main=efecto[j])}
```

Código base en Splus, correspondiente a la aplicación para *el Modelo lineal funcional para respuesta funcional y covariables escalares* de Abramovich et al. (2004)

El conjunto de datos, para este modelo fueron proporcionados por la Doctora Silvia Amaya.

El siguiente código está programado en el software matlab 6,5, y se basa en el desarrollado por Felix Abramovich y colaboradores.

Lo que se muestra a continuación son los códigos de las funciones: *testfixed.m* que hace las pruebas de hipótesis, y *queso8datos.m* que manipula los datos para hacer una transformación discreta *wavelet* de los tratamientos.

Parámetros de entrada y salida para la función *testfixed*:

```
function[phi_star,pval_star,Tj,V0,zed_alpha]
= testfixed([dY,type,eta,js,jn,alpha,wtype,par]
```

Parámetros de entrada:

- dY es el vector de observaciones.
- $type$ denota el tipo de prueba. Es decir, *Smooth* si se asume que f está contenido en una bola de Besov $B_{p,q}^s(C)$ con $p \geq 2$ y *NonSmooth* si se asume que f está contenido en una bola de Besov $B_{p,q}^s(C)$ con $1 \leq p < 2$
- $eta = \eta$ es la desviación estándar del ruido.
- js es el número de niveles *wavelets* , donde la estadística Tj considera significativos.

- j_η es el número de niveles de *wavelets*, donde la estadística Tj no considera significativos.
- *alpha* nivel de significancia.
- *wtype* tipo de familia *wavelet*.
- *par* es un entero, este parámetro está relacionado con

si $\text{phi_star} = 0 \Rightarrow$ se acepta H_0

si $\text{phi_star} = 1 \Rightarrow$ se rechaza H_0

- *pval_star* es el valor p .
- Tj es la prueba estadística (para la prueba *Smooth* o *NonSmooth*).
- $V0$ es la desviación estándar del estadístico Tj bajo la hipótesis nula.
- *zed_alpha* es el valor crítico para $Tj/V0$ para el nivel un significancia α .

Código de la función *testfixed*:

```
if nargin < 7
```

```
    Tipo de familia:
```

```
    wtype = 'Coiflet';
```

```
    par = 3;
```

```
end
```

```
if nargin < 6
```

```
    alpha = 0.05;
```

```
end
```

```
% Parámetros iniciales
```

```
Tj = 0;
```

```
Qj = 0;
```

```
v02 = 0;
```

```
w02 = 0;
```

```
if alpha == 0.05
```

```
zed_alpha = 1.645;
```

```
else
```

```
z = 1.0;
```

```
p = 1 - normcdf(z);
```

La función *normcdf()*, calcula la función de distribución acumulativa de la normal.

```
while p > alpha & z < 3.5;
```

```
z = z+0.001;
```

```
p = 1-normcdf(z);
```

```
end
```

```
zed_alpha = z;
```

```
end
```

```
qmf = makeonfilter(wtype,par);
```

La función *makeonfilter()*, Generate una transformación *wavelet* ortonormal.

```
wdY = fwt_po(dY,0,qmf);
```

La función *fwt_po()*, efectúa una transformación *wavelet* periódica u ortogonal.

```
if strcmp(type,'Smooth')
    fprintf('PERFORMING TRUNCATED TEST \n')
```

Los siguientes 8 renglones de código, muestra la ejecución de prueba *asintótica minimax adaptativa*, considerando los niveles de resolución de $j(s)$, en el caso $j \in \mathcal{J}_-$ y tomando $p \geq 2$.

```
for lev = 0:js-1
    Tj = Tj + sum( wdY(dyad(lev)).^2 - eta^2);
    La variable Tj equivale a la expresión  $T(j(s)) = \sum_{j=0}^{j(s)-1} S_j$ , y  $S_j$ .
end
V0 = sqrt(2 * eta^4 * 2^js);
    La variable V0 equivale a la expresión  $v_0^2(j(s)) = 2\eta^4 2^{j(s)}$ 
phi_star = (Tj/V0 > zed_alpha);
pval_star = 1 - normcdf(Tj/V0);
elseif strcmp(type,'NonSmooth')
    fprintf('PERFORMING THRESHOLDED TEST \n')
```

El resto del código, corresponde también a la ejecución de prueba *asintótica minimax adaptativa*, pero considerando ahora los niveles de resolución de $j(s)$, cuando $j \in \mathcal{J}_+$ y tomando $1 \leq p < 2$.

```
for lev = 0:js-1
    Tj = Tj + sum( wdY(dyad(lev)).^2 - eta^2);
end
v02 = 2 * eta^4 * 2^js;
```



```
sq = 0;
```

```
for j = js:jn-1
```

```
lambda = 4*sqrt((lev-js+8)*log(2));
```

La variable *lambda* en el renglón anterior equivale a la expresión

$$\lambda_j = 4\sqrt{j - j(s) + 8 \ln(2)}.$$

```
blambda = sqrt(2/pi)*exp(-lambda^2/2)*lambda+2*(1-normcdf(lambda));
```

```
dlambda = sqrt(2/pi)*exp(-lambda^2/2)*lambda*(3 +lambda^2)+  
6*(1-normcdf(lambda));
```

```
dummy = hardthresh(wdY(dyad(lev)),lambda*eta);
```

Básicamente lo que se efectúa en la variable *dummy* es el cálculo de la expresión

$Y_{j,\kappa} \mathbf{1}(|Y_{j,\kappa}| > \eta\lambda)$, donde $Y_{j,\kappa} \int_0^1 \psi_{j,\kappa}(t) dt$.

```
Qj = Qj + sum( dummy.^2 - eta^2 * blambda);
```

La variable *Qj* equivale a la expresión $Q(j(s)) = \sum_{j=j(s)}^{j_\eta-1} S_j(\lambda_j)$.

```
sq = sq + 2^lev * eta^4 * (dlambda - blambda^2);
```

La variable *sq* está efectuando el cálculo de la varianza del parámetro $Q(j(s))$, es

decir $w_0^2(j(s)) = \eta^4 \sum_{j=j(s)}^{j_\eta-1} 2^j d(\lambda_j)$.

```
end
```

```
w02 = sq;
```

```
Tj = Tj+Qj;
```

```
V0 = sqrt(v02+w02);
```

```
phi_star = (Tj/V0 > zed_alpha);
```

```
pval_star = 1-normcdf(Tj/V0);
```

end

Código de la función queso8datos.m

closeall

clearall

%—DefaultParameters—

wavelet='Coiflet';

parameter=1;

filt = MakeONFilter(wavelet, parameter);

%————- Loading the Data —————

load C:\MATLAB6p5\toolbox\wavelab\queso.dat

load C:\MATLAB6p5\toolbox\wavelab\disematrix.dat

load C:\MATLAB6p5\toolbox\wavelab\timeT.dat

%————— Selección de datos equiespaciados en el tiempo

Ye= zeros(16,86);

queso=queso';

te=zeros(1,86);

j=2;

k=1;

while j <=173

Ye(:,k)=queso(:,j);

te(k)=timeT(j);

k=k+1;

```
j=j+2;
```

```
end
```

El valor de n , que representa el total de datos a ser considerados, tiene que ser una potencia de 2, es decir $n = 2^J$, donde J denota en número de componentes de la multirresolución.

```
%----- Dimensión del vector time y la matriz Y
```

```
Y=zeros(16,16);
```

```
time=zeros(1,16);
```

```
N=2^4;
```

```
j=1;
```

```
k=1;
```

```
while j<=86
```

```
Y(:,k)=Ye(:,j);
```

```
time(k)=te(j);
```

```
k=k+1;
```

```
j=j+6;
```

```
end
```

```
Y(:,16)=Ye(:,86);
```

```
time(16)=te(86);
```

```
%----- Cálculo del vector Beta -----
```

Lo que se efectúa en los siguientes tres renglones es el cálculo de $B = (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'\mathbf{Y}$

```
B = zeros(11,N);
```

```

X = disematrix(:,:);
B =inv(X'*X)*(X)'*Y;

subplot(2,2,1)

plot(time,B),title('\fontsize{8} Curvas de los 10 efectos y la media global');

%————— Varianza robusta de los errores —————

sigrobust=[];

sigma=0;

for i = 1:16

wt=FWT_PO(Y(i,:),0,flt);

subplot(2,2,2)

PlotWaveCoeff(wt,0,0)

subplot(2,2,3)

PlotWaveCoeff(wt,2,0)

finest_level = wt(dyad(log2(N)-1));

addme = 1/0.6745 * median(abs(finest_level));

sigrobust=[sigrobust addme];

end

sigma= mean(sigrobust);

```

La función *PlotWaveCoeff* grafica los coeficientes $c_{J-1,\kappa}$, $d_{j,\kappa}$.

La función *FWT_PO*, regresa, la transformación *wavelet* de cada uno de los y_1 .

Finalmente en *finest_level* se encuentran los coeficientes *wavelet* para hacer una estimación robusta de $\hat{\sigma}$.

```

vcB=(inv(((X')*X)));
vB=[];
for i = 1:11
    mii=0;
    mii=sqrt(vcB(i,i));
    vB=[vB mii];
end
ES=(sqrt(N/2))*sigma*(vB);

```

En la variable `cons_ii` se encuentran los m_{ii} , en total son 11 valores. En la variable `ES`, se encuentran los 11 valores para η correspondiente a cada uno de los $\hat{\beta}_l(t)$.

En los siguientes renglones se hacen las pruebas de hipótesis

$$H_0 : \sum_{j=0}^{\infty} \sum_{\kappa=0}^{2^j-1} d_{j,\kappa}^2 = 0 \text{ vs } H_1 : \sum_{j=0}^{\infty} \sum_{\kappa=0}^{2^j-1} d_{j,\kappa}^2 \neq 0$$

para cada uno de los $\hat{\beta}_l(t)$.

```

%===== Prueba de Hipotesis
disp(['-----']);
fprintf('Hipotesis \n')
disp(['*****']);
for k=1:11
    Tj=0;
    V0=0;
    phi_star=0;
    pval_star=0;

```

```

Tj_dum=0;

V0_dum=0;

phi_dum=0;

pval_dum=0;

ESBi=0;

ESBi=ES(1,k);

Bi=zeros(1,N);

Bi(1,:)=B(k,:);

[phi_dum, pval_dum, Tj_dum, V0_dum,zed_alpha]=
testfixed(Bi,'NonSmooth',ESBi,js,j_eta,0.05,wavelet,parameter);

Tj=Tj_dum;

V0=V0_dum;

disp(['*****']);

pval_star = 1-normcdf(Tj/V0);

phi_star = (Tj > V0*zed_alpha);

fprintf(' Beta_ %2.0f \n',k-1)

fprintf('ESTIMATE OF THE STANDARD DEVIATION = %f \n',ESBi);

disp(['H_o: Beta_ i=0: T(j(s))+ Q(j(s))=' num2str(Tj)'];

```

```

Critical value=' num2str(V0*zed_alpha));
disp([' p-value=' num2str(pval_star)]);
disp([' phi^*=' num2str(phi_star)]);
disp(['_____']);
end

```

La función *normcdf*, calcula la *cdf* de la expresión T_j/V_0 , la variable *pval_star* da el *p-value*.

En la variable *phi_star*, se efectúa la prueba

$$\phi^* = \{ \mathbf{1} \{ T(j(s)) > v_0(j(s))z_{1-\alpha} \} \text{ si } p \geq 2 \},$$

el significado de esta variable es, como ya se comentaba:

si *phi_star* = 0 \Rightarrow se acepta H_0

si *phi_star* = 1 \Rightarrow se rechaza H_0

Código base en Splus, correspondiente a la aplicación para *el Modelo lineal funcional para respuesta funcional y covariables funcionales* donde β depende de un sólo parámetro t

Los dos conjuntos de datos para este modelo lineal funcional, se obtuvieron de las direcciones electrónicas:

<http://data.giss.nasa.gov/gistemp/graphs/fig.A2.txt>

http://cdiac.esd.ornl.gov/ftp/trends/emis_mon/emis_mon_usatotal.dat

*# ******

```

# Análisis funcional con datos de la temperatura global de la tierra
# *****
#::::::::::::::::::::: Datos de entrada
tempGlobal <- as.matrix(Tabla.tem,16,1)
dioxGlobal <- as.matrix(anualco2,16,1)
#::::::::::::::::::::: Los valores de estos vectores son centrados sobre sus prome-
dios

tempGlobal<-tempGlobal-mean(tempGlobal[1:16])
dioxGlobal<-dioxGlobal-mean(dioxGlobal[1:16])
etiqueta<-c("1988-2003")
ndatos<-16
durtime<-(0:(ndatos-1))
tiempo<-(0:(ndatos-1))+1988
intervalo<-c(1988,2003)
#::::::::::::::::::::: Gráfica de la variable respuesta y
par(mfrow=c(1,1), mar=c(5,5,4,2)+3.2, pty="m")
plot(tiempo,tempGlobal,main="Temperatura global anual de 1998 al 2003")
#::::::::::::::::::::: Representación funcional del vector y
norder <- 4
numtempbasis <- 13
tempbasis <- create.bspline.basis(intervalo, numtempbasis, norder)
TGAfd <- data2fd(tempGlobal, tiempo,tempbasis)

```



```
#:::::::::::::::::::: Suavizamiento de la respuesta funcional
```

```
lambda <- 0
```

```
fdParobj <- fdPar(tempbasis , 2, lambda)
```

```
smoothList <- smooth.basis(tiempo, tempGlobal, fdParobj)
```

```
Map1<-smoothList$y2cMap
```

Map1 es la primera matriz de mapeo que convierte los datos del vector *y* a coeficientes de un vector *c*, es decir $y = C\phi^{-1}$, entonces $Map1 = \phi^{-1}$

```
#:::::::::::::::::::: Gráfica del dióxido de carbono global anual emitido.
```

```
par(mfrow=c(1,1), mar=c(5,5,4,2)+3.2, pty="m")
```

```
plot(tiempo,dioxiGlobal,main="Dióxido de carbono global anual del 1998 al 2003")
```

```
#:::::::::::::::::::: Representación funcional del vector z.
```

```
norder <- 4
```

```
numDioxbasis <- 12
```

```
Dioxbasis <- create.bspline.basis(intervalo, numDioxbasis, norder)
```

```
DCGAfd <- data2fd(dioxiGlobal ,tiempo,Dioxbasis)
```

```
names(DCGAfd$fdnames) = c("tiempo", , "datos")
```

#:::::::::::::::::::: Expansión en funciones base, para poder obtener una estimación funcional del coeficiente $\beta_1(t)$

```
p<-1
```

```
norder <- 4
```

```
nbetabasis <- 13
```

```
betabasis <- create.bspline.basis(intervalo, nbetabasis, norder)
```

```

betafd <- fd(matrix(0,nbetabasis,p), betabasis)

estimate <- T

lambda <- 0

betafdPar <- fdPar(betafd,2,lambda,estimate)

betalist <-list(betafdPar)

xfdlista <- list(DCGAfd )

#::::::::::::::::::::: Regresión funcional

fRegressList <- fRegress(TGAfd, xfdlista, betalist)

betaestlist <- fRegressList$betaestlist

yhatfdoj <- fRegressList$yhatfdoj

#::::::::::::::::::::: Gráfica del coeficiente funcional de la regresión

par(mfrow=c(1,1))

betafdPar <- betaestlist[[1]]

betafd <- betafdPar$fd

plot(betafd, xlab=".años", ylab="beta(t)",main="Coeficiente funcional de la regresión")

#::::::::::::::::::::: Gráfica de la respuesta funcional  $\hat{y}(t)$ 

par(mfrow=c(1,1),mar=c(5,5,4,2)+3.2,pty="m")

names(yhatfdoj$fdnames)=c("tiempo",,"datos")

plotfit.fd(tempGlobal,tiempo,yhatfdoj,cex=1.2)

#::::::::::::::::::::: Cálculo del vector de residuales a partir de  $\epsilon(t) = y(t) - \hat{y}(t)$ 

yhatmat <- eval.fd(tiempo, yhatfdoj)

```

```

ymat <- eval.fd(tiempo, TGAfd)

resid<- ymat - yhatmat

plot(tiempo,resid,xlab=".ños",ylab=residuales",main="Gráfica de los residuales")

#:::::::::::::::::::: Cálculo de algunas funciones estadísticas para valorar el
ajuste de la función  $\hat{y}(t)$ 

SigmaE <- mean(resid^2)

SigmaE <- SigmaE*diag(rep(1,16))

#:::::::::::::::::::: Repetimos la regresión para calcular los intervalos de confi-
anza para la función  $\beta(t)$ 

stderrList <- fRegress.stderr(fRegressList, Map1 ,SigmaE)

betastderrlist <- stderrList$betastderrlist

#:::::::::::::::::::: Gráfica del error estándar del coeficiente funcional  $\beta(t)$ 

par(mfrow=c(1,1))

betastderrfdj <- betastderrlist[[1]]

betastderrj <- eval.fd(tiempo, betastderrfdj)

plot(tiempo, betastderrj , type="l",lty=1, xlab="tiempo", ylab="Reg. Coeff.",
main=".Error estándar del coeficiente funcional  $\beta(t)$ ")

#:::::::::::::::::::: Gráficas del coeficiente funcional con su intervalo de confi-
anza puntual

par(mfrow=c(1,1))

betafdParj <- betaestlist[[1]]

betafdj <- betafdParj$fd

```

```

betaj <- eval.fd(tiempo, betafdj)

betastderrfdj <- betastderrlist[[1]]

betastderrj <- eval.fd(tiempo, betastderrfdj)

matplot(tiempo, cbind(betaj, betaj+2*betastderrj, betaj-2*betastderrj),
type="l",lty=c(1,4,4), xlab="tiempo", ylab="Reg. Coeff.",
main="intervalo de confianza para beta(t)")

```

Código base en Splus, correspondiente a la aplicación para *el Modelo lineal funcional para respuesta funcional y covariables funcionales* donde β depende dos veces del mismo parámetro t

Los dos conjuntos de datos para este modelo lineal funcional se obtuvieron de las direcciones electrónicas:

<http://data.giss.nasa.gov/gistemp/tabledata/GLB.Ts+dSST.txt>

http://cdiac.esd.ornl.gov/ftp/trends/emis_mon/emis_mon_usatotal.dat

```
# *****
```

```
# Análisis funcional con datos de la temperatura global mensual de la tierra y el
```

océano

```
# *****
```

```
#:.....: Datos de entrada
```

```
Temp.mensual.global<-as.matrix(Tabla.mensual.Tem,16,12)
```

```
Temp.mensual<-t(Temp.mensual.global)
```

```
Diox.Carbono <- as.matrix(Tabla.CO2,16,12)
```

```

Diox.Carbono <- t(Diox.Carbono )

tiempo <- seq(1,12,1)

intervalo <-c(1,12)

etiqueta<-c("1988","1989","1990","1991","1992","1993","1994","1995",
            "1996","1997","1998","1999","2000","2001","2002","2003")

#:::::::::::::::::::::::::: Representación funcional de las variables independientes

knots<-tiempo

norder <- 4

numDioxbasis <- length(knots)

Dioxbasis <- create.bspline.basis(intervalo, numDioxbasis, norder)

Dioxidofd <- data2fd(Diox.Carbono,tiempo,Dioxbasis)

par(mfrow=c(1,1), mar=c(5,5,4,2)+3.2, pty="m")

plot(Dioxidofd,xlab="meses", ylab="teragrams del dióxido de carbono de
los U.S.A",main="Funciones del CO2")

names(Dioxidofd$fdnames)=c("meses",,"teragrams del dióxido de carbono")

#:::::::::::::::::::::::::: Gráfica de las temperaturas globales mensuales de la tierra
y el océano

par(mfcol=c(2,2), mar=c(5,5,4,2)+3.2, pty="m")

matplot(tiempo ,Temp.mensual, type="b",cex=1.0,
xlab="meses",ylab="TGMTO",ylim=c(14,15),
main="Valores escalares de las temperaturas globales mensuales de la tierra y el
océano")

```

```

#..... Representación funcional las 16 variables dependientes,  $y_i =$ 
 $\sum_{k=1}^K c_k \phi_k(t)$ 

knots<-tiempo

norder <- 4

numTempbasis <- length(knots)

Tempbasis <- create.bspline.basis(intervalo, numTempbasis, norder)

TGMfd <- data2fd(Temp.mensual,tiempo,Tempbasis)

#..... Gráfica de las funciones  $y_i(t)$ 

par(mfrow=c(1,1), mar=c(5,5,4,2)+3.2, pty="m")

plot(TGMfd ,xlab="meses", ylab="temperatura en grados centígrados en escala
absoluta",

main="Funciones de las temperaturas globales mensuales de la tierra y el océano")

#..... Ajuste de la función al conjunto de datos que pertenecen al
año 2003.

names(TGMfd$fdnames)=c("meses", "Temperatura en °C")

```

Continuando con el análisis de regresión funcional tenemos que definir dos parámetros de suavizamiento λ para cada conjunto de funciones, estos son escogidos por el método de validación cruzada. Además se tiene que definir dos operadores diferenciales lineales $L_y = D(y(t))$ y $L_z = D(z(t))$ con argumento t para cada conjunto de funciones para penalice la rugosidad de éstas.

```

xLfdobj <- int2Lfd(2)

yLfdobj <- int2Lfd(0)

```

```
xlambda <- 0.1
```

```
ylambda <- 0.0025
```

```
wtvec <- matrix(1,16,1)
```

La variable *wtvec* define un vector de pesos para cada observación, *wtvec* es de longitud 16 conteniendo el escalar uno.

```
linmodstr <- linmod(Dioxidofd, TGMfd, wtvec, xLfdobj, yLfdobj, xlambda, ylambda)
```

```
afd <- linmodstr$alphafd
```

La variable *afd* es la función intersección o del término independiente.

```
bfd <- linmodstr$regfd
```

La variable *bfd* es la función de regresión bivariada conocida también como función Beta.

```
#::::::::::::::::::::: Gráfica de la función  $\alpha(t)$ 
```

```
plot(afd, xlab="meses", ylab="Función alfa", main="Función del término independiente")
```

```
#::::::::::::::::::::: Gráfica de la función Beta como superficie
```

```
bfdmat <- eval.bfd(tiempo, tiempo, bfd)
```

```
persp(tiempo, tiempo, bfdmat, xlab="mes(t)", ylab="mes(s)")
```

```
#::::::::::::::::::::: Gráficas de las funciones estimadas  $\hat{y}_i$ 
```

```
TGMhatfd <- linmodstr$yhatfd
```

```
plot(TGMhatfd, xlab="meses", ylab="valores",
```

```
main="Funciones de las temperatura globales mensuales estimadas")
```

#:::::::::::::::::::::::::: Código para graficar las funciones estimadas de cada y_i y el promedio global funcional de las funciones $z_i(s)$.

```
TGMmeanfd <- mean.fd(TGMfd)
```

La función *mean.fd* calcula la media funcional de las funciones *TGMfd*, es decir,

$$\overline{TGMfd}(t) = N^{-1} \sum_{i=1}^N TGMfd_i(t).$$

```
TGMhat0 <- eval.fd(tiempo, TGMmeanfd)
```

```
TGMmat <- eval.fd(tiempo, TGMfd )
```

En los dos renglones anteriores, se evalúan las funciones \hat{y}_i en el tiempo.

```
TGMhatmat <- eval.fd(tiempo,TGMhatfd)
```

```
plotrange <- c(min(TGMmat),max(TGMmat))
```

```
par(mfrow=c(2,3), mar=c(5,5,4,2)+3.2, pty="m")
```

```
for (i in 1:16){
```

```
  TGMi <- eval.fd(tiempo, TGMfd [i])
```

```
  TGMhati <- eval.fd(tiempo,TGMhatfd[i])
```

```
  SSE <- sum((TGMi-TGMhati)^2)
```

La variable *SSE* contiene la expresión: $SS_E = \sum_i^N \epsilon_i^2(t) = \sum_i^N \{\hat{y}_i(t) - y_i(t)\}^2$.

```
  SST <- sum((TGMi- TGMhat0)^2)
```

La variable *SST* contiene la expresión: $SS_T = \sum_i^N y_i^2 - N\bar{y}^2(t) = \sum_i^N \{y_i(t) - \bar{y}(t)\}^2$.

```
  RSQ <- (SST-SSE)/SST
```

La variable *RSQ* contiene la expresión: $R^2 = 1 - \frac{SS_E}{SS_T}$.

#:::::::::::::::::::::::::: Gráficas de las funciones \hat{y}_i vs sus promedios funcionales y

el valor del R^2


```

plot(tiempo,TGMmat[,i], type="l", lty=4, ylim=plotrange, xlab="meses",
ylab="temperatura", main=paste(etiqueta[i],R^2=",round(RSQ,3)),ask=T)

lines(tiempo, TGMhatmat[,i])

```

Código base en Splus, correspondiente a la aplicación para *el Modelo lineal funcional para respuesta escalar y covariables funcionales*

Los dos conjuntos de datos para este modelo lineal funcional se obtuvieron de las direcciones electrónicas:

<http://data.giss.nasa.gov/gistemp/graphs/Fig.A2.txt>

http://cdiac.esd.ornl.gov/ftp/trends/emis_mon/emis_mon_usatotal.dat

```

#*****

# Análisis funcional con datos de la temperatura global de la tierra

#*****

#::::::::::::::::::::: Datos de entrada

temperatura <- Tabla.tem[,1]

Diox.Carbono <- as.matrix(Tabla.CO2,16,12)

Diox.Carbono <- t(Diox.Carbono )

tiempo <- seq(1,12,1)

year <- seq(1988,2003,1)

intervalo <- c(1,12)

etiqueta<-c("1988","1989","1990","1991","1992","1993","1994","1995",
"1996","1997","1998","1999","2000","2001","2002","2003")

```

```

#:::::::::::::::::::::::::: Gráfica del vector y
cexval<-1.2

plot (year, temperatura, type="b", cex=1.0,
xlab=".años1988-2003",ylab="temperatura en °C",
main="Promedios de las temperaturas globales anuales",col=3)

#:::::::::::::::::::::::::: Gráfica del dióxido de carbono global mensual emitido.
matplot(tiempo ,Diox.Carbono, type="b",cex=1.0, xlab="meses",ylab="CO2",
ylim=c(350,600),main="Dióxido de Carbono emitido durante los años 1988-2003")

```

Lo que sigue es representar a la matriz **Z** en funciones base, en este caso, se decidió usar funciones *spline*, pues explican mejor el comportamiento de estos datos. El orden las funciones base será 3 y el número de funciones base se elige de acuerdo al comportamiento de los datos.

```

#:::::::::::::::::::::::::: Representación funcional del vector Z.

knots <- tiempo

norder <- 3

numDioxbasis <- length(knots)+norder-3

Dioxbasis <- create.bspline.basis(intervalo, numDioxbasis, norder)

Dioxidofd <- data2fd(Diox.Carbono,tiempo,Dioxbasis)

#:::::::::::::::::::::::::: Gráficas de las funciones CO2

par(mfrow=c(1,1))

plot.fd(Dioxidofd, xlab="meses", ylab="CO2",type="b",
main="Representación funcional del CO2")

```

Ahora bien como es necesario introducir la variable artificial $z_{j0} = 1$ para formar la primer columna de \mathbf{Z} , entonces se crea el vector *xfdlista* de longitud 2 para dar lugar a una función constante.

```
#:::::::::::::::::::: Creación funcional de la matriz Z
```

```
p <- 2
```

```
constantfd <- fd(matrix(1,1,16), create.constant.basis(intervalo))
```

```
xfdlista <- vector("list",2)
```

```
xfdlista[[1]] <- constantfd
```

```
xfdlista[[2]] <- Dioxidofd[1:16]
```

#:::::::::::::::::::: Para el vector β tenemos que crear un vector betalist de longitud 2 tal que en la primera posición fijemos una función constante para el término β_0 , y en la segunda posición las funciones del CO_2 .

```
betalist <- vector("list",2)
```

```
betafd1 <- fd(0, create.constant.basis(intervalo))
```

```
betafdPar1 <- fdPar(betafd1)
```

```
betalist[[1]] <- betafdPar1
```

```
knots <- tiempo
```

```
norder <- 3
```

```
nbetabasis <- length(knots)+norder-2
```

```
betabasis2 <- create.bspline.basis(intervalo, nbetabasis, norder).
```

```
betafd2 <- fd(matrix(0,nbetabasis,1), betabasis2)
```

```
lambda <- 10^3.5
```

```

betafdPar2 <- fdPar(betafd2, int2Lfd(1), lambda)

betalist[[2]] <- betafdPar2

#::::::::::::::::::::::::::::: Regresión funcional

fRegressList <- fRegress(temperatura, xfdlista, betalist)

betaestlist <- fRegressList$betaestlist

TempAnual <- fRegressList$yhatfdoj

TempAnual es un objeto con los valores  $\hat{y}_j$ 

alphafdPar <- betaestlist[[1]]

betaestlist[[1]] almacena la función constante  $\alpha$  del modelo  $y = \alpha + \int_0^n \mathbf{Z}(t)\beta(t)dt..$ 

alphafdPar$fd$coefs

#::::::::::::::::::::::::::::: Gráfica del coeficiente funcional estimado

betafdPar <- betaestlist[[2]]

betafd <- betafdPar$fd

par(mfrow=c(1,1), mar=c(5,5,4,2)+cexval+2, pty="m")

plot(betafd, cex=1.2,xlab="meses",ylab="temperatura en °C")

title(Coeficiente funcional estimado)

#::::::::::::::::::::::::::::: Gráfica del modelo ajustado

par(mfrow=c(1,1), mar=c(5,5,4,2)+cexval+2, pty="m")

plot(TempAnual, temperatura, type="p", cex=1.2)

lines(TempAnual, TempAnual, lty=4)

```

Para saber que tan bueno es nuestro modelo calculamos coeficiente de determinación para validar el modelo estimado, este está dado por:

$$R = +\sqrt{\frac{1 - SSE}{SSy}}$$

donde $SSE = \sum_j^n \varepsilon_j^2(t) = \sum_j^n \{\hat{y}_j(t) - y_j(t)\}^2$ y $SSy = \sum_j^n y_j^2 - n\bar{y}^2(t) = \sum_j^n \{y_j(t) - \bar{y}(t)\}^2$ tal que SSE mide la variación residual que queda sin explicar por el modelo ajustado y SSy es la suma total corregida de los cuadrados de y_j .

#.....: Cálculo del coeficiente de determinación.

```
covmat <- var(cbind(temperatura, TempAnual))
```

```
Rsqr <- covmat[1,2]^2/(covmat[1,1]*covmat[2,2])
```

Entonces $Rsqr = \frac{0,001025534}{0,001275748} = 0,80387$ lo cual nos está indicando que el modelo

es muy bueno.

Para poder calcular el intervalo de confianza para la función coeficientes, necesitamos hacer los siguientes cálculos:

```
resid <- temperatura - TempAnual
```

```
SigmaE <- mean(resid^2).
```

```
SigmaE <- SigmaE*diag(rep(1,16))
```

```
stderrList <- fRegress.stderr(fRegressList, NULL, SigmaE)
```

```
betastderrlist <- stderrList$betastderrlist
```

```
betastderr1 <- betastderrlist[[1]]
```

```
betastderr1$coefs
```

La variable `betastderr1$coefs` contiene los coeficientes constantes del error estándar.

Sabemos que los intervalo de confianza puntual para el parámetro $\hat{\beta}$ está dado por

$$\hat{\beta} \pm 2 \times \sqrt{\text{Var}[\hat{\beta}]} \sqrt{(r+1) \mathbf{F}_{r+1, n-r-1}(\alpha)}$$

con

$$\text{Var}[\hat{\beta}] = \sigma_e^2 (\mathbf{Z}'\mathbf{Z} + \lambda \mathbf{R}_0)^{-1} \mathbf{Z}'\mathbf{Z} (\mathbf{Z}'\mathbf{Z} + \lambda \mathbf{R}_0)^{-1}.$$

de donde la matriz de varianzas y covarianzas de \mathbf{y} usando el modelo $\mathbf{y} = \mathbf{Z}(t)\beta(t) + \boldsymbol{\varepsilon}$ está dada por matriz de varianzas y covarianzas \sum_e del vector de residuales $\boldsymbol{\varepsilon}$, tal que

$$\sum_e = \sigma_e^2 = (n-1)^{-1} \mathbf{E}'\mathbf{E}$$

teniendo esto en cuenta será más fácil entender lo que se está calculando en el siguiente código.

#:::::::::::::::::::::::::: Gráficas del coeficiente funcional con su intervalo de confianza puntual

```

betafdParj <- betaestlist[[2]]

betafd <- betafdParj$fd

betastderrfd <- betastderrlist[[2]]

betavec <- eval.fd(tiempo, betafd)

betastderrvec <- eval.fd(tiempo, betastderrfd)

betaplotmat <- cbind(betavec, betavec+2*betastderrvec,
betavec-2*betastderrvec)

matplot(tiempo, betaplotmat, type="l", lty=c(1,4,4),
xlab="meses", ylab="Temperatura Reg. Coeff.")

```

```
lines(intervalo,c(0,0),lty=2)
```

Código base en Splus, correspondiente a la aplicación para *Ecuaciones diferenciales ordinarias en el análisis de datos funcionales*, para el conjunto de datos que representa la pérdida de peso en ajos Mexicanos

El conjunto de datos para este análisis funcional, fueron proporcionados por

```
# *****
# Análisis con ADP de los ajos
# *****
#::::::::::::::::::::: Datos de entrada
ajo <-as.matrix(ajod, 19, 18, byrow=T)
dias <-c(10,20,30,40,50,60,70,80,90,100,
110,120,130,140,150,160,170,180,190)
rango <-c(10,190)
#::::::::::::::::::::: Gráfica de las 18 curvas de los valores discretos
par(mfrow=c(1,1), pty="m")
matplot(dias ,ajo, type="b",cex=1.0, xlab="días",ylab="datos",
main="18 curvas de los valores discretos")
#::::::::::::::::::::: Representación funcional de las 18 curvas
knoss <-dias
ordenp <-4
nbasis <-20+ordenp-2
```

```

ajobases <- create.bspline.basis(rango,nbasis,ordenp,knoss)

#:::::::::::::::::::::::::: Suavizamiento de las 18 funciones

Lfdobj <- int2Lfd(2)

lambda <- 1e-11

ajosfdPar <- fdPar(ajobases, Lfdobj, lambda)

ajosfd <- smooth.basis(dias,ajo, ajosfdPar)$fd

names(indiceemfd$fdnames) <- c("Días","Datos")

#:::::::::::::::::::::::::: Grafica las 18 funciones de pérdida de peso ya suavizadas

tipo <-c(1,1,1,2,2,2,3,3,3,4,4,4,5,5,5,6,6,6)

color <-c(1,1,1,2,2,2,3,3,3,4,4,4,5,5,5,6,6,6)

par(mfrow=c(1,1), pty="m")

plot(ajosfd,xlab="tiempo reportado en días",ylab="pérdida de peso en gramos",
main="Gráficas de los 6 tratamientos",lty=tipo,col=color)

tiplin <-c(1,4,1,6,13,14,37,1,4)

colr <-c(2,8,4,5,6,4,3,16,5)

axis(1,at=c(25,75,125,175))

key ( x=0, y=25,lines = list(lty = c(1,2,3,4,5,6), col=c(1,2,3,4,5,6)),cex=0.8,
text=list( c("0° C", "20° C", "30° C", "5° C", "0° C / 70RH","no control"),
transparent=T))

#:::::::::::::::::::::::::: Gráfica de las tres réplicas funciones del tratamiento 6

vajos<- eval.fd(dias, ajosfd)

ajostrat6 <- vajos[,16:18]

```



```

ajostrat6fd <- smooth.basis(dias, ajostrat6, ajosfdPar)$fd
tipo <- c(6,6,6)
color <- c(6,6,6)
plot(ajostrat6fd, xlab="tiempo reportado en dias", ylab="pérdida de peso",lty=tipo,
col=color)

```

```

mtext(Rélicas funcionales del tratamiento 6", line=-4, outer=T, cex=1.5)

```

```

#*****

```

```

# Análisis principal diferencial

```

```

# *****

```

La ecuación diferencial que se propone es de tercer orden (*ed3*).

```

#:::::::::::::::::::::::::: Expansión funcional para los coeficientes de la ed3

```

Los siguientes 5 renglones de código crean los coeficientes variables y constantes para la *ed3*

```

pdabasis <-create.bspline.basis(rango, nbasis,ordenp,knoss)

```

```

betafdParbb <-fdPar(pdabasis)

```

```

pdabasisC <-create.constant.basis(rango)

```

```

pdabasisfd <-fd(0,pdabasisC)

```

```

betafdParBc <-fdPar(pdabasisfd)

```

Para el tratamiento 6 el primer coeficientes es variable y el resto es constante.

```

bwtlist <- vector("list", 3)

```

```

bwtlist[[1]] <- betafdParbb

```

```

bwtlist[[2]] <- betafdParBc

```

```

bwtlist[[3]] <- betaafdParBc

xfdlist <- list(ajostrat6fd)

#.....: Parámetros funcionales estimados para el tratamiento seis

pdaList <- pda.fd(xfdlist, bwtlist)

bwtestlist <- pdaList$bwtlist

par(mfrow=c(2,2),pty="m")

for (j in 1:3)
{
bfdParj <- bwtestlist[[j]]

plot(bfdParj$fd, main="Parámetros funcionales estimados")
}

#.....: Cálculo de la función forzada f(t)

Una vez que se tiene los coeficientes  $\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2$ , definimos el operador

```

$$L = \hat{\beta}_0 I + \hat{\beta}_1 D + \hat{\beta}_2 D^2 + D^3$$

el cual aplicamos a cada una de las réplicas funcionales $y_i(t)$ del tratamiento seis, obteniendo así tres funciones forzadas o "*residual funcional*"

$$L y_i(t) = f_i(t).$$

estos cálculos son efectuados por los siguientes dos renglones de código.

```

Lfdest <- Lfd(3, bwtestlist)

forceLfd <- eval.fd(dias, ajostrat6fd, Lfdest)

#.....: Resolver la ecuación

```

Ahora que ya se ha definido el operador diferencial, procedemos a hacer una estimación de las funciones $y_i(t)$ es decir a encontrar las tres soluciones ξ_r linealmente independientes de este operador diferencial. Para que estas soluciones sean linealmente independientes, debemos definir el operador de restricciones $\mathbf{B}_0 y(t)$ que será una matriz para garantizar que nuestras soluciones sean linealmente independientes. En este caso

$$\mathbf{B}_0 y(t) = \begin{bmatrix} y_i(t) & 0 & 0 \\ 0 & Dy_i(t) & 0 \\ 0 & 0 & D^2 y_i(t) \end{bmatrix}.$$

lo anterior es calculado con el siguiente código:

```

fdajostat6 <-eval.fd(dias,ajostat6fd)

fdajostat6mean <-apply(fdajostat6,1,mean)

D1ajostat6 <-eval.fd(dias,ajostat6fd,1)

D2ajostat6 <-eval.fd(dias,ajostat6fd,2)

D1ajostat6mean <-apply(D1ajostat6,1,mean)

D2ajostat6mean <-apply(D2ajostat6,1,mean)

ystart <-matrix(0,3,3)

ystart[1,1] <-fdajostat6mean[[1]]

ystart[2,2] <-D1ajostat6mean[[1]]

ystart[3,3] <-D2ajostat6mean[[1]]

#::::::::::::::::::::: Código para encontrar las tres soluciones  $\xi_r$  del espacio  $\ker L$ .

EPSval <-1e-3

MAXSTP<-1000

result <-odesolv(bwtestlist,ystart,EPS=EPSval,MAXSTP)

```

```

tp <- result[[1]]
yp <- result[[2]]

#:::::::::::::::::::::::::: Código para graficar las tres soluciones del operador diferencial

par(mfrow=c(1,1),pty="m")

pltrng <-c(min(yp[1,,]),max(yp[1,,]))

matplot(tp,t(yp[1,,]), type="l", lty=1, ylim=pltrng,xlab="tiempo",
ylab="valores")

abline(h=0,lty=2)

mtext("Soluciones del tratamiento 6",line=-0.5,outer=T,cex=1.5)

#:::::::::::::::::::::::::: Soluciones estimadas del operador  $L = \hat{\beta}_0 I + \hat{\beta} D + \hat{\beta}_2 D^2 + D^3$ 

vajostrat6 <- eval.fd(dias,ajostrat6fd)

umats <-matrix(0,length(dias),3)

umats[,1] <-approx(tp,t(yp[1,1,]),dias)$y
umats[,2] <-approx(tp,t(yp[1,2,]),dias)$y
umats[,3] <-approx(tp,t(yp[1,3,]),dias)$y

#:::::::::::::::::::::::::: Gráfica de las funciones forzadas  $f_i(t)$  y de las función  $\hat{y}(t)$ 
para el tratamiento 6

for(i in 1:3)
{
par(mfrow=c(1,2),pty="s")

ajo1 <-vajostrat6[,i]

```

```

plot(dias,forceLfd[,i],type="l",ask=T,ylim=c(-0.01,0.01),
xlab="dias",ylab=,main=paste("función forzada",i))

abline(h=0,lty=2)

#::::::::::::::::::::: Combinación lineal de las soluciones  $\xi_r$ 

xhats <- vajostrat6[,i]-lsfit(umats,vajostrat6[,i],int=F)$residual

matplot(dias,cbind(ajo1,xhats,vajostrat6[,i]),type="pll",
pch=5,col=c(1,5,4),lty=c(1,3),xlab="dias",ylab="datos",
main="Modelo Estimado",ylim=c(-2,25))

mtext("Tratamiento 6",line=-4,outer=T,cex=2)

}

```

Código base en Splus, correspondiente a la aplicación para *Ecuaciones diferenciales ordinarias en el análisis de datos funcionales*, para el conjunto de datos del Indicador Global de la Actividad Económica Mexicana (1993–2005)

El conjunto de datos para este análisis funcional, se obtuvieron de la página electrónica:

<http://dgcnesyp.inegi.gob.mx/cgi-win/bdieintsi.exe/Consultar>

```

# *****

# Indicador Global de la Actividad Económica Mexicana

# *****

#::::::::::::::::::::: Datos de entrada

indiceem <- as.matrix(indice,156,1,byrow=T)

```

```

datos <- as.matrix(indice,156,1,byrow=T)
tiempo<-seq(1993+1/12, 2006, 1/12)
intervalo <- c(min(tiempo),max(tiempo))
#::::::::::::::::::::: Gráfica de los valores discretos
par(mfrow=c(1,1), pty="m")
plot(tiempo ,indiceem, type="b",cex=1.0; xlab=".^ños",ylab="datos",
main="Gráfica de los valores discretos")
#::::::::::::::::::::: Representación funcional de los datos del Indicador Global
de la Actividad Económica Mexicana
knots <- tiempo
orden <- 8
nbasis <- 156 + orden-2
iembasis <- create.bspline.basis(intervalo,nbasis,orden,knots)
#::::::::::::::::::::: Suavizamiento de los datos del Indicador Global de la Ac-
tividad Económica Mexicana
Lfdobj <- int2Lfd(3)
lambda <- 1e-11
iemfdPar <-fdPar(iembasis,Lfdobj,lambda)
indiceemfd <-smooth.basis(tiempo,indiceem,iemfdPar)$fd
#::::::::::::::::::::: Gráfica de la aceleración de los la función del IGAEM
par(mfrow=c(1,1),mar=c(5,5,4,2)+cexval+2, pty="m")
plot(indiceemfd,Lfd=2,xlab=".^ños",

```

```
main=.^Aceleración de la función del IGAEM")
```

```
#::::::::::::::::::::::::::::: Análisis diferencial principal
```

```
# *****
```

```
#::::::::::::::::::::::::::::: Ecuación diferencial de tercer orden
```

La ecuación diferencial que se propone es de tercer orden (ed3).

```
#::::::::::::::::::::::::::::: Expansión funcional para los coeficientes de la ed3
```

Los siguientes 5 renglones de código crean los coeficientes variables y constantes para la ed3

```
pdabasisB <- create.bspline.basis(intervalo,156)
```

```
betafdParbb <- fdPar(pdabasisB)
```

```
pdabasisC <- create.constant.basis(intervalo)
```

```
pdabasisfd <- fd( 1, pdabasisC)
```

```
betafdParbc <- fdPar(pdabasisfd)
```

#::::::::::::::::::::::::::::: Se fijan tres coeficientes, dos variables funcionales y uno funcional constante

```
bwtlist = vector("list", 3)
```

```
bwtlist[[1]] <- betafdParbb
```

```
bwtlist[[2]] <- betafdParbb
```

```
bwtlist[[3]] <- betafdParbc
```

```
xfdlist <- list(indiceemfd)
```

```
#::::::::::::::::::::::::::::: Gráficas de los parámetros funcionales  $\hat{\beta}_i$  de la ecuación
```

$$\beta_0 + \beta_1 Dy(t) + \beta_2 D^2y(t) + D^3y(t) = 0$$

```

pdaList <- pda.fd(xfdlist, bwtlist)

bwtestlist <- pdaList$bwtlist

par(mfrow=c(2,2),pty="m")

for (j in 1:3) {

bfdParj <- bwtestlist[[j]]

plot(bfdParj$fd,xlab=".Años",ylab="valores",

main=paste("Coeficiente estimado",j))

}

#.....Cálculo de la función forzada  $Ly_i(t) = f_i(t)$ 

Lfdest <- Lfd(3, bwtestlist)

forzada <- eval.fd(tiempo, indiceemfd, Lfdest)

D3indiceem <- eval.fd(tiempo, indiceemfd, 3)

```

Se dan las condiciones iniciales del operador $L = \hat{\beta}_0 I + \hat{\beta} D + \hat{\beta}_2 D^2 + D^3$ y se resuelve. Las condiciones

iniciales están dadas por el primer valor del *IGAEM*, la primer valor de la primera derivada del *IGAEM* y el primer valor de la segunda derivada del *IGAEM*.

```

datosindiceem <- eval.fd(tiempo,indiceemfd)

D1indiceemfd <- eval.fd(tiempo,indiceemfd,1)

D2indiceemfd <- eval.fd(tiempo,indiceemfd,2)

ystart <- matrix(0,3,3)

ystart[1,1] <- datosindiceem[1]

ystart[2,2] <- D1indiceemfd[1]

```



```

ystart[3,3] <- D2indiceemfd[1]

#:::::::::::::::::::::::::: Código para encontrar las tres soluciones  $\xi_r$  del espacio  $\ker L$ .

EPSval = 1e-4

result <- odesolv(bwtestlist, ystart, EPS=EPSval, MAXSTP=1e6)

xt <- result[[1]]

yp <- result[[2]]

#:::::::::::::::::::::::::: Gráfica de las tres soluciones del operador diferencial  $L =$ 
 $\hat{\beta}_0 I + \hat{\beta} D + \hat{\beta}_2 D^2 + D^3$ 

par(mfrow=c(1,1),pty="m")

pltrng <- c(min(yp[1,,]), max(yp[1,,]))

matplot(xt,t(yp[1,,]), type="l", lty=1,xlab=".ªños", ylab="IGAEM",
ylim=pltrng,main="Funciones estimadas")

abline(h=0, lty=2)

#:::::::::::::::::::::::::: Gráfica de la función forzada  $Ly(t)$ 

par(mfrow=c(1,1),pty="m")

plot(tiempo , forzada , type="l", ask=T, ylim=c(-250,250),
xlab=".ªños", ylab=, main=paste("función forzada"))

abline(h=0,lty=2)

#:::::::::::::::::::::::::: Soluciones estimadas del operador  $L = \hat{\beta}_0 I + \hat{\beta} D + \hat{\beta}_2 D^2 + D^3$ 

indiceem <- eval.fd(tiempo, indiceemfd)

umats <- matrix(0,length(tiempo),3)

umats[,1] <- approx(xt, t(yp[1,1,]), tiempo)$y

```

```
umats[,2] <- approx(xt, t(yp[1,2,]), tiempo)$y
```

```
umats[,3] <- approx(xt, t(yp[1,3,]), tiempo)$y
```

```
#::::::::::::::::::::: Gráfica de las funciones forzadas  $f(t)$  y de las función  $\hat{y}(t)$ 
```

generada por las soluciones ξ_r ,

```
xhats <- indiceem - lsfit(umats, indiceem, int=F)$residual
```

```
matplot(tiempo, cbind(datos, xhats), type="pl",pch=1,col=c(1,5),
```

```
lty=1, xlab="años", ylab="datos", main="Modelo diferencial estimado")
```

```
xhats <- indiceem - lsfit(umats, indiceem, int=F)$residual
```

```
matplot(tiempo, xhats, type="l",col=5, lty=1)
```

Apéndice B

Funciones de Splus

En cada uno de los programas ejecutados en *Splus* que se presentan en este trabajo y que fueron necesarios para generar resultados y conclusiones se utilizan varias funciones predeterminadas en *Splus* para ser usadas cuantas veces sea necesario. Para que no sonara repetitivo las explicaciones de estas y fuera de contexto (por el amplio contenido matemático que tiene cada una de ellas), a continuación se enumeran las más importantes y algunas observaciones de estas funciones.

1. `apply(X, MARGIN, FUN, ...)`

Descripción: aplica una función determinada a secciones de una matriz o a un vector.

2 `approx(x, y, xout=<<see below>>, method="linear", n=50, rule=1, f=0)`

Descripción: efectúa una interpolación lineal entre un conjunto de puntos cuyas coordenadas son x, y .

Esta función regresa una lista de valores $y(t)$, correspondientes a valores y los cuales son una interpolación del conjunto de datos de entrada.

3 `contour(x, y, z, nlevels=5, levels=pretty(z,nlevels), add=F, labex=1, save=F,`

`plotit=save, triangles=F)`

Descripción: grafica contornos

Esta función hace una gráfica de contornos y posiblemente regresa las coordenadas de las líneas de contorno. El argumento \mathbf{x} es un vector conteniendo las coordenadas $y_i(t)$ de la malla sobre la cual z es evaluada, \mathbf{y} es el vector que contiene las coordenadas y_j , \mathbf{z} es la matriz de de tamaño \mathbf{x} por \mathbf{y} es decir $\mathbf{z}[i,j]$ es evaluada en $(\mathbf{x}[i], \mathbf{y}[j])$.

4 `cbind(...)`

Descripción: construye una matriz de renglones.

Esta función regresa una matriz integrada por varios vectores o matrices.

5 `create.bspline.basis(rangeval, nbasis, norder=4, breaks)`

Descripción: crea una base del tipo *spline*

Cuando las funciones base ϕ_k son del tipo *spline* la función `create.bspline.basis` las genera del grado que se quiera. También genera los coeficientes c_k para definir una combinación lineal de estas funciones base. Las bases *spline* es un sistema que se usa cuando la función $y(t)$ es no periódica.

6 `create.constant.basis(rangeval=c(0, 1))`

Descripción: crea una base constante

Esta función define una función cuyo valor en todo un intervalo τ es el mismo.

7 `data2fd(y, argvals=seq(0, 1, len = n), basisobj, fdnames=defaultnames, argnames=c("time", "reps", "values"))`

Descripción: convierte un conjunto de datos discretos a un *Functional Data*

Esta función convierte un vector de valores y_j a valores de una función $y(t)$, es decir $y(t_j) = y_j$ para todo $t_j \in \tau$. Los valores $y(t_j)$ son obtenidos por medio el criterio de mínimos cuadrados.

8 `deriv.fd(fdobj, Lfdobj=1)`

Descripción: calcula la derivada de un Functional Data Object

La función `deriv.fd` obtiene la derivada de un objeto funcional. Esta función también es usada para aplicar un operador diferencial lineal de la forma $L = \beta_0 I + \dots + \beta_{m-1} D^{m-1} + D^m$ a un objeto funcional, el orden del operador de se indica en su argumento `Lfdobj`.

9 `eval.fd(evalarg, fdoobj, Lfdobj=0)`

Descripción: generalmente esta función es usada para evaluar a una función $y(t)$ en un intervalo τ para obtener los valores $y(t_j) = \hat{y}_j$. Esta función también puede derivar a la función $y(t)$ o aplicar un operador diferencial lineal a la función $y(t)$.

10 `eval.basis(evalarg, basisobj, Lfdobj=0)`

Descripción: genera los valores de funciones base o de sus derivadas.

Con esta función, se puede evaluar un conjunto de funciones base en un vector τ . Con esta función se puede aplicar un operador diferencial lineal a las funciones base y luego ser evaluadas.

11 `fd(coef=matrix(0, 2, 1), basisobj, fdnames=defaultnames)`

Descripción: define un *Functional Data Object*. Dado un conjunto de funciones base ϕ_k conocidas que son matemáticamente independientes una de otra y que tienen la propiedad de aproximar una función $y(t)$ al hacer una combinación lineal de estas funciones. La función `fd` encuentra esta combinación lineal de las funciones base ϕ_k , es decir $y(t) = \sum_{k=1}^K c_k \phi_k(t)$.

12 `fdPar(fdoobj=fd(), Lfdobj=int2Lfd(0), lambda, estimate=T)`

Descripción: define un *Functional Parameter Object*. Esta función también efectúa una combinación lineal de funciones base para hacer una estimación de la función que esta implícita en el conjunto de datos. La función `fdPar` se utiliza cuando se desea suavizar la función estimada $y(t)$.

13 `fRegress(yfdPar, xfdlist, betalist)`

Descripción: efectúa una regresión.

En el capítulo cuatro observamos que existen tres tipos de modelos, la función `fRegress` ayuda con el primer y tercer tipo de modelo que se mencionan. La función `fRegress` lleva a cabo un análisis de regresión funcional del tipo concurrente, y estima los parámetros funcionales de regresión correspondiente a cada variable independiente. Básicamente en el argumento `yfdPar` se almacena la variable dependiente, `xfdlist` la variable independiente y en `betalist` se crea el tipo de coeficiente funcional para la regresión.

14 `fRegress.stderr(fRegressList, y2cMap, SigmaE)`

Descripción: calcula los errores estándar e_{β_i} de los parámetros funcionales obtenidos por la función `fRegress`.

15 `int2Lfd(m)`

Descripción: convierte un entero en un operador diferencial lineal.

Lo que hace esta función es especificar el orden del operador diferencial o el orden de la derivada a ser tomada en cuenta para el análisis funcional.

16 `list(...)`

Descripción: crea una lista de objetos

Esta función regresa un objeto del modo lista, con tantos componentes como argumentos existan.

17 `lsfit(x, y, wt=<<see below>>, intercept=T, tolerance=1.e-07,`

`yname=NULL)`

Descripción: ajuste lineal por mínimos cuadrados

Esta función hace una regresión multivariada por mínimos cuadrados. Una lista de los coeficientes estimados y residuales, así como también una descomposición QR de la matriz \mathbf{x} es regresada. El argumento \mathbf{x} es el vector o matriz de las variables explicativas y el argumento \mathbf{y} es la matriz o vector de las variables dependientes.

18 `Lfd(nderiv=0, bwtlist=vector("list", 0))`

Descripción: define un operador diferencial lineal de orden m .

Esta función sirve para penalizar la rugosidad de alguna función $y(t)$.

19 `linmod(xfdobj, yfdobj, wtvec=rep(1,nrep), xLfdobj=int2Lfd(2),`

`yLfdobj=int2Lfd(2),xlambd=0, ylambda=0)`

Descripción: efectúa el ajuste de un modelo funcional completo.

Esta función hace una regresión funcional lineal, cuando los argumentos `xf-dobj`, `yfdobj` son funciones.

20 `max(...)`

Descripción: encuentra el valor máximo de un conjunto de escalares
Esta función regresa un número que es el máximo de un conjunto de datos.

21 `min(...)`

Descripción: encuentra el valor mínimo de un conjunto de escalares
Esta función regresa un número que es el mínimo de un conjunto de datos.

22 `mean(x, trim = 0, na.rm = F)`

Descripción: calcula la media aritmética de un vector o matriz.
Esta función regresa un valor, el cual es la media de un conjunto de datos.
Si tenemos un conjunto de datos y_1, \dots, y_n , la media esta dada por

$$\bar{y} = n^{-1} \sum_{j=1}^n y_j$$

23 `mean.fd(fdobj)`

Descripción: promedio de un dato funcional
Evalúa la media de un conjunto de funciones. Dado el conjunto de funciones $y_i(t)$ $i = 1, \dots, N$ entonces la media funcional, está definida por

$$\bar{y}(t) = N^{-1} \sum_{i=1}^N y_i(t)$$

24 `odesolv(bwtlist, ystart=diag(rep(1,norder)), h0=width/100, hmin=width*1e-10,`

`hmax=width*0.5, EPS=1e-4, MAXSTP=1000)`

Descripción: se obtienen las m soluciones de un sistema de ecuaciones diferenciales lineales de orden m , las soluciones son calculadas por el método de Runge-Kutta .

Esta función es usada cuando se hace un *ADP*. Con `odesolv` se estima la función $x(t)$ que satisface el sistema de ecuaciones diferenciales.

25 `pda.fd(xfdlist, bwtlist)`

Descripción: Análisis diferencial principal *ADP*

Cuando se tiene una función $y(t)$, que se ajusta a los datos, la función *pda.fd* estima una ecuación diferencial lineal homogénea cuya solución sea $y(t)$, es decir: $\frac{d^m y(t)}{dt^m} + \beta_{m-1} \frac{d^{m-1} y(t)}{dt^{m-1}} + \dots + \beta_1 \frac{dy(t)}{dt} + \beta_0 y(t) = 0$ de esta manera se logra obtener un modelo que brinda una mayor interpretación en la variabilidad de los datos. La función *pda.fd* trabaja de forma simple: como toda ecuación diferencial lineal de orden m puede ser reducida a un sistema de matrices primer orden con un simple cambio de variable.

Es decir, la ecuación diferencial de orden m homogénea

$$\beta_0 y(t) + \beta_1 D y(t) + \beta_2 D^2 y(t) + \beta_3 D^3 y(t) + D^4 y(t) + \dots + \beta_{m-1}(t) D^{m-1} y(t) + D^m y(t) = 0 \quad (\text{B.1})$$

puede expresarse como

$$\dot{y}(t) = \mathbf{A}(t)h(t)$$

definimos primero m nuevas variables

$$y_1(t), y_2(t), y_3(t), \dots, y_m(t)$$

tal que

$$y_1(t) = y(t), y_2(t) = \frac{dy(t)}{dt}, y_3(t) = \frac{d^2 y(t)}{dt^2}, y_4(t) = \frac{d^3 y(t)}{dt^3}, \dots, y_{m+1}(t) = \frac{d^m y(t)}{dt^m}$$

estas nuevas variables estas interrelacionadas por las ecuaciones

$$\dot{y}_1(t) = y_2(t), \dot{y}_2(t) = y_3(t), \dot{y}_3(t) = y_4(t), \dots, \dot{y}_m(t) = y_{m+1}(t) \quad (\text{B.2})$$

Entonces reescribiendo la ecuación (B.1) tenemos

$$\dot{y}_m(t) = \beta_0 y_1(t) + \beta_1 y_2(t) + \beta_2 y_3(t) + \beta_3 y_4(t) + \dots + \beta_{m-1} y_m(t) \quad (\text{B.3})$$

las ecuaciones (B.2) y (B.3) forman un sistema de ecuaciones diferenciales lineales de primer orden en $y_1(t), y_2(t), y_3(t), \dots, y_m(t)$, entonces este sistema es equivalente a la ecuación matricial única $\dot{y}(t) = \mathbf{A}(t)y(t)$, donde la el vector $\dot{y}(t)$ es

$$\dot{y}(t) \begin{bmatrix} \dot{y}_1(t) \\ \dot{y}_2(t) \\ \vdots \\ \dot{y}_{m-1}(t) \end{bmatrix} = \begin{bmatrix} y_2(t) \\ y_3(t) \\ \vdots \\ y_m(t) \end{bmatrix}$$

la matriz $A(t)$ es

$$A(t) = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ 0 & 0 & 0 & 0 & \dots & 1 \\ \beta_0 & \beta_1 & \beta_2 & \beta_3 & \dots & \beta_{m-1} \end{bmatrix}$$

y $y(t)$ es igual a

$$y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \cdot \\ \cdot \\ y_m(t) \end{bmatrix}$$

Como ya mencionamos en la expresión $\dot{y}(t) = A(t)y(t)$ conocemos $y(t)$ y a $\dot{y}(t)$ y lo que se desea es estimar a los coeficientes β_l donde $l = 0, \dots, m - 1$. La función *pda.fd* resuelve este sistema de ecuaciones diferenciales lineales obteniendo así una estimación de los coeficientes $\beta_0, \beta_1, \beta_2, \dots, \beta_{m-1}$.

26 `plot.fd(fdobj, Lfdobj=0, xlab=xlabel, ylab=ylabel, xlim=rangex,`

`ylim=rangey)`

Descripción: grafica datos funciones.

27 `plotfit.fd(y, argvals, fdobj, rng = rangeval, index = 1:nrep, nfine = 101,`

`residual = F, sortwrd = F, titles=NULL, ...)`

Descripción: grafica la función $y_i(t)$ vs los datos y_{i1}, \dots, y_{in} donde $i = 1, \dots, N$, donde $y_i(t) = \sum_{k=1}^K c_k \phi_k(t)$. La función `plotfit.fd` sirve para evaluar el ajuste de la función $y_i(t)$ para los datos y_{i1}, \dots, y_{in}

28 `rbind(...,)`

Descripción: construye una matriz de columnas.

Esta función regresa una matriz integrada por varios vectores o matrices.

29 `round(x, digits=0)`

Descripción: redondea funciones

Esta función regresa un objeto del mismo tipo que `x`, pero con los números redondeados. El redondeo se especifica con el argumento `digits`.

30 `solve(a, b, ...)`

Descripción: resuelve ecuaciones lineales e invierte matrices

Esta función calcula la matriz inversa de una matriz dada o resuelve sistemas de ecuaciones lineales si tenemos dos matrices de datos.

31 `sum(...)`

Descripción: calcula la suma de un conjunto de datos.

Esta función regresa un número que es la suma de todos los elementos de un vector.

32 `smooth.basis(argvals, y, fdParobj, wtvec=rep(1,n), dffactor=1,`

`fdnames=list(NULL, dimnames(y)[2], NULL))`

Descripción: suaviza la función $y(t)$ por medio del parámetro λ .

Esta función se auxilia de la función `fdPar`, donde se define el operador diferencial que va a ser utilizado para penalizar la rugosidad de los datos y el valor del parámetro λ .

33 `var(x, y, na.method="fail", unbiased=T, SumSquares=F)`

Descripción: calcula la varianza de un conjunto de datos

Esta función regresa la varianza de un vector, la matriz de varianza-covarianza de una matriz de datos, o las covarianzas entre matrices o vectores. El argumento `x` es una matriz o vector de escalares, si `x` es una matriz entonces las columnas representan las variables y los renglones las observaciones.

Bibliografía

- [1] B. Andrew and G. Hong-Ye (1996) *Applied Wavelet Analysis with S-Plus*, Springer.
- [2] Brown, L. D. y Low, M. G. (1996), Asymptotic equivalence of nonparametric regression and white noise. *Ann Statist.*, **24**, 2384 – 2398
- [3] B. Vidakovic (1995) *Statistical Modeling by Wavelets*. A Wiley-Interscience Publication.
- [4] David L. Donoho y Lain M. Johnstone, December 1995 ‘Adapting to Unknow Smoothness via Wavelet Shrikage’, *American Statistical Association Journal of the American Statistical Association*, Vol. 432, No. 432, *Theory and Methods*.
- [5] Douglas C. Montgomery (2006) *Diseño y análisis de experimentos*. Limusa Wiley, segunda edición.
- [6] Felix Abramovich et al. 2004 ‘Optimal Testing in a Fixed-Effects Functional Analysis of Variance Model’ *International Journal of Wavelets, Multiresolution and Information Processing*, Vol. 2, No. 4, pp 323 – 349. *World Scientific Publishing Company*.

- [7] J.O. Ramsay, B.W. Silverman (2005) *Functional Data Analysis*. Springer, segunda edición.
- [8] P. Barrera Sánchez, V. Hernández Mederos y C. Durán Sánchez (1996) *El ABC de los Spline*. Sociedad matemática Mexicana.
- [9] P.J Green and B.M. Silverman (1994) *Nonparametric Regression and Generalized Linear Models, A roughness penalty approach*. Chapman & Hall/Crc.
- [10] R.L. Shepley (1984) *Introducción a las Ecuaciones Diferenciales*. Interamericana, tercera edición.
- [11] Stephen H. Friedberg, Arnold J. Insel, Lawrence E. Spence (1982) *Algebra Lineal*. Publicaciones Cultural, S.A., primera edición.