

**ELECCIÓN DEL PARÁMETRO DE SUAVIZACIÓN ÓPTIMO EN EL
PROBLEMA DE LA SELECCIÓN DE VARIABLES EN REGRESIÓN
NO-PARAMÉTRICA A TRAVÉS DE UNA
SOLUCIÓN NUMÉRICA**

MARCO ANTONIO TRIANA LOZANO

**UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY
UNIVERSIDAD AUTÓNOMA DE OCCIDENTE
SANTIAGO DE CALI**

2004

**ELECCIÓN DEL PARÁMETRO DE SUAVIZACIÓN ÓPTIMO EN EL
PROBLEMA DE LA SELECCIÓN DE VARIABLES EN REGRESIÓN
NO-PARAMÉTRICA A TRAVÉS DE UNA
SOLUCIÓN NUMÉRICA**

MARCO ANTONIO TRIANA LOZANO

MAESTRIA EN CIENCIAS COMPUTACIONALES

Tesis

Director:

JAVIER OLAYA OCHOA

PhD en Estadística (Management Science)

**UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY
UNIVERSIDAD AUTÓNOMA DE OCCIDENTE
SANTIAGO CALI**

2004

Nota de aceptación:

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Cali, Marzo de 2004

A mi hija **VALENTINA TRIANA** quien fue la motivación principal para llevar a cabo este proyecto.

AGRADECIMIENTOS

Este trabajo fue posible gracias a la colaboración incondicional de Javier Olaya Ochoa, PhD en Estadística (Management Science), profesor titular de la Universidad del Valle. El doctor Olaya dirigió el proyecto de investigación prestando asesoría técnica adecuada para el desarrollo del proyecto. Este trabajo también fue posible gracias a la colaboración de la Universidad Autónoma de Occidente, entidad que brindo la financiación total del proyecto.

Gracias a mi esposa **VIVIANA RESTREPO** y mi madre **DORIS LOZANO** que por su ayuda y respaldo logré cumplir una meta más en mi vida personal y profesional.

CONTENIDO

	Pág.
INTRODUCCIÓN	16
1. REGRESIÓN NO-PARAMÉTRICA	20
1.1 ESTIMADORES	22
1.2 CONSISTENCIA	26
1.3 ALGUNOS CRITERIOS PAA DETERMINAR UN BUEN ESTIMADOR	27
1.4 SECCIÓN DEL PARÁMETRO DE SUAVIZACIÓN	29
2. SELECCIÓN DEL PARÁMETRO DE SUAVIZACIÓN EN EL PROBLEMA DE SELECCIÓN DE VARIABLES EN REGRESIÓN NO-PARAMÉTRICA	31
2.1 EL PROBLEMA DE LA SELECCIÓN DE VARIABLE	31
2.1.1 Un método para seleccionar variables en regresión no-paramétrica	33
2.2 ALGUNOS MÉTODOS PARA SELECCIONAR EL PARÁMETRO DE SUAVIZACIÓN E REGRESIÓN NO-PARAMÉTRICA	47
2.3 SELECCIÓN DEL PARÁMETRO DE SUAVIZACIÓN EN LA COMPARACIÓN DE DOS CURVAS DE REGRESIÓN	53
2.3.1 Selección del valor óptimo para el parámetro de suavización que maximiza la potencia de la prueba	53
3. RESULTADOS DE ALGUNAS SIMULACIONES	60
3.1 ESTIMACIÓN DE LA FUNCIÓN DE REGRESIÓN $\mu(x)$	60
3.1.1 Algunos aspectos computacionales de la simulación	61

3.1.2 Resultados de la simulación	62
3.2 SELECCIÓN DE UN VALOR ÓPTIMO DE PARÁMETRO SUAVIZACIÓN PARA ESTIMAR LA FUNCIÓN DE REGRESIÓN $\mu(x)$	64
3.2.1 Algunos aspectos computacionales de la simulación	65
3.2.2 Resultados de la simulación	65
3.3 SELECCIÓN DE UN VALOR ÓPTIMO DEL PARÁMETRO DE SUAVIZACIÓN PARA MAXIMIZAR LA POTENCIA DE UNA PRUEBA	69
3.3.1 algunos aspectos computacionales de la simulación	69
3.3.2 Resultados dela simulación	70
3.4 SELECCIÓN DE UN VALOR ÓPTIMO DEL PARÁMETRO DE SUAVIZACIÓN EN EL PROBLEMA DE SELECCIÓN DE VARIABLES EN REGRESIÓN NO-PARAMÉTRICA	79
3.4.1 Caso donde se tiene en el modelo dos variables explicativas independientes X_1 y X_2	80
3.4.2 Caso donde se tienen en el modelo tres variables explicativas independientes X_1, X_2 y X_3	87
4. DOCUMENTACIÓN DE PROGRAMAS EN C++ Y EN LENGUAJE S	96
4.1 PROGRAMA NEWK1.XXC (Procedimiento en lenguaje S)	96
4.2 PROGRAMA NEWK (Programa en C++)	97
4.3 PROGRAMA NEWK1000.SSC (Procedimiento en lenguaje S)	98
4.4 PROGRAMA NEWK100(Programa en C++)	99
4.5 PROGRAMA NEWK1001.SSC (Procedimiento en lenguaje S)	100
4.6 PROGRAMA NEWK1001 (procedimiento en C++)	101

4.7 PROGRAMA NEWK1002.SSC (procedimiento en Lenguaje S)	102
4.8 PROGRAMA NEWK1002 (Programa en C++)	104
4.9 PROGRAMA NEWK1003.SSC (Procedimiento en lenguaje S)	105
4.10 PROGRAMA NEWK1003 (programa en C++)	107
4.11 PROGRAMA NEWK1004 (Procedimiento en Lenguaje S)	108
4.12 PROGRAMA NEWK104_RANDOM.SSC (Procedimiento en Lenguaje S)	111
4.13 PROGRAMA NEW1005.SSC (Procedimiento en lenguaje S)	114
4.14 PROGRAMA NEWK1005 (programa en C++)	118
4.15 PROGRAMA NEWK1005_RANDOM.SSC (procedimiento en lenguaje S)	121
5. CONCLUSIONES	132
BIBLIOGRAFÍA	135

LISTA DE TABLAS

	Pág.
Tabla 1. (1000 repeticiones) Con errores normales tipo 1. Tamaño de muestra igual 100. <i>Criterio de validación cruzada Generalizada</i> según <i>Gasser y Müller</i> (1988) . . .	67
Tabla 2. (1000 repeticiones) Con errores normales tipo 2. Tamaño de muestra igual 100. <i>Criterio de validación cruzada Generalizada</i> según <i>Gasser y Müller</i> (1988) . . .	68
Tabla 3. Potencia empírica (1000 repeticiones) Con errores normales tipo 1. Se generaron dos muestras n_1 y n_2 de tamaño 50. Diseño de puntos diferente para cada muestra. Se utilizó una sola función $f(x)$ para generar los valores observados de Y . . .	72
Tabla 4. Potencia empírica (1000 repeticiones) Con errores normales tipo 1. Se generaron dos muestras n_1 y n_2 de tamaño 50. Diseño de puntos diferente para cada muestra. Se utilizó una sola función $f(x)$ para generar los valores observados de Y . . .	73
Tabla 5. Potencia empírica (1000 repeticiones) Con errores normales tipo 1. Se generaron dos muestras n_1 y n_2 de tamaño 50. Diseño de puntos igual para cada muestra. Se utilizaron dos funciones $f(x)$ y $g(x)$ para generar los valores observados de Y	76
Tabla 6. Potencia empírica (1000 repeticiones) Con errores normales tipo 2. Se generaron dos muestras n_1 y n_2 de tamaño 50. Diseño de puntos igual para cada muestra. Se utilizaron dos funciones $f(x)$ y $g(x)$ para generar los valores observados de Y	76

Tabla 7. Potencia empírica (1000 repeticiones) Con errores normales tipo 1. Se generaron dos muestras n_1 y n_2 de tamaño 50. Diseño de puntos diferente para cada muestra. Se utilizaron dos funciones $f(x)$ y $g(x)$ para generar los valores observados de Y	78
Tabla 8. Potencia empírica (1000 repeticiones) Con errores normales tipo 2. Se generaron dos muestras n_1 y n_2 de tamaño 50. Diseño de puntos diferente para cada muestra. Se utilizaron dos funciones $f(x)$ y $g(x)$ para generar los valores observados de Y	79
Tabla 9. Potencia empírica (1000 repeticiones) Con errores normales tipo 1. Se generaron dos submuestras n_1 y n_2 de tamaño 50. Diseño de puntos aleatorios de distribución uniforme con valores en el intervalo $[0,1]$. Se utilizó una función $f(x)$ para generar los valores observados de Y	83
Tabla 10. Potencia empírica (1000 repeticiones) Con errores normales tipo 2. Se generaron dos submuestras n_1 y n_2 de tamaño 50. Diseño de puntos aleatorios de distribución uniforme con valores en el intervalo $[0,1]$. Se utilizó una función $f(x)$ para generar los valores observados de Y	84
Tabla 11. Potencia empírica (1000 repeticiones) Con errores normales tipo 1. Se generaron dos submuestras n_1 y n_2 de tamaño 50. Diseño de puntos aleatorios no de distribución uniforme con valores en el intervalo $[0,1]$. Se utilizó una función $f(x)$ para generar los valores observados de Y	86
Tabla 12. Potencia empírica (1000 repeticiones) Con errores normales tipo 2. Se generaron dos submuestras n_1 y n_2 de tamaño 50. Diseño de puntos aleatorios no de distribución uniforme con valores en el intervalo $[0,1]$. Se utilizó una función $f(x)$ para generar los valores observados de Y	87

Tabla 13. Potencia empírica (1000 repeticiones) Con errores normales tipo 1. Se generaron dos submuestras n_1 y n_2 de tamaño 108. Diseño de puntos aleatorios de distribución uniforme con valores en el intervalo [0,1]. Se utilizó una función $f(x)$ para generar los valores observados de Y.	91
Tabla 14. Potencia empírica (1000 repeticiones) Con errores normales tipo 2. Se generaron dos submuestras n_1 y n_2 de tamaño 108. Diseño de puntos aleatorios de distribución uniforme con valores en el intervalo [0,1]. Se utilizó una función $f(x)$ para generar los valores observados de Y.	92
Tabla 15. Potencia empírica (1000 repeticiones) Con errores normales tipo 1. Se generaron dos submuestras n_1 y n_2 de tamaño 108. Diseño de puntos no aleatorios de distribución uniforme con valores en el intervalo [0,1]. Se utilizó una función $f(x)$ para generar los valores observados de Y.	94
Tabla 16. Potencia empírica (1000 repeticiones) Con errores normales tipo 2. Se generaron dos submuestras n_1 y n_2 de tamaño 108. Diseño de puntos no aleatorios de distribución uniforme con valores en el intervalo [0,1]. Se utilizó una función $f(x)$ para generar los valores observados de Y.	95

LISTA DE FIGURAS

Pág.

- Figura 1.** Un ejemplo simulado para una muestra de tamaño 100. Un ancho de banda de 0,01 con errores tipo 1. Los valores estimados de Y (estimación de $\mu(x)$) aparecen en una línea continua. Los valores de la función $\mu(x)$ aparecen en círculos consecutivos. Los valores observados de Y aparecen en círculos dispersos. 63
- Figura 2.** Un ejemplo simulado para una muestra de tamaño 100. Un ancho de banda de 0,03 con errores tipo 1. Los valores estimados de Y (estimación de $\mu(x)$) aparecen en una línea continua. Los valores de la función $\mu(x)$ aparecen en círculos consecutivos. Los valores observados de Y aparecen en círculos dispersos. 63
- Figura 3.** Un ejemplo simulado para una muestra de tamaño 100. Un ancho de banda de 0,05 con errores tipo 1. Los valores estimados de Y (estimación de $\mu(x)$) aparecen en una línea continua. Los valores de la función $\mu(x)$ aparecen en círculos consecutivos. Los valores observados de Y aparecen en círculos dispersos. 64
- Figura 4.** Un ejemplo simulado para una muestra de tamaño 100. Con errores tipo 1. Los círculos representan los valores *del criterio de validación cruzada Generalizada* según Gasser y Müller (1988). 65
- Figura 5.** Un ejemplo simulado para una muestra de tamaño 100. Con errores tipo 2. Los círculos representan los valores *del criterio de validación cruzada Generalizada* según Gasser y Müller (1988). 66
- Figura 6.** Un ejemplo simulado para dos muestras n_1 y n_2 de tamaño 50. Diseño De puntos diferentes para cada muestra. Se utilizo una sola función $f(x)$ para generar los valores observados de Y. Errores tipo 1. Los círculos representan los

valores de la función $\hat{\Delta}(\lambda)$ 70

Figura 7. Un ejemplo simulado para dos muestras n_1 y n_2 de tamaño 50. Diseño De puntos diferentes para cada muestra. Se utilizo una sola función $f(x)$ para generar los valores observados de Y. Errores tipo 2. Los círculos representan los

valores de la función $\hat{\Delta}(\lambda)$ 71

Figura 8. Un ejemplo simulado para dos muestras n_1 y n_2 de tamaño 50. Diseño De puntos igual para cada muestra. Se utilizaron dos funciones $f(x)$ y $g(x)$ para generar los valores observados de Y. Errores tipo 1. Los círculos representan los

valores de la función $\hat{\Delta}(\lambda)$ 74

Figura 9. Un ejemplo simulado para dos muestras n_1 y n_2 de tamaño 50. Diseño De puntos igual para cada muestra. Se utilizaron dos funciones $f(x)$ y $g(x)$ para generar los valores observados de Y. Errores tipo 2. Los círculos representan los

valores de la función $\hat{\Delta}(\lambda)$ 75

Figura 10. Un ejemplo simulado para dos muestras n_1 y n_2 de tamaño 50. Diseño De puntos diferente para cada muestra. Se utilizaron dos funciones $f(x)$ y $g(x)$ para generar los valores observados de Y. Errores tipo 1. Los círculos

representan los valores de la función $\hat{\Delta}(\lambda)$ 77

Figura 11. Un ejemplo simulado para dos muestras n_1 y n_2 de tamaño 50. Diseño De puntos diferente para cada muestra. Se utilizaron dos funciones $f(x)$ y $g(x)$ para generar los valores observados de Y. Errores tipo 2. Los círculos representan los

valores de la función $\hat{\Delta}(\lambda)$ 78

Figura 12. Un ejemplo simulado para dos submuestras n_1 y n_2 de tamaño 50. Diseño

de puntos aleatorios de distribución uniforme con valores en el intervalo [0,1]. Se utilizó una función $f(x)$ para generar los valores observados de Y. Errores tipo 1. Los círculos representan los valores *de la función* $\hat{\Delta}(\lambda)$ 82

Figura 13. Un ejemplo simulado para dos submuestras n_1 y n_2 de tamaño 50. Diseño de puntos aleatorios de distribución uniforme con valores en el intervalo [0,1]. Se utilizó una función $f(x)$ para generar los valores observados de Y. Errores tipo 2. Los círculos representan los valores *de la función* $\hat{\Delta}(\lambda)$ 82

Figura 14. Un ejemplo simulado para dos submuestras n_1 y n_2 de tamaño 50. Diseño de puntos no aleatorios de distribución uniforme con valores en el intervalo [0,1]. Se utilizó una función $f(x)$ para generar los valores observados de Y. Errores tipo 1. Los círculos representan los valores *de la función* $\hat{\Delta}(\lambda)$ 85

Figura 15. Un ejemplo simulado para dos submuestras n_1 y n_2 de tamaño 50. Diseño de puntos no aleatorios de distribución uniforme con valores en el intervalo [0,1]. Se utilizó una función $f(x)$ para generar los valores observados de Y. Errores tipo 2. Los círculos representan los valores *de la función* $\hat{\Delta}(\lambda)$ 86

Figura 16. Un ejemplo simulado para dos submuestras n_1 y n_2 de tamaño 108. Diseño de puntos aleatorios de distribución uniforme con valores en el intervalo [0,1]. Se utilizó una función $f(x)$ para generar los valores observados de Y. Errores tipo 1. Los círculos representan los valores *de la función* $\hat{\Delta}(\lambda)$ 89

Figura 17. Un ejemplo simulado para dos submuestras n_1 y n_2 de tamaño 108. Diseño de puntos aleatorios de distribución uniforme con valores en el intervalo [0,1]. Se utilizó una función $f(x)$ para generar los valores observados de Y. Errores tipo 2.

Los círculos representan los valores <i>de la función</i> $\hat{\Delta}(\lambda)$	90
Figura 18. Un ejemplo simulado para dos submuestras n_1 y n_2 de tamaño 108. Diseño de puntos no aleatorios de distribución uniforme con valores en el intervalo [0,1]. Se utilizó una función $f(x)$ para generar los valores observados de Y.	
Errores tipo 1. Los círculos representan los valores <i>de la función</i> $\hat{\Delta}(\lambda)$	93
Figura 19. Un ejemplo simulado para dos submuestras n_1 y n_2 de tamaño 108. Diseño de puntos no aleatorios de distribución uniforme con valores en el intervalo [0,1]. Se utilizó una función $f(x)$ para generar los valores observados de Y.	
Errores tipo 2. Los círculos representan los valores <i>de la función</i> $\hat{\Delta}(\lambda)$	95
Figura 20. Diagramas de flujo programas NEWK -NEWK1.SSC.	123
Figura 21. Diagramas de flujo programas NEWK1000 -NEWK1000.SSC.	124
Figura 22. Diagramas de flujo programas NEWK1001 -NEWK1001.SSC.	125
Figura 23. Diagramas de flujo programas NEWK1002 -NEWK1002.SSC.	126
Figura 24. Diagramas de flujo programas NEWK1003 -NEWK1003.SSC.	127
Figura 25. Diagramas de flujo programas NEWK1002 -NEWK1004.SSC.	128
Figura 26. Diagramas de flujo programas NEWK1002-NEWK1004_RANDOM.SSC.	129
Figura 27. Diagramas de flujo programas NEWK1005 -NEWK1005.SSC.....	130
Figura 28. Diagramas de flujo programas NEWK1005-NEWK1005_RANDOM.SSC..	131

INTRODUCCIÓN

Cuando se habla de análisis de regresión, usualmente se piensa en la estimación de los coeficientes de regresión, estimación de la función de regresión, diseño de gráficos de residuales, etc. los cuales han sido obtenidos por ajuste de un modelo lineal a un conjunto de datos. Realmente esto es un aspecto en el análisis de regresión; más sin embargo, existe otro tópico importante generalmente referido como regresión no-paramétrica.

En un análisis de regresión, específicamente se supone que se toman n observaciones de una variable aleatoria continua Y en n valores determinados de una variable continua independiente X . Sea $(x_i, y_i), i = 1, \dots, n$ los valores de X y Y resultado de un plan de muestreo y se asume que x_i y y_i están relacionadas por medio del modelo

$$y_i = \mu(x_i) + \varepsilon_i, \quad i = 1, 2, \dots, n. \quad (1.1)$$

Donde los ε_i son variables aleatorias no observables con media 0 y varianza común δ^2 , no están correlacionados y los $\mu(x_i)$ son los valores de alguna función $\mu(x)$ desconocida en un diseño de puntos $x_1, x_2, x_3, \dots, x_n$. La función $\mu(x)$ en (1.1) es llamada *curva de regresión* o *función de regresión*. En este ambiente, el “análisis de regresión” se refiere a los métodos inferenciales estadísticos que permiten una aproximación de la función $\mu(x)$. La determinación de un método inferencial conveniente para el modelo (1.1) depende de los supuestos que es posible hacer sobre $\mu(x)$. En un modelo de regresión paramétrico se supone que la forma de $\mu(x)$ es conocida excepto por un número finito de parámetros desconocidos. Más específicamente, se asume que el vector de parámetros $(B_1, B_2, B_3, \dots, B_p)^T \in \mathbf{B}$ algún subconjunto de \mathbb{R}^p y una función conocida $\mu(x; \mathbf{B})$.

En un análisis paramétrico, la inferencia acerca de $\mu(\mathbf{x})$ es equivalente a la inferencia para **B**. Los modelos paramétricos pueden depender de los parámetros, bien sean lineales o no lineales. Las técnicas de un análisis de regresión para modelos paramétricos representan una aproximación para $\mu(\mathbf{x})$. Usando una apropiada metodología se puede estimar los parámetros y estimar así $\mu(\mathbf{x})$. El resultado es una curva de regresión ajustada.

El uso de *métodos paramétricos* puede ser verdaderamente inapropiado en situaciones donde existe poco conocimiento acerca de la función de regresión. En tales casos, los datos proporcionan muy poca información acerca de $\mu(\mathbf{x})$. Es conveniente usar técnicas inferenciales las cuales dependan mayormente en los datos. Por esta razón, las *técnicas de regresión no-paramétricas* son idealmente favorecidas en problemas inferenciales cuando el conocimiento acerca de $\mu(\mathbf{x})$ es limitado.

Las técnicas de regresión *paramétricas* y *no-paramétricas* representan dos diferentes aproximaciones al problema de *análisis de regresión*. Los métodos paramétricos requieren mucha especificación, información cuantitativa del investigador acerca de la forma de la función de regresión en lugar de restricciones sobre los datos que se puedan distinguir acerca de la función de regresión. Tales técnicas son más apropiadas cuando están disponibles resultados teóricos, experiencias pasadas u otras fuentes que proporcionan conocimiento detallado acerca del problema bajo estudio. En contraste, las técnicas de regresión no-paramétricas dependen del investigador en suministrar solo información cualitativa acerca de la función de regresión y permitir que los datos muestren por sí mismos la forma actual de la curva de regresión. En general, estos métodos de análisis de regresión no-paramétricos son óptimos para realizar inferencias en situaciones o problemas del mundo real donde se tenga poca información disponible acerca de la curva de regresión, o en casos de no tener un conocimiento acerca del problema, es decir, acerca del comportamiento de la función de regresión.

Las técnicas de regresión no-paramétricas superan inherentemente la dificultad de las técnicas paramétricas, entre las cuales podemos mencionar el conocimiento de la forma funcional de $\mu(\mathbf{x})$. Generalmente, las estimaciones no-paramétricas son menos eficientes que las estimaciones paramétricas cuando el modelamiento paramétrico es válido. De hecho, los procedimientos no-paramétricos pueden representar la etapa final en el análisis de datos o meramente un paso exploratorio en el proceso de modelamiento.

En la actualidad, encontramos algunas investigaciones recientes en el marco no-paramétrico que tratan algunos de los muchos problemas teóricos aun sin resolver. Estos desarrollos teóricos se refieren a la búsqueda de los parámetros de suavización más adecuados de acuerdo con algunos supuestos para realizar estimaciones confiables de la función de regresión; al problema de la selección de variables en el marco no-paramétrico, hasta ahora se han propuesto muy pocos métodos para la solución de este problema y en todos ellos un problema central, aún sin resolver, es la selección del parámetro de suavización (ancho de banda) óptimo; problemas de aplicación de la suavización no-paramétrica a modelos de regresión lineal y algunos problemas con respecto a la reducción de *dimensionalidad* en el análisis de regresión.

En este proyecto de investigación se pretende estudiar el problema de la escogencia del parámetro de suavización y encontrar por métodos numéricos un valor óptimo de este parámetro en el problema de la selección de variables en los modelos de regresión no-paramétricos.

Por medio de métodos de simulación se busca implementar una solución numérica para la escogencia del parámetro de suavización óptimo. Se diseñará un mecanismo computacional que permita la selección automática del parámetro de suavización óptimo en el problema de selección de variables en regresión no-paramétrica. Se usará como criterio de optimización la maximización de la potencia de una prueba estadística de comparación de curvas de regresión no-paramétricas.

En el *capítulo 2* se estudiará el problema de la selección de variables, se analizarán algunos aspectos teóricos del método para seleccionar el valor óptimo del parámetro de suavización λ en el problema de selección de variables en el marco *no-paramétrico*, también se resalta la importancia del parámetro de suavización en el problema de la selección de variables.

Algunos resultados de un estudio de simulación son discutidos en el *capítulo 3*. Inicialmente, se realiza la estimación de la función de regresión $\mu(\mathbf{x})$, se evalúan algunos criterios para seleccionar el parámetro de suavización el cual proporciona una mejor estimación de la función $\mu(\mathbf{x})$, luego se evalúa un método para encontrar el valor óptimo del parámetro de suavización λ en la comparación de dos curvas de regresión; por último se evalúa un método para encontrar el valor óptimo del parámetro de suavización en el problema de selección de variables maximizando la potencia de una prueba en regresión *no-paramétrica*. En el *capítulo 4* se describen los procedimientos de cada uno de los programas desarrollados en este proyecto que permiten realizar la evaluación de algunos métodos para la selección automática del parámetro de suavización óptimo en el problema de selección de variables en regresión no-paramétrica.

1. REGRESIÓN NO-PARAMÉTRICA

Las técnicas de análisis de regresión para modelos no-paramétricos representan una estimación de la curva de regresión. Estos métodos dan estimaciones de la función de regresión, los cuales permiten gran flexibilidad en la posible forma de la curva de regresión, y en particular, no hacen suposiciones acerca de la forma paramétrica. La expresión *Regresión No-Paramétrica* se refiere a una forma flexible de la función de una curva de regresión.

Para empezar, asumiremos que el modelo dado en (1.1),

$$y_i = \mu(x_i) + \varepsilon_i, \quad i = 1, 2, \dots, n.$$

representa una relación de una variable explicativa independiente \mathbf{X} y una variable de respuesta \mathbf{Y} , donde $\mu(\mathbf{x}) \in \mathbf{W}$, y \mathbf{W} es algún espacio de funciones de dimensión infinita. El objetivo de un *Análisis de Regresión No-Paramétrica* es realizar inferencias acerca de $\mu(\mathbf{x})$ con respecto a la forma de la función de regresión, es decir estimar y probar las características de la función de regresión.

En la especificación de un modelo de *regresión no-paramétrico* el experimentador elegirá un apropiado *espacio de funciones* el cual se cree que contiene una curva de regresión desconocida. Esta elección es usualmente motivada por ciertas propiedades de *suavización* (continuidad y diferenciabilidad) de la función de regresión que pueden ser asumidas. Los datos se utilizan para determinar un elemento del espacio de funciones el cual es representativo de la curva de regresión desconocida.

Según *Eubank (1999)*, existen muchas opciones naturales para determinar el espacio de funciones \mathbf{W} . Dependiendo de que tan *suave* es posible asumir $\mu(\mathbf{x})$, puede ser razonable tomar \mathbf{W} como el conjunto de todas las funciones $\mu(\mathbf{x})$ continuas en un intervalo cerrado $[\mathbf{a}, \mathbf{b}]$ o, más generalmente, el conjunto de todas las funciones con \mathbf{k} derivadas continuas en un intervalo cerrado $[\mathbf{a}, \mathbf{b}]$, para algún entero no negativo \mathbf{k} . Es importante pensar que la idea global es que \mathbf{W} supuestamente contiene la verdadera curva de regresión.

Las técnicas de regresión no-paramétricas dependen más de los datos para obtener información acerca de la función de regresión que su contraparte paramétrica. El resultado de un análisis de regresión no-paramétrico es una curva ajustada a un conjunto de datos. Esta curva de regresión es estimada sin asumir una forma paramétrica para $\mu(x)$. Al procedimiento para encontrar una aproximación de la curva de regresión en un análisis de regresión no-paramétrico se le llama *Suavización*.

El análisis de regresión no-paramétrico es indudablemente durante los últimos años una herramienta muy útil para realizar estimaciones de una curva de regresión. Estas curvas de regresión no-paramétricas se pueden aproximar por medio de algunos métodos de estimación, entre los cuales tenemos los *métodos de suavización*.

La teoría y los métodos de suavización han sido desarrollados en gran parte durante los últimos años. El interés se ha venido incrementando en esta última década principalmente en la estimación de curvas de regresión, también el desarrollo del hardware que permita el procesamiento computarizado para encontrar estimaciones no-paramétricas.

Las técnicas de suavización tienen una gran tradición. A finales del siglo XIX las aproximaciones no-paramétricas se usaban como la mejor herramienta para realizar análisis empíricos. Desde esa época, las aproximaciones no-paramétricas por medio de métodos de suavización no se le daba la importancia que realmente se merece. Los desarrollos matemáticos de las teorías estadísticas en la primera mitad del siglo XX fueron implementados para realizar aproximaciones paramétricas por su simplicidad computacional.

Estos métodos de suavización pueden ser usados en Economía, Ingeniería y Matemáticas.

En los actuales momentos existen varios métodos de suavización, entre ellos podemos citar: Estimador de series Ortogonales, Suavización a través de *Splines*, Estimadores mediante Polinomios Locales y los estimadores *Kernel*.

Al asumir que $\mu(\mathbf{x})$ es una función suave, se puede estimar $\mu(\mathbf{x})$ en un punto \mathbf{x}_i . Al disponer de observaciones de $\mu(\mathbf{x})$ (teniendo en cuenta el error) en algunos puntos \mathbf{x}_i que pertenecen al diseño de puntos $\{\mathbf{x}_i, i = 1, \dots, n\}$, estos valores \mathbf{x}_i pueden contener información acerca de los valores de la función $\mu(\mathbf{x})$ en \mathbf{x} porque la función es suave. Por esta razón será posible usar algún tipo de aproximación en las respuestas teniendo en cuenta los puntos \mathbf{x}_i para dar una estimación de la función $\mu(\mathbf{x})$.

La Suavización es una herramienta para resumir el comportamiento de las respuestas \mathbf{Y} como una función de la covariable \mathbf{X} . Los estimadores *no-paramétricos* más comunes se conocen como *estimadores lineales* de la función de regresión.

1.1 ESTIMADORES LINEALES

En este contexto apreciaremos varios tipos de estimadores de regresión no-paramétricos, primero consideremos el problema de estimación de la media del vector $\boldsymbol{\mu} = (\mu(x_1), \dots, \mu(x_n))^T$. El vector de respuestas $\mathbf{Y} = (y_1, \dots, y_n)^T$ da un posible estimador para el vector $\boldsymbol{\mu}$. Este estimador es insesgado donde $E(y_i) = \mu(\mathbf{x}_i)$, $i = 1, 2, \dots, n$, y tiene una matriz de *varianza-covarianza* $\sigma^2 \mathbf{I}$. Si las observaciones se repiten para algunos valores de \mathbf{X} , estos valores tendrían que agruparse y su promedio se usaría para obtener más precisión en los estimadores. En general, el vector de respuestas \mathbf{Y} no será un estimador confiable para $\mu(\mathbf{x}_i)$. Una razón para esto es que los valores de y_i presentan un error cuadrático grande. En particular, el error cuadrático medio para el vector de respuesta \mathbf{Y} como un estimador de $\mu(\mathbf{x}_i)$ es $n^{-1} \sum_{i=1}^n E(\mu(x_i) - y_i)^2 = \sigma^2$. Se observa que la característica de \mathbf{Y} como un buen estimador no mejora así se incrementa el tamaño de muestra.

Otro problema en la elección del vector de respuestas \mathbf{Y} como un buen estimador es que solo proporciona una estimación de $\mu(\mathbf{x}_i)$ y no resuelve el problema de estimación de la función $\mu(\mathbf{x})$ en algunos puntos que están por fuera del diseño de puntos. Como el objetivo en un *análisis de regresión no-paramétrico* es la estimación de la función

completa $\mu(\mathbf{x})$, no será posible obtener un estimador insesgado porque esto involucraría la estimación de más parámetros (en realidad un número infinito de parámetros).

Según *Eubank* (1999), si se asume que $\mu(\mathbf{x})$ es suave, entonces las observaciones en puntos \mathbf{x}_i cercanos a \mathbf{x} deben contener información acerca del valor de $\mu(\mathbf{x})$ en el punto \mathbf{x} . Así, será posible utilizar un *método de suavización*, tal como un promedio local de los datos cercanos a \mathbf{x} en la construcción de un estimador de $\mu(\mathbf{x})$. De otro lado, si usamos esta aproximación en un diseño de puntos $\{\mathbf{x}_i, i = 1, \dots, n\}$, debemos ser capaces de obtener un estimador de $\mu(\mathbf{x})$ con una varianza pequeña. Además, el sesgo que es introducido no debe de ser demasiado grande. Este tipo de *estimadores lineales* incluye promedios locales así como muchos otros estimadores de interés.

Un estimador lineal de $\mu(\mathbf{x})$ tiene la forma general:

$$\mu_\lambda(x) = \sum_{i=1}^n K(x, x_i; \lambda) y_i, \quad (1.2)$$

donde $K(x, x_i; \lambda), i = 1, \dots, n$, es una colección de pesos que no depende del diseño $\{\mathbf{x}_i, i = 1, \dots, n\}$, pero depende de un parámetro λ . Obsérvese que bajo este contexto $\mu_\lambda(x)$ depende de λ . La función (1.2) es llamada un estimador lineal porque para un λ dado, los estimadores resultan ser funciones lineales de las respuestas y_1, \dots, y_n . Si los pesos en (1.2) son todos *no-negativos* y $\sum_{i=1}^n K(x, x_i; \lambda) = 1$, entonces el estimador (1.2) resulta ser un promedio ponderado de pesos de las respuestas y_i . El parámetro λ en (1.2) se llama el *ancho de banda* o *parámetro de suavización*, el cual puede ser cualquier número *no-negativo*.

Algunos métodos de estimación en regresión no-paramétrica se basan en un diseño de puntos *no-aleatorio*. Un diseño de puntos *no-aleatorio* es un conjunto de valores de la variable explicativa \mathbf{X} , $\{\mathbf{x}_i, i = 1, \dots, n\}$, los cuales se fijan con anterioridad por el investigador. Luego se observan los valores de la respuesta y_i en el diseño de puntos.

También es posible definir estimadores en regresión no-paramétrica que se basen en un diseño de puntos *aleatorio*.

De acuerdo con *Eubank (1999)*, un ejemplo muy simple de un estimador lineal es el *regresograma*, un *estimador* utilizado en el análisis exploratorio de datos que debe su origen a *Kotz, Jhonsosn y Read (1988)*. Definamos explícitamente este estimador, sea λ un entero positivo y el intervalo cerrado $[0,1]$ se divide en λ sub-intervalos de la forma:

$$P_i = \left[\frac{i-1}{\lambda}, \frac{i}{\lambda} \right], i=1, \dots, \lambda-1 \text{ y } P_\lambda = \left[\frac{\lambda-1}{\lambda}, 1 \right].$$

Entonces, el *regresograma* se obtiene de (1.2) usando:

$$K(x, x_i; \lambda) = \frac{\sum_{T=1}^{\lambda} I_{P_T}(x) I_{P_T}(x_i)}{\sum_{j=1}^n \sum_{T=1}^{\lambda} I_{P_T}(x) I_{P_T}(x_j)}$$

Con $I_{P_T}(\cdot)$ la función indicador para el intervalo P_T .

Otro ejemplo de un estimador de este tipo es el estimador de *Nadaraya-Watson* (Nadaraya y Seckler, 1964; Watson, 1964):

$$\mu_\lambda(x) = \frac{\sum_{i=1}^n K(\lambda^{-1}(x - x_i)) y_i}{\sum_{i=1}^n K(\lambda^{-1}(x - x_i))} \quad (1.3)$$

Según *Eubank (1999)*, este estimador pertenece a una clase de estimadores lineales conocidos como estimadores *Kernel*. Los estimadores *Kernel* son estimadores lineales en el sentido que pueden expresar el valor del estimador de la función $\mu(\mathbf{x})$ en cualquier punto \mathbf{x}_i como una suma ponderada de los valores de las respuestas \mathbf{y}_i . En un estimador *Kernel* los pesos se derivan de una función *Kernel* común que no depende del

diseño de puntos $\{x_i, i = 1, \dots, n\}$. La función Kernel se denota por K , y en general una función de densidad de probabilidad simétrica alrededor de cero y cuyo pico ocurre en cero. La función de asignación de pesos disminuye a cero a medida que se aleja del punto de estimación x .

Un método para producir la función de pesos con ciertas propiedades deseadas es usar pesos de la forma:

$$K(x, x_i; \lambda) = \frac{1}{\lambda} K\left(\frac{x - x_i}{\lambda}\right), \text{ , donde } \lambda \text{ es cualquier número no-negativo,} \quad (1.4)$$

con una función K definida en $[-1, 1]$. Si la función es simétrica y apuntada en cero, entonces y_i contribuye más a la estimación de $\mu(x)$ en el punto x_i asociado con y_i . Debe ser claro, que una buena elección del *ancho de banda* parece ser muy importante para obtener un ajuste satisfactorio de los datos usando un estimador Kernel.

Generalmente se requieren funciones Kernel que cumplan con las siguientes condiciones:

$$\int_{-1}^1 uK(u)du = 0 \quad , \quad \int_{-1}^1 K(u)du = 1 \quad , \quad \int_{-1}^1 u^2 K(u)du \neq 0 \quad \text{y} \quad \int_{-1}^1 K(u)^2 du < \infty \quad (1.5)$$

Una función Kernel que satisface (1.5) corresponde a la densidad de alguna distribución con media cero y varianza finita. Esta clase de funciones son definidas como *funciones Kernel de segundo orden*.

Algunas funciones de este tipo son el *Kernel uniforme* $K(u) = \frac{1}{2} I_{[-1,1]}(u)$, *Kernel*

cuadrático $K(u) = \frac{3}{4}(1 - u^2)I_{[-1,1]}(u)$ y el *Kernel Biweight*

$$K(u) = \frac{15}{16}(1 - u^2)^2 I_{[-1,1]}(u).$$

Un estimador lineal que se obtiene de (1.4) es

$$\mu_{\lambda}(x) = (n\lambda)^{-1} \sum_{i=1}^n K(\lambda^{-1}(x - x_i)) y_i \quad (1.6)$$

Sin embargo encontramos algunas variantes de la forma general del estimador lineal (1.6), el estimador de *Nadaraya-Watson* (Nadaraya y Seckler, 1964; Watson, 1964) es un ejemplo.

Otra forma del estimador lineal (1.6) es el estimador de *Gasser y Müller* (1979)

$$\mu_{\lambda}(x) = \sum_{i=1}^n (\lambda^{-1} \int_{S_{i-1}}^{S_i} K(\lambda^{-1}(x - s)) ds) y_i \quad (1.7)$$

Con $S_0 = 0$, $S_n = 1$ y $S_i = (x_{i+1} + x_i) / 2$, $i = 1, \dots, n-1$.

Los estimadores de *Gasser y Müller* (1979) y *Nadaraya-Watson* (Nadaraya y Seckler, 1964; Watson, 1964) no son estimadores insesgados de $\mu(\mathbf{x})$, más sin embargo son estimadores consistentes. Bajo las condiciones (1.5), para un λ fijo y $\mu(\mathbf{x})$ una función suave, $\mu_{\lambda}(x)$ es asintóticamente insesgado sí $\lambda \rightarrow 0$ cuando $n \rightarrow \infty$ y también será consistente sí $n\lambda \rightarrow \infty$ cuando $n \rightarrow \infty$.

Otros ejemplos de estimadores de este tipo son: el estimador de *Series Ortogonales*, estimadores mediante *Polinomios Locales*, Suavización a través de *Splines*.

1.2 CONSISTENCIA

Como vemos en el estimador (1.2) es posible pensar en sugerir diferentes estimadores para la función de regresión. Se puede pensar ahora en un nuevo problema: las propiedades que debería poseer un estimador de $\mu(\mathbf{x})$. Como previamente se nombró, no se puede esperar que un estimador en regresión no-paramétrica sea insesgado sí la función $\mu(\mathbf{x})$ es simplemente un elemento arbitrario de \mathbf{W} . Sin embargo, se busca un estimador que sea casi insesgado (es decir, que presente un sesgo muy pequeño) para un tamaño de muestra n grande sobre todas las funciones de regresión en \mathbf{W} . Lo anterior relaciona la *propiedad de consistencia*. Cuando n crece (aumenta el tamaño de muestra)

cualquier estimador razonable de $\mu(x)$ debe converger a $\mu(x)$ con algún grado o nivel probabilístico.

En la actualidad, es posible encontrar estimadores consistentes de una curva de regresión aun cuando se formulen supuestos no-paramétricos respecto de su forma. También estos estimadores no-paramétricos pueden tener algunas propiedades de eficiencia.

1.3 ALGUNOS CRITERIOS PARA DETERMINAR UN BUEN ESTIMADOR

Uno de los problemas en consideración es la elección de la mejor estimación de $\mu(x)$. Para escoger el *mejor* estimador se debe evaluar el estimador. Para ello, definiremos los siguientes indicadores,

$$L(\mu_\lambda(x)) = n^{-1} \sum_{i=1}^n (\mu_\lambda(x_i) - \mu(x_i))^2, \text{ es la pérdida en la estimación de } \mu(x). \quad (1.8)$$

$L(\mu_\lambda(x))$ representa una medida de la proximidad o cercanía de $\mu_\lambda(x)$ a $\mu(x)$.

Para determinar un *buen estimador* de $\mu(x)$, se mide el *riesgo promedio*:

Tenemos $R(\mu_\lambda(x)) = E(L(\mu_\lambda(x)))$

$$R(\mu_\lambda(x)) = n^{-1} \sum_{i=1}^n E(\mu_\lambda(x_i) - \mu(x_i))^2 \quad (1.9)$$

Pequeños valores de $L(\mu_\lambda(x))$ y $R(\mu_\lambda(x))$ indican que $\mu_\lambda(x)$ es un buen estimador. Sí

$R(\mu_\lambda(x))$ converge a *cero* cuando n aumenta esto representa una forma consistente del estimador $\mu_\lambda(x)$.

Un valor de λ el cual minimiza $L(\mu_\lambda(x))$ proporciona un mejor estimador de $\mu(x)$ para un conjunto particular de datos, mientras un valor de λ el cual minimiza $R(\mu_\lambda(x))$ puede ser visto como óptimo para la predicción de futuras respuestas o para la estimación de

$\mu(\mathbf{x})$ en muestreo repetitivo. Un *buen estimador* para este caso, es aquel que presente un *riesgo* lo más bajo posible.

La medida del *riesgo* se puede realizar separándolo en dos componentes:

$$R(\mu_\lambda(x)) = B^2(\mu_\lambda(x_i)) + V(\mu_\lambda(x_i)) \quad (1.10)$$

Donde $B^2(\mu_\lambda(x_i))$ es el sesgo del estimador de $\mu(\mathbf{x})$ y $V(\mu_\lambda(x_i))$ es la varianza del estimador de $\mu(\mathbf{x})$.

Otro criterio cercano a $R(\mu_\lambda(x))$ es el riesgo de predicción (algunas veces llamado *error cuadrático medio de predicción*). Para aplicar este criterio supondremos que se pretende sacar n observaciones nuevas \mathbf{y}_i^* . Al evaluar la representación de $\mu_\lambda(x)$ como un predictor de futuras observaciones calculamos el error cuadrático esperado de $\mu_\lambda(x)$ como un predictor de \mathbf{y}^* .

$$P(\mu_\lambda(x)) = n^{-1} \sum_{i=1}^n E(y_i^* - \mu_\lambda(x_i))^2 \quad (1.11)$$

El *riesgo de predicción* también se puede expresar de la siguiente forma,

$$P(\mu_\lambda(x)) = \sigma^2 + R(\mu_\lambda(x)) \quad (1.12)$$

Podemos ver que un estimador de $\mu(\mathbf{x})$ el cual minimiza $R(\mu_\lambda(x))$, también minimiza a $P(\mu_\lambda(x))$ y lo contrario.

Un estimador óptimo puede ser caracterizado como aquel cuyo valor de λ minimice $R(\mu_\lambda(x))$ (equivalentemente $P(\mu_\lambda(x))$).

Desafortunadamente $L(\mu_\lambda(x))$, $R(\mu_\lambda(x))$ y $P(\mu_\lambda(x))$ no pueden ser calculados sin el conocimiento de $\mu(\mathbf{x})$. Así, en la práctica el interés será la estimación de estos criterios con respecto a los datos observados y las estimaciones entonces serán minimizadas con

respecto a λ . $R(\mu_\lambda(x))$ y $P(\mu_\lambda(x))$ proporcionan una medida global del funcionamiento de $\mu(x)$, estos criterios dan una indicación de cómo $\mu_\lambda(x)$ se comporta como un estimador sobre todos los elementos de $\mu(x)$.

En el contexto de regresión no-paramétrica, tenemos que $y_i = \mu(x_i) + \varepsilon_i$, $i = 1, 2, \dots, n$. Donde $\mu(x)$ es una función de regresión desconocida suave y $\{x_i, i = 1, \dots, n\}$ es un diseño de puntos en el intervalo $[a, b]$; así, $R(\mu_\lambda(x))$ o $P(\mu_\lambda(x))$ proporcionan descripciones de la eficacia de un estimador de la función de regresión sobre el intervalo cerrado de observaciones.

De acuerdo con *Eubank (1999)*, un estimador de $\mu(x)$ es *asintóticamente óptimo* sí, $\mu(x) \in \mathbf{W}$, $R(\mu_\lambda(x))$ disminuye cercano a $n^{-\delta}$ para algún $\delta > 0$ cuando n aumenta y ningún otro estimador puede lograr una mejor razón de convergencia. En el caso de modelos paramétricos se encuentra que $\delta = 1$ bajo condiciones de regularidad normales. Sin embargo, esto no es válido para modelos no-paramétricos y generalmente $\delta < 1$ en este último caso.

Observando una de las características del *parámetro de suavización* (λ) en la estimación *no-paramétrica* para un n dado y $\lambda < 1$, algunos estimadores pueden presentar una mayor varianza y un sesgo pequeño. En general, valores pequeños de λ causan en los estimadores lineales una mayor varianza y sesgos mínimos y viceversa. Este fuerte impacto en los estimadores, por causa de λ pequeños producirá estimadores muy variables, mientras que para λ grandes se pueden producir estimadores más similares (*suaves*) pero también estimadores muy sesgados.

1.4. SELECCIÓN DEL PARÁMETRO DE SUAIVIZACIÓN

El problema de decidir acerca del *parámetro de suavización* λ es de gran importancia en el análisis de regresión no-paramétrico. Merece la pena mencionar que la selección del parámetro de suavización siempre se relaciona a una cierta interpretación de

suavidad. Una buena selección del parámetro siempre será un punto de inicio muy útil en un análisis de regresión no-paramétrico.

Una de las ventajas de una selección automática del parámetro de suavización por medio del método de *suavización Kernel* es que la comparación entre varias pruebas puede ser realizada sobre la base de un método estandarizado. Una ventaja adicional de un método automático queda en la aplicación de modelos aditivos para investigaciones de análisis de regresión de datos de alta dimensionalidad.

La elección del parámetro de suavización se hace bajo algún criterio global con respecto al error cuadrático, el cual se busca minimizar. Varias posibilidades se pueden presentar de acuerdo a la elección del parámetro de suavización en la *curvatura local* de la función de regresión.

Según *Härdle* (1990), el método de *suavización Kernel* como estimador de la función de regresión o de sus derivadas esta en función de K (función Kernel) y del parámetro λ . La efectividad de la función de pesos $\{K(\mathbf{x}, \mathbf{x}_i; \lambda)\}$, $i = 1, \dots, n$, del método de suavización Kernel es determinada por la función K y el parámetro λ .

La exactitud de la curva estimada de $\mu(\mathbf{x})$ no depende solo del parámetro λ exclusivamente, más precisamente, depende del par (K, λ) . El estudio del comportamiento de las medidas de distancia del error cuadrático para este método de estimación busca minimizar el error. Estas medidas de distancia están en función de K , la variación de estas medidas de distancia como una función de K puede ser debida al problema de encontrar un buen parámetro de suavización.

Realmente la exactitud de este método para obtener una mejor precisión de la curva estimada de $\mu(\mathbf{x})$ es más un problema en la elección del parámetro de suavización λ . En general, estos procedimientos para seleccionar λ buscan optimizar la medida del error cuadrático para la curva de regresión y sus derivadas. Típicamente, el sesgo de los estimadores no-paramétricos incrementa con este *parámetro de suavización* mientras la varianza decrece cuando este parámetro tiende a ser grande.

2. SELECCIÓN DEL PARÁMETRO DE SUAVIZACIÓN EN EL PROBLEMA DE SELECCIÓN DE VARIABLES EN REGRESIÓN NO-PARAMÉTRICA

2.1 EL PROBLEMA DE LA SELECCIÓN DE VARIABLES

Existen algunos problemas del mundo real en el marco no-paramétrico, donde la variable explicativa \mathbf{X} presenta una dimensionalidad mayor a 1 (el caso unidimensional), es decir, consideramos el caso *p-dimensional*. Para este caso, las estimaciones no-paramétricas multivariadas son necesariamente más complicadas que las estimaciones univariadas debido a que se tienen muchas formas diferentes para encontrar estimaciones de la curva de regresión y se deben seleccionar más parámetros de suavización.

Según *Olaya (2000)*, para el caso *p-dimensional* los estimadores son cada vez más inexactos (a medida que se incrementa la dimensionalidad de la variable explicativa \mathbf{X}), consecuentemente, se requieren tamaños de muestra muy grandes para lograr estimaciones más confiables. Lo anterior se conoce como la *maldición de la dimensionalidad*.

En muchos casos es conveniente buscar la forma de reducir la dimensionalidad del análisis de regresión no-paramétrico en alguna dimensión menor que p , es decir, $k < p$.

Al reducir la dimensión del análisis de regresión no-paramétrico ($k < p$) se reduce el efecto de la *maldición de la dimensionalidad* y, desde luego, encontraremos un modelo más sencillo y más fácil de aplicar.

Algunos investigadores como *Olaya (2000)*, definen como variable explicativa a cada una de las componentes de la variable explicativa de *dimensión p*, es decir, la meta sería reducir el número de variables explicativas de p en algún k , con k lo más pequeño

posible con respecto a p . Este tipo de problemas se conoce como *el problema de la selección de variables*.

Para el caso paramétrico, en la actualidad muchos métodos razonables han sido desarrollados y validados. *Draper y Smith (1966, 1981, 1998)*, han desarrollado algunos problemas para el caso de selección de variables en el marco paramétrico.

En el caso no-paramétrico es diferente, el *problema de la selección de variables* en un análisis de regresión no-paramétrica esta todavía en una etapa exploratoria. Existen algunos métodos que han sido propuestos para *reducir la dimensionalidad*, pero muy pocos métodos se han propuesto para resolver el problema de la selección de variables (reducir el número de variables explicativas).

De acuerdo con *Olaya (2000)*, se puede reducir la dimensionalidad en un análisis de regresión no-paramétrico sin reducir realmente el número de variables explicativas originales en el modelo. Un método consiste en utilizar k *combinaciones lineales* de todas las variables explicativas del modelo inicial (p variables originales), donde $k < p$. Al aplicar este método se reduce la dimensión del análisis de regresión de p a dimensión k . Es importante tener claro que se deben medir todas las variables explicativas para aplicar el modelo de dimensión k , el cual es construido con las k *combinaciones lineales*. De acuerdo a esto, la expresión "*el problema de la selección de variables*" se refiere en forma muy clara y explícita al problema de conservar en el modelo reducido (de dimensión k) todas las p variables explicativas del modelo original, es decir, del modelo de dimensión p .

Se pueden citar algunos procedimientos para *reducir la dimensionalidad* en un análisis de regresión. El procedimiento SIR (*Sliced Inverse Regression*) (*Li, K. C. 1991*), reduce la dimensión de un análisis de regresión de una forma eficiente conservando en el modelo reducido (de dimensión k) las p variables explicativas del modelo original en función de k *combinaciones lineales*.

Otro procedimiento para reducir la dimensión en un análisis de regresión es el procedimiento GAM (*Generalized Additive Model*) (Hastie y Tibshirani, 1990). Este método permite representar la función del modelo de regresión como una suma de funciones univariadas sencillas, una función por cada variable explicativa.

Existen otros procedimientos, entre ellos SIM (*Single Index Models*) (Härdle et al. 1993), el procedimiento PPR (*Projection Pursuit Regression*) (Friedman y Stuetzle, 1981; Hall, 1989). Härdle y Korostelev (1996) propusieron un método *ad hoc* para reducir el número de variables explicativas en un procedimiento GAM. Los supuestos del método determinan que solamente $k < p$ componentes no-paramétricas de GAM son diferentes de cero. Zhang (1990) propuso un método para seleccionar k variables ($k < p$) en un análisis de regresión no-paramétrico. Olaya y Kulasekera (2000) proponen métodos que no requieren suavización multidimensional en ningún paso y que se basan en pruebas de hipótesis formales.

Centraremos nuestra discusión en el *método para seleccionar variables en regresión no-paramétrica* propuesto por Olaya (2000). El método propuesto permite seleccionar variables significativas en el modelo. En el proceso de selección se aplica una prueba formal de hipótesis de igualdad de dos curvas de regresión en cada etapa del proceso de selección, donde las variables seleccionadas tienen orden de importancia de acuerdo al criterio del investigador.

El método de Olaya (2000) y Olaya-Kulasekera (2000) utiliza eficazmente algunos estadísticos diseñados para realizar pruebas formales de comparación de dos curvas de regresión que se encuentran en la literatura (Kulasekera, 1995). Mostraremos como funciona este procedimiento propuesto por Olaya (2000).

2.1.1 Un método para seleccionar variables en regresión no-paramétrica

Supóngase que se tiene un conjunto de observaciones de una variable aleatoria continua Y de n valores determinados por p variables aleatorias continuas $x_1, x_2, x_3, \dots, x_p$, donde Y es la variable dependiente o de *respuesta*, X es la variable independiente o

variable *explicativa*. Sea $(\mathbf{x}_i, \mathbf{y}_i) = (\mathbf{x}_{1i}, \mathbf{x}_{2i}, \mathbf{x}_{3i}, \dots, \mathbf{x}_{pi}, \mathbf{y}_i)$, para $i = 1, 2, \dots, n$ las n observaciones de (\mathbf{x}, \mathbf{y}) , y se asume que están relacionadas por el modelo,

$$\mathbf{y}_i = \boldsymbol{\mu}(\mathbf{x}_{1i}, \mathbf{x}_{2i}, \mathbf{x}_{3i}, \dots, \mathbf{x}_{pi}) + \boldsymbol{\varepsilon}_i, \quad i = 1, 2, \dots, n. \quad (2.1)$$

Donde $E(\boldsymbol{\varepsilon}_i) = 0$ y $V(\boldsymbol{\varepsilon}_i) = \sigma^2 < \infty$. La función de regresión $\boldsymbol{\mu}(\mathbf{x})$ es desconocida, se asume que esta función es continua y diferenciable (al menos presenta dos derivadas continuas). La covariable $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_p)^T$ esta definida en el intervalo cerrado de \mathfrak{R}^p . Tomaremos como dominio de la covariable \mathbf{X} el intervalo cerrado $[0, 1]^p$ para facilitar su aplicación técnica. Las p variables aleatorias continuas $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_p$ están ordenadas de acuerdo a su importancia, es decir, \mathbf{x}_1 es más importante que $\mathbf{x}_2, \dots, \mathbf{x}_{p-1}$ es más importante que \mathbf{x}_p . Otro supuesto es que la covariable \mathbf{X} es independiente del error $\boldsymbol{\varepsilon}_i$.

Inicialmente asumiremos que $\boldsymbol{\mu}(\mathbf{x})$ es una función propia de $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_{r-1}$. Se desea probar si \mathbf{x}_r debe incluirse en el modelo, es decir, queremos probar si \mathbf{x}_r es una variable significativa en la función $\boldsymbol{\mu}(\mathbf{x})$, donde $r \leq p$.

Entonces se plantean las siguientes hipótesis:

$$H_0: \boldsymbol{\mu}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_r) = \boldsymbol{\mu}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_{r-1})$$

$$H_a: \boldsymbol{\mu}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_r) \neq \boldsymbol{\mu}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_{r-1})$$

Si rechazamos la hipótesis nula H_0 , entonces \mathbf{x}_r debería ser incluida en el modelo $\boldsymbol{\mu}(\mathbf{x})$.

Para el caso $r = 2$, las hipótesis de prueba serían:

$$H_0: \boldsymbol{\mu}(\mathbf{x}_1, \mathbf{x}_2) = \boldsymbol{\mu}(\mathbf{x}_1)$$

$$H_a: \boldsymbol{\mu}(\mathbf{x}_1, \mathbf{x}_2) \neq \boldsymbol{\mu}(\mathbf{x}_1) \quad (2.2)$$

Para realizar este tipo de pruebas se propone el uso del método de *Cuasi-Residuales* propuesto por *Kulasekera* (1995). El método de *Cuasi-Residuales* compara dos curvas de regresión. Para encontrar las dos curvas de regresión, el conjunto de observaciones se divide en dos subconjuntos de datos de acuerdo a los valores de \mathbf{x}_r . La división se realiza dependiendo si $\mathbf{x}_r \in Z$ o $\mathbf{x}_r \notin Z$, donde Z es un subconjunto que se escoge del

dominio de la variable \mathbf{x}_r de una manera apropiada. En muchas aplicaciones, los elementos del conjunto Z pueden ser aquellos valores que toma la variable \mathbf{x}_r que están por debajo de su punto medio. En otras situaciones prácticas los elementos del conjunto Z pueden ser aquellos valores que toma la variable \mathbf{x}_r que están alrededor del punto medio del dominio de la variable \mathbf{x}_r . Tenemos entonces dos subconjuntos de datos $\{\mathbf{x}_{1j}, \mathbf{x}_{2j}, \mathbf{y}_j\}$, $j = 1, \dots, n_1$ para aquellos valores de $\mathbf{x}_2 \in Z$ y $\{\mathbf{x}_{1m}, \mathbf{x}_{2m}, \mathbf{y}_m\}$, $m = 1, \dots, n_2$ para aquellos valores de $\mathbf{x}_2 \in Z^c$, donde Z^c es el complemento del conjunto Z .

Ahora, se modela Y contra \mathbf{x}_1 utilizando un método de suavización en ambos subconjuntos y luego se comparan las dos curvas de regresión obtenidas $\mu_Z(\mathbf{x}_1)$ y $\mu_{Z^c}(\mathbf{x}_1)$.

Donde,

$$Y_Z = \mu_Z(\mathbf{x}_1) + \varepsilon, \text{ para aquellos valores de } \mathbf{x}_2 \in Z \quad (2.3)$$

$$Y_{Z^c} = \mu_{Z^c}(\mathbf{x}_1) + \varepsilon, \text{ para aquellos valores de } \mathbf{x}_2 \in Z^c$$

Entonces se comparan las dos curvas de regresión por medio de las siguientes hipótesis:

$$H_0: \mu_Z(\mathbf{x}_1) = \mu_{Z^c}(\mathbf{x}_1)$$

$$H_a: \mu_Z(\mathbf{x}_1) \neq \mu_{Z^c}(\mathbf{x}_1) \quad (2.4)$$

Si rechazamos la hipótesis nula H_0 , entonces \mathbf{x}_2 debería incluirse en el modelo. Esto significa que \mathbf{x}_2 tiene un papel importante en el modelo de regresión.

Para el caso $r > 2$ no existen métodos desarrollados eficientes para comparaciones multidimensionales de este tipo en un análisis *no-paramétrico*. En la práctica, para los casos donde la dimensión de la covariable \mathbf{X} es mayor a 2, es decir, consideramos el caso *p-dimensional*, se propone reducir el problema a dos dimensiones escogiendo una apropiada *combinación lineal* de las primeras $r-1$ covariables que están incluidas en el modelo inicial como la primera variable explicativa \mathbf{x}_1 y la covariable \mathbf{x}_r que es la

variable candidata como la segunda variable explicativa \mathbf{x}_2 . Así, obtenemos un problema *bidimensional*. En este caso particular, se pueden usar métodos de prueba para la comparación de dos curvas de regresión evaluando la importancia de \mathbf{x}_r (segunda variable explicativa) en el modelo de regresión. Específicamente, definimos una combinación lineal de la forma:

$$U_i = \sum_{k=1}^{r-1} c_k X_{ki} = c_1 \mathbf{x}_{1i} + c_2 \mathbf{x}_{2i} + \dots + c_{r-1} \mathbf{x}_{r-1i}$$
, es el valor de U asociado con la i -ésima observación. Donde las c_k son constantes reales escogidas convenientemente.

El conjunto original de observaciones $\{\mathbf{x}_{1i}, \mathbf{x}_{2i}, \dots, \mathbf{x}_{ri}, \mathbf{y}_i\}$, $i = 1, \dots, n$ es transformado en un conjunto de nuevas observaciones $\{U_i, \mathbf{x}_{ri}, \mathbf{y}_i\}$, $i = 1, \dots, n$. Al realizar esta transformación obtenemos una covariable bidimensional $\{U, \mathbf{x}_r\}$. Para encontrar la mejor combinación lineal se utiliza el primer paso del procedimiento PPR (*Projection Pursuit Regression*), porque PPR selecciona esta combinación lineal teniendo en cuenta tanto las respuestas Y como la variable explicativa \mathbf{X} . El procedimiento PPR fue propuesto por *Friedman y Stuetzle (1981)* en la aproximación de funciones de alta dimensionalidad, $\mu(\mathbf{x}_i) = E(Y_i | \mathbf{X}_i = \mathbf{x}_i)$, $\mathbf{x}_i \in \mathfrak{R}^p$ por simples funciones que operan en espacios de baja dimensión, típicamente espacios *unidimensionales*. Este procedimiento aproxima la superficie de la función de regresión como una suma de funciones univariadas desconocidas suaves $g(c'_m \mathbf{x})$ de combinaciones lineales de las variables predictoras de una manera iterativa. Donde,

$$\varphi(\mathbf{x}_i) = \sum_{m=1}^M g(c'_m \mathbf{x}), \text{ donde } c'_m \text{ es el vector de constantes reales escogidas}$$

convenientemente por el procedimiento PPR. PPR es más general que el procedimiento *paso a paso* estándar de un análisis de regresión, no requiere la definición de un espacio para la variable explicativa. La aproximación de $\varphi(\mathbf{x}_i)$ se construye de manera iterativa, siguiendo el siguiente algoritmo (*Friedman y Stuetzle, 1981*):

1. Asignación de residuales e inicio del término de conteo

$$r_i \leftarrow y_i, i = 1, 2, \dots, n$$

$$M \leftarrow 0$$

(Se asume que las respuestas y_i están centradas, $\sum y_i = 0$)

2. Buscar el siguiente término en el modelo.

Para una combinación lineal dada $\mathbf{U}_i = \sum_{k=1}^{r-1} c_k X_{ki}$, construir una función suave $g(\mathbf{U}_i)$ de los residuales ordenados en valores ascendentes de \mathbf{U}_i . Tomar como *criterio de ajuste* para esta combinación lineal la fracción:

$$I(c) = 1 - \frac{\sum_{i=1}^n (r_i - g(c'x))^2}{\sum_{i=1}^n r_i^2}$$

Encontrar el vector de coeficientes c_{m+1} que maximice $I(c)$, y la correspondiente función $g(c'_{m+1}\mathbf{x})$.

3. Si el *criterio de ajuste* es menor que un valor específico (determinado por el investigador) parar (terminar el proceso de iteración), entonces el último término no es incluido en el modelo final. En caso contrario, actualizar los residuales y el término de conteo.

$$r_i \leftarrow r_i - g(c'_{m+1}\mathbf{x}_i), i = 1, 2, \dots, n$$

$$\mathbf{M} \leftarrow \mathbf{M} + 1$$

Regresar al paso 2.

Desde el punto de vista computacional PPR es absolutamente factible de aplicar. Una característica importante de PPR es que los resultados de cada iteración se pueden representar gráficamente, facilitando la interpretación. Esta salida gráfica se puede utilizar para ajustar los parámetros principales del procedimiento, entre ellos el ancho de banda $\hat{\lambda}$ y el valor del criterio de ajuste para la terminación. El procedimiento PPR se puede aplicar claramente a los residuales de cualquier modelo inicial.

PPR (Regresión por Búsqueda de la Proyección) construye un modelo de la superficie de la función de regresión, basado en proyecciones de los datos sobre planos construidos por la variable de respuesta \mathbf{Y} y una combinación lineal $c \cdot \mathbf{x}$ de la covariable explicativa. Aquí el *criterio de ajuste* para una proyección es la fracción de la variación explicada por una suavización de \mathbf{Y} contra $c \cdot \mathbf{x}$. La estructura cambia

formando los residuales de la suavización substituyéndolo por la respuesta. El modelo en cada iteración es la suma de funciones suaves que fueron restadas previamente y así construir la estructura encontrada hasta ahora.

Ahora, el modelo resultante es el siguiente,

$$\mathbf{Y} = \mu(\mathbf{U}, \mathbf{x}_r) + \varepsilon^* \quad (2.5)$$

Para alguna función μ , donde \mathbf{Y} , \mathbf{U} , \mathbf{x}_r y ε^* son variables aleatorias.

Se deben satisfacer las siguientes dos condiciones:

- a. μ en el modelo es una función propia de \mathbf{U} .
- b. ε^* en el modelo satisface que $E(\varepsilon^*) = 0$ y $V(\varepsilon^*) < \infty$.

Se debe definir una función μ en el modelo y verificar esta función con una apropiada condición sobre la estructura original del error satisfaciendo las dos condiciones anteriores.

En este caso, se asume que existen μ y ε^* que cumplen con las condiciones del modelo anterior.

Supóngase que se tiene una variable explicativa de dimensión r y $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_r$ son variables aleatorias definidas en $[0, 1]$ tales que las primeras $r-1$ juegan un papel muy importante en la función de regresión; ahora se debe probar si la variable \mathbf{x}_r debería ser incluida en el modelo. Supóngase que se observa si $\mathbf{x}_r \in Z$ o $\mathbf{x}_r \in Z^c$, entonces se puede dividir el conjunto de observaciones $\{\mathbf{U}_i, \mathbf{x}_{ri}, \mathbf{y}_i\}$, $i = 1, \dots, n$ en dos subconjuntos $\{\mathbf{U}_j, \mathbf{x}_{rj}, \mathbf{y}_j\}$, $j = 1, \dots, n_1$ y $\{\mathbf{U}_m, \mathbf{x}_{rm}, \mathbf{y}_m\}$, $m = 1, \dots, n_2$.

De acuerdo a lo anterior, se pueden formar los siguientes modelos:

$$\begin{aligned} Y_Z &= \mu_Z(\mathbf{U}) + \varepsilon^*, \text{ para aquellos valores de } \mathbf{x}_r \in Z \\ Y_{Z^c} &= \mu_{Z^c}(\mathbf{U}) + \varepsilon^*, \text{ para aquellos valores de } \mathbf{x}_r \in Z^c \end{aligned} \quad (2.6)$$

Se encuentra una relación de la variable de respuesta Y con la covariable U en los dos conjuntos. Ahora, se pueden probar las siguientes hipótesis:

$$H_0: \mu_Z(\mathbf{U}) = \mu_Z^c(\mathbf{U})$$

Contra

(2.7)

$$H_a: \mu_Z(\mathbf{U}) \neq \mu_Z^c(\mathbf{U})$$

para alguna función μ propia de U , donde $\mu_Z(\mathbf{U})$ y $\mu_Z^c(\mathbf{U})$ son las curvas de regresión de la variable de respuesta Y contra U en los dos subconjuntos. La igualdad de las curvas de regresión se prueba usando el método de los *Cuasi-Residuales* (Kulasekera, 1995). Rechazar la hipótesis nula H_0 indica que x_r tiene un papel importante en el modelo de regresión μ . De acuerdo a lo anterior, el modelo $Y = \mu(\mathbf{U}, \mathbf{x}_r) + \varepsilon^*$ será significativo si la variable de respuesta Y puede ser explicada por las covariables U y \mathbf{x}_r a través de la función μ .

La siguiente definición para la función μ satisface las condiciones a y b del modelo (2.5)

- **Definición**

Sea $U = \sum_{k=1}^{r-1} c_k X_k$, donde c es un vector de números reales. Entonces $\mu(U)$ está dada por,

$$\mu(U) = E[\mu(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_r) | U]$$

Donde $\mu(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_r)$ es una función propia continua de variables aleatorias $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_r$ definidas en el intervalo cerrado $[0, 1]^P$. Para el caso de selección de variables, si x_r no tiene un papel importante en el modelo de regresión, conociendo que $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_{r-1}$ están incluidas en el modelo, la combinación lineal $U = \sum_{k=1}^{r-1} c_k X_k$ será usada en la predicción de la variable de respuesta Y a través del modelo $Y = \mu^*(U) + \varepsilon^*$ donde $\mu^*(U) = E[\mu(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_r) | U]$ y $\mu_Z(\mathbf{U}) = \mu_Z^c(\mathbf{U}) = \mu^*(U)$ para cualquier conjunto de elementos Z .

El siguiente teorema verifica la condición **a** del modelo (2.5) bajo la hipótesis nula
 $H_0: \mu_Z(\mathbf{U}) = \mu_Z^c(\mathbf{U})$

- **Teorema 1**

Sea $\mu(\mathbf{x})$ una función propia continua de variables aleatorias $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_{r-1}$. Existe un vector $\mathbf{c} = (c_1, c_2, \dots, c_{r-1})^T$ tal que $\mu^*(\mathbf{U}) = E[\mu(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_{r-1}) | \mathbf{U}]$ es una función propia de \mathbf{U} donde $\mathbf{U} = \sum_{k=1}^{r-1} c_k X_k$

El teorema anterior determina la existencia de un vector \mathbf{c} y una combinación lineal \mathbf{U} tal que proporciona un modelo significativo relacionando la variable de respuesta \mathbf{Y} con la covariable \mathbf{U} .

El siguiente Lema verifica la condición **b** del modelo (2.5).

- **Lema 1**

Sea $\mu(\mathbf{U}) = E[\mu(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_r) | \mathbf{U}]$. Entonces $E(\epsilon^*) = 0$, y $V(\epsilon^*) < \infty$.

La demostración del teorema 1 y el lema 1 aparecen en *Olaya (2000)*.

La prueba de *Cuasi-Residuales (Kulasekera, 1995)* para igualdad de curvas de regresión se construye bajo el supuesto de que las dos curvas de regresión tienen al menos dos derivadas continuas sobre el dominio de la covariable \mathbf{X} , excepto posiblemente en un número finito de puntos sobre el dominio común. Según *Kulasekera (1995)* las pruebas de hipótesis formales (comparación de dos curvas de regresión) usan muestras independientes para las dos poblaciones, donde las pruebas no requieren un *diseño de puntos común* o igual tamaños de muestra.

Al realizar la prueba de hipótesis de igualdad de dos curvas de regresión usaremos algunos estadísticos de prueba W_i , $i = 1, 2, 3$ dados en *Kulasekera (1995)* que se acomoden a tamaños de muestras diferentes y diseño de puntos de covariables no

idénticos. Las primeras dos pruebas, las cuales denotamos por W_1 y W_2 , usan el peso promedio de la suma parcial de cuadrados de los *Cuasi-Residuales*.

Los *Cuasi-Residuales* son definidos como,

$$e_{Zj} = Y_{Zj} - \hat{\mu}_Z^c(x_j), \quad j = 1, 2, \dots, n_1$$

y (2.8)

$$e_{Zm}^c = Y_{Zm}^c - \hat{\mu}_Z^c(x_m), \quad m = 1, 2, \dots, n_2$$

Donde $\hat{\mu}_Z(x_j)$ y $\hat{\mu}_Z^c(x_m)$ son estimadores lineales confiables, tales como estimadores Kernel, de μ_Z y μ_Z^c respectivamente, cada uno basado solo en las observaciones de la correspondiente población, es decir, para cada subconjunto de datos. Las pruebas W_1 y W_2 son aplicadas eficazmente para n_1 y n_2 suficientemente grandes y bajo la hipótesis nula, e_{Zj} y e_{Zm}^c tienen distribuciones aproximadas a las de ε y ε^* . La distribución de estos *Cuasi-Residuales* depende de la diferencia $\mu_Z - \mu_Z^c$, bajo la hipótesis alterna.

La tercera prueba denotada por W_3 , es una prueba basada en el estimador de la varianza de la distribución del error. Al realizar la prueba de hipótesis (2.7), podemos usar los tres estadísticos de prueba W_1 , W_2 y W_3 desarrolladas en *Kulasekera (1995)*.

Sean los *Cuasi-Residuales* definidos en (2.8), donde los estimadores lineales de μ_Z y μ_Z^c son un tipo de estimadores Kernel discutidos en *Eubank (1999)* y *Müller (1979)*.

Para los valores de la covariable \mathbf{X} , con \mathbf{x}_j , $j = 1, 2, \dots, n_1$ y \mathbf{x}_m , $m = 1, 2, \dots, n_2$ donde el dominio de la covariable \mathbf{X} es el intervalo cerrado $[0, 1]$, definimos

$$\hat{\mu}_Z(x) = \sum_{j=1}^{n_1} \frac{Y_{Zj} W_{1j}}{\lambda}$$

y (2.9)

$$\hat{\mu}_{Z^c}(x) = \sum_{m=1}^{n_2} \frac{Y_{Z^c m} W_{2m}}{\lambda}$$

donde $w_{1j} = \int_{S_{j-1}}^{S_j} K(\lambda^{-1}(x-u))du$, $w_{2m} = \int_{S_{m-1}}^{S_m} K(\lambda^{-1}(x-u))du$

Con $S_0 = 0$, $S_{n_1} = S_{n_2} = 1$, $S_j = (\mathbf{x}_{j+1} + \mathbf{x}_j) / 2$, $j = 1, \dots, n_1 - 1$

y $S_m = (\mathbf{x}_{m+1} + \mathbf{x}_m) / 2$, $m = 1, \dots, n_2 - 1$. Donde K es una función de densidad simétrica conocida en $[-1, 1]$ que es una función Kernel de *segundo orden*. Usando los estimadores (2.9) de μ_Z y μ_{Z^c} definimos la suma parcial,

$$\omega_{n_1}(t) = \sum_{j=1}^{\lceil n_1 t \rceil} \frac{e_{Z_j}}{\sqrt{n_1}}$$

y

$$\omega_{n_2}(t) = \sum_{m=1}^{\lceil n_2 t \rceil} \frac{e_{Z^c m}}{\sqrt{n_2}}$$

(2.10)

Para $0 < t < 1$. $\lceil n_i t \rceil$ $i = 1, 2$ es la parte entera de $n_i t$.

Tests W_1 , W_2 y W_3 .

Sea $\hat{\sigma}_{n_1}^2$ y $\hat{\sigma}_{n_2}^2$ estimadores consistentes de $V(\epsilon) = \sigma^2$ en (2.3) y definimos,

$$T_1 = \frac{\sum_{k=1}^{n_1} \omega_{n_1}^2(k/n_1)}{n_1 \hat{\sigma}_{n_1}^2} , \quad T'_1 = \frac{\sum_{k=1}^{n_2} \omega_{n_2}^2(k/n_2)}{n_2 \hat{\sigma}_{n_2}^2} \quad (2.11)$$

Bajo la hipótesis alterna, T_1 y T'_1 tienden a ser grandes. Entonces podemos rechazar H_0 si $W_1 = \Delta(T_1, T'_1)$ toma valores grandes para alguna función $\Delta(a, b)$ escogida convenientemente, como por ejemplo $\Delta(T_1, T'_1) = \min(T_1, T'_1)$.

Los estadísticos de prueba T_1 y T'_1 son similares a T_n en *Hall y Hart (1990)*.

Definimos T_2 y T'_2 como,

$$T_2 = \sum_{k=1}^{n_1-1} \frac{\omega_{n_1}^2(k/n_1)e_{Z_{k+1}}}{\sqrt{n_1} \hat{\sigma}_{n_1}^3}, \quad T'_2 = \sum_{k=1}^{n_2-1} \frac{\omega_{n_2}^2(k/n_2)e_{Z^c_{k+1}}}{\sqrt{n_2} \hat{\sigma}_{n_2}^3} \quad (2.12)$$

Bajo la hipótesis alterna, las magnitudes T_2 y T'_2 tienden a ser grandes. Entonces podemos rechazar la hipótesis nula H_0 si $W_2 = \Delta(T_2, T'_2)$ también toma valores grandes para alguna función $\Delta(a, b)$. Esta función $\Delta(a, b)$ puede ser escogida para realizar una prueba unilateral derecha, por ejemplo podemos escoger $\Delta(T_2, T'_2) = T_2 - T'_2$.

La prueba W_3 es basada en la comparación del estimador de la varianza. Este estimador fue discutido por *Hall, Kay y Titterington (1990)*. Usualmente es llamado *estimador de la varianza basado en la diferencia óptima*. El estimador de la varianza del error se aplica en el modelo de regresión no-paramétrico,

$Y_{Z^c_m} = \mu_{Z^c}(\mathbf{x}_{1m}) + \varepsilon_m$, para aquellos valores de $\mathbf{x}_2 \in Z^c$. ($m = 1, 2, \dots, n_2$), donde μ es una función desconocida y los errores ε_m son variables aleatorias independientes e idénticamente distribuidos con media cero y varianza σ^2 . Se asume que las observaciones \mathbf{x}_{1m} han sido ordenadas de acuerdo a su importancia, así que, $\mathbf{x}_{11} < \mathbf{x}_{12} < \dots < \mathbf{x}_{1m}$. Una sucesión de diferencias $\{d_j\}$ es una sucesión de números reales tal que,

$$\sum d_j = 0, \quad \sum d_j^2 = 1, \quad j = 0, 1, \dots, r \quad (2.13)$$

Asumimos que $d_j = 0$ para $j < -m_1$ y $j > m_2$, y $d_{-m_1} d_{m_2} \neq 0$, donde $m_1, m_2 \geq 0$. Entonces tenemos que $m_0 = m_1 + m_2$ es llamado el orden de la sucesión. Usualmente es conveniente tomar $m_1 = 0$ y $m_0 = m_2$. El estimador de σ^2 basado en la sucesión de diferencias es,

$$\hat{\sigma}^2 = \frac{\sum_{k=1}^{m-m_2} \left(\sum_{j=1}^{m_2} d_j Y_{Z^c_{j+k}} \right)^2}{m - m_0} \quad (2.14)$$

Si la función $\mu(\mathbf{x})$ es suave y si el diseño de puntos \mathbf{x}_{1m} toma valores cercanos a medida que el tamaño de muestra incrementa, entonces el efecto de $\mu(\mathbf{x})$ sobre el error

cuadrático medio de $\hat{\sigma}^2$ es insignificante. Los supuestos que $\mu(\mathbf{x})$ tiene acerca de su continuidad y diferenciabilidad más que de suavidad, y la condición que el diseño de puntos \mathbf{x}_{lm} están espaciados regularmente en un intervalo cerrado, es más que suficiente para garantizar que los puntos \mathbf{x}_{lm} están muy cercanos. Bajo estas condiciones el estimador $\hat{\sigma}^2$ depende solo de la distribución del error y de la elección de $\{d_j\}$. Una elección adecuada de $\{d_j\}$ produce una minimización de la varianza σ^2 . Para la diferencia de *primer-orden* solamente la elección disponible de $\{d_0, d_1\}$ es, $(2^{-1/2}, -2^{-1/2})$ o el signo contrario de este vector. Para la diferencia de *segundo-orden* la elección disponible de $\{d_0, d_1, d_2\}$ es $(-6^{-1/2}, (2/3)^{1/2}, -6^{-1/2})$, aunque se puede mostrar que la sucesión $\{(1/4)(5^{1/2} + 1), -(1/2), -(1/4)(5^{1/2} - 1)\}$ funciona sustancialmente mejor.

Ahora, definimos

$$T_3 = \sum_{k=1}^{n_1} \frac{e^2 z k}{n_1}$$

El estadístico de prueba W_3 es definido como,

$$W_3 = \frac{\sqrt{n_1 + n_2} (T_3 - \hat{\sigma}^2)}{\hat{\sigma}_0^2} \quad (2.15)$$

Definimos,

$$\hat{\sigma}_0^2 = \sqrt{\frac{v(\hat{\mathcal{E}}^2)}{\varpi^2} + \frac{a^2}{1 - \varpi^2}}$$

$$\text{Donde } \varpi^2 = \frac{n_1}{n_1 + n_2}, \quad a^2 = \hat{k} \hat{\sigma}^4 + 2 \hat{\sigma}^4 \sum_{i=1}^{n_2-r} \left(\sum_{j=0}^r d_j d_{i+j} \right)^2 \quad \text{y} \quad \hat{k} = \frac{E(\hat{\mathcal{E}}^4)}{\hat{\sigma}^4} - 3$$

Donde $E(\hat{\mathcal{E}}^4)$ y $V(\hat{\mathcal{E}}^2)$ son estimadores consistentes de $E(\mathcal{E}^4)$ y $V(\mathcal{E}^2)$.

El término \hat{k} es un estimador consistente de la *kurtosis* de la distribución del error.

Note que bajo la hipótesis alterna, T_3 tiende a desviarse de σ^2 . Entonces rechazamos la hipótesis nula para valores grandes de W_3 .

Propiedades asintóticas.

Enunciaremos algunas propiedades asintóticas de $T_i, T'_i, i = 1, 2$ y W_3 . Si la función $\Delta(a, b)$ usada en los estadísticos de prueba es continua, entonces la independencia de las muestras puede ser usada junto con los teoremas 2.1.1.2 y 2.1.1.3 (que se enuncian más adelante) en obtener una distribución asintótica de $W_i = \Delta(T_i, T'_i), i = 1, 2$. Para alguna elección de $\Delta(a, b)$, por ejemplo $\Delta(a, b) = \min(a, b)$.

Además de las condiciones impuestas sobre la función Kernel K (1.5), los siguientes supuestos pueden ser necesarios,

- $\partial_1.$ Las funciones μ_Z y μ_Z^c tienen dos derivadas continuas.
- $\partial_2.$ El parámetro de suavización λ usado en la estimación de μ_Z y μ_Z^c satisface $\min(n_1, n_2)\lambda \rightarrow \infty$ y $\min(n_1, n_2)\lambda^4 \rightarrow 0$ cuando $\min(n_1, n_2) \rightarrow \infty$.
- $\partial_3.$ Los diseños de puntos $\{X_{1j}, j = 1, \dots, n_1\}$ y $\{X_{1m}, m = 1, \dots, n_2\}$ se hacen densos en $[-1, 1]$ cuando $\min(n_1, n_2) \rightarrow \infty$.
- $\partial_4.$ El parámetro de suavización λ usado en la estimación de μ_Z y μ_Z^c satisface $\min(n_1, n_2)\lambda \rightarrow \infty$ y $\min(n_1, n_2)\lambda^5 \rightarrow 0$ cuando $\min(n_1, n_2) \rightarrow \infty$.

• **Teorema 2**

Si los supuestos $\partial_1 - \partial_3$ se satisfacen. Entonces, bajo la hipótesis nula,

$H_0: \mu_Z(x_1) = \mu_Z^c(x_1), T_I$ converge en distribución a la variable aleatoria,

$$\xi = \int_0^1 B^2(t) dt, \text{ , Cuando } \min(n_1, n_2) \rightarrow \infty.$$

Donde B es un *proceso Wiener* en $[0, 1]$. La distribución de ξ es la misma como de la variable aleatoria,

$$\sum_{j=0}^{\infty} \frac{4Z_j^2}{[(2j+1)\pi]^2}, \quad (2.16)$$

Donde los Z_j 's son variables aleatorias independientes con distribución normal estándar. T'_1 también converge en distribución a una variable aleatoria ζ' , la cual tiene la misma representación como ζ bajo las mismas condiciones y es independiente de ξ .

- **Teorema 3.**

Bajo la hipótesis nula $H_0: \mu_Z(x_1) = \mu_Z^c(x_1)$ y bajo las condiciones del teorema 2, T_2 converge en distribución a la variable aleatoria,

$\zeta = \frac{B^3(1)}{3} - \int_0^1 B(t)dt$, , Donde B es un *proceso Wiener* en $[0, 1]$. Mas sin embargo,

podemos expresar $\zeta = \left(\frac{N_1^3}{3}\right) - \left(\frac{N_2}{\sqrt{3}}\right)$, donde N_i , $i = 1, 2$ son variables aleatorias

normales con media cero, varianza marginal 1, y covarianza $(N_1, N_2) = \frac{\sqrt{3}}{2}$.

El estadístico T'_2 también converge en distribución a una variable aleatoria ζ' , la cual tiene la misma representación como ζ bajo las mismas condiciones y es independiente de ζ .

- **Teorema 4**

De acuerdo a las condiciones ∂_1 , ∂_3 y ∂_4 y que $n_2/(n_1 + n_2) \rightarrow \theta$, $0 < \theta < 1$, cuando $\min(n_1, n_2) \rightarrow \infty$. Entonces bajo la hipótesis nula $H_0: \mu_Z(x_1) = \mu_Z^c(x_1)$, el estadístico de prueba W_3 converge en distribución a una variable normal estándar.

De acuerdo a *Kulasekera (1995)*, T_1 , T'_1 , T_2 y T'_2 divergen a ∞ a razón n bajo la hipótesis alterna $H_a: \mu_Z(x_1) \neq \mu_Z^c(x_1)$, esto permite a estas estadísticas detectar

diferencias de orden $\frac{1}{\sqrt{n}}$. W_3 diverge a ∞ a una proporción mucho más lenta (\sqrt{n})

bajo la hipótesis alterna $H_a: \mu_Z(x_1) \neq \mu_Z^c(x_1)$.

2.2. ALGUNOS MÉTODOS PARA SELECCIONAR EL PARÁMETRO DE SUAVIZACIÓN EN REGRESIÓN NO-PARAMÉTRICA

Muchas de las soluciones a este problema a lo largo de la literatura consisten en definir algunas medidas en función del estimador de la función de regresión $\mu(\mathbf{x})$ y entonces probar el valor estimado del *parámetro de suavización* el cual es óptimo con respecto a esta medida, además este valor no es observable. Algunos procedimientos estadísticos han sido diseñados para estimar este parámetro.

Presentaremos algunas técnicas para obtener una razonable elección del parámetro λ cuando no conocemos la función de regresión $\mu(\mathbf{x})$.

Consideremos algunas medidas de distancia entre $\mu_\lambda(x)$ y $\mu(x)$:

- Error cuadrático medio(ASE)

$$\text{ASE}((\mu_\lambda(x))) = n^{-1} \sum_{i=1}^n (\mu(x_i) - \mu_\lambda(x_i))^2 \quad (2.17)$$

Uno de los principales problemas de este criterio es que el error cuadrático medio es una variable aleatoria (además depende del diseño de puntos $\{\mathbf{x}_i, i = 1, \dots, n\}$).

- Integral del error cuadrático(ISE)

$$\text{ISE}((\mu_\lambda(x))) = \int_{-\infty}^{\infty} (\mu_\lambda(x) - \mu(x))^2 dx \quad (2.18)$$

Este criterio también es una variable aleatoria. Una manera simple de librarse de esta aleatoriedad consiste en usar el valor esperado de ISE.

- Valor esperado de la Integral del error cuadrático(MISE)

$$\text{MISE}((\mu_\lambda(x))) = E \left(\int_{-\infty}^{\infty} (\mu_\lambda(x) - \mu(x))^2 dx \right) \quad (2.19)$$

Este criterio no es aleatorio, además se puede calcular el MISE para cualquier diseño de puntos $\{\mathbf{x}_i, i = 1, \dots, n\}$.

Algunas propiedades asintóticas de equivalencia entre ISE y MISE indican que ambos métodos deben ser muy aproximados para tamaños de muestra grandes. Algunos cálculos simples permiten descomponer el MISE en dos términos:

$$\text{MISE}((\mu_\lambda(x))) = B((\mu_\lambda(x))) + V((\mu_\lambda(x))),$$

Donde,

$$B((\mu_\lambda(x))) = \int_{-\infty}^{\infty} (E(\mu_\lambda(x) - \mu(x)))^2 dx$$

y

$$V((\mu_\lambda(x))) = \int_{-\infty}^{\infty} \text{Var}(\mu_\lambda(x))^2 dx$$

Para derivar las propiedades asintóticas en este contexto, uno tiene que controlar el parámetro λ para que desaparezca asintóticamente ($\lambda \rightarrow 0$) cuando el tamaño de muestra tiende a ser grande ($n \rightarrow \infty$). Resulta que la integral de la varianza tiende a cero si $n\lambda \rightarrow \infty$ cuando $n \rightarrow \infty$.

- La elección de un valor óptimo para λ implica la minimización de cualquiera de estas medidas de distancia.
- Es conveniente en muchos casos utilizar el criterio del *error cuadrático medio*, el cual puede ser calculado más fácilmente, ver *Härdle (1990)*.

De acuerdo a la definición del *error cuadrático medio*,

$$\begin{aligned} \text{ASE}((\mu_\lambda(x))) &= n^{-1} \sum_{i=1}^n (\mu(x_i) - \mu_\lambda(x_i))^2 \\ &= n^{-1} \sum_{i=1}^n \mu(x_i)^2 + n^{-1} \sum_{i=1}^n \mu_\lambda(x_i)^2 - 2n^{-1} \sum_{i=1}^n \mu(x_i)\mu_\lambda(x_i) \end{aligned}$$

- El primer término es independiente de λ y puede ser ignorado para minimizar ASE $((\boldsymbol{\mu}_\lambda(\mathbf{x}))$.
- El segundo término puede ser directamente calculado de la estimación dada.
- El último término tiene que ser estimado.

Para estimar el tercer término reemplazaremos el valor desconocido $\boldsymbol{\mu}(\mathbf{x})$ por \mathbf{y}_i . Además el primer término no depende de λ , puede ser reemplazado $\boldsymbol{\mu}(\mathbf{x})$ por \mathbf{y}_i en este término. Esto induce a definir el criterio de *restitución en la estimación*,

$$P(\lambda) = n^{-1} \sum_{i=1}^n (y_i - \mu_\lambda(x_i))^2 \quad (2.20)$$

Desafortunadamente $P(\lambda)$ es un estimador sesgado de $ASE((\boldsymbol{\mu}_\lambda(\mathbf{x}))$. Se tiene que $\boldsymbol{\mu}_\lambda(\mathbf{x}) \rightarrow \mathbf{y}_i$ cuando $\lambda \rightarrow 0$ y así $P(\lambda) \rightarrow 0$. La razón para que se presente sesgo en $P(\lambda)$ es que las observaciones \mathbf{y}_i son usadas en $\boldsymbol{\mu}_\lambda(\mathbf{x})$ para predecir sus propios valores \mathbf{y}_i . Por lo anterior, se tiene que al derivar la formula de $P(\lambda)$ encontramos el componente del sesgo de $P(\lambda)$. Este término se puede cancelar si no se incluye la observación $(\mathbf{x}_i, \mathbf{y}_i)$ en la estimación de $\boldsymbol{\mu}(\mathbf{x})$. Así, los cálculos pueden ser basados en la estimación “*dejar-uno-fuera*” $\mu_{\lambda,i}(x_i)$, la suavización Kernel sin $(\mathbf{x}_i, \mathbf{y}_i)$.

$$\text{Donde } \mu_{\lambda,i}(x_i) = \frac{1}{(n-1)\lambda} \sum_{j=1, j \neq i}^n K\left(\frac{x_i - x_j}{\lambda}\right),$$

Esta idea corresponde al concepto de *cross-validation*. Este criterio emplea la estimación “*dejar-uno-fuera*” $\mu_{\lambda,i}(x_i)$ en la formula de la predicción del error cambiando la estimación original. Esto induce a definir el criterio de *Cross-Validación*,

$$CV(\lambda) = n^{-1} \sum_{i=1}^n (y_i - \mu_{\lambda,i}(x_i))^2 \quad (2.21)$$

$CV(\lambda)$ es asintóticamente un estimador insesgado de $ASE((\boldsymbol{\mu}_\lambda(\mathbf{x}))$. Una sucesión de anchos de banda basados en $CV(\lambda)$ es asintóticamente óptimo y produce el mismo grado de convergencia como las otras técnicas.

Otro método muy popular en la selección del ancho de banda λ es el criterio de *validación cruzada de mínimos cuadrados*. La idea del método, es descomponer la expresión de ISE en tres términos,

$$\text{ISE}((\mu_\lambda(x))) = \int_{-\infty}^{\infty} (\mu_\lambda(x) - \mu(x))^2 dx = \int (\mu_\lambda(x))^2 - 2 \int \mu_\lambda(x) \mu(x) dx + \int \mu(x)^2 dx$$

Observe que el término $\int \mu(x)^2 dx$ no depende del parámetro λ , esto no es relevante en el procedimiento de minimización. De otro lado el primer termino es observable (se puede calcular con el diseño de puntos $\{x_i, i = 1, \dots, n\}$), mientras el termino de producto cruzado,

$\int \mu_\lambda(x) \mu(x) dx$, puede ser estimado por $n^{-1} \sum_{i=1}^n \mu_\lambda(x_i)$, o aun mejor, por el estimador insesgado “*dejar-uno-fuera*”

$$n^{-1} \sum_{i=1}^n \mu_{\lambda,i}(x_i)$$

Donde

$$\mu_{\lambda,i}(x_i) = \frac{1}{(n-1)\lambda} \sum_{j=1, j \neq i}^n K\left(\frac{x_i - x_j}{\lambda}\right)$$

Usando este estimador podemos probar para determinar el mejor ancho de banda que minimice ISE $((\mu_\lambda(x)))$. Esto induce a definir el criterio de *validación cruzada de mínimos cuadrados*.

$$\text{CV}((\mu_\lambda(x))) = \int \mu_\lambda(x)^2 dx - \frac{2}{n} \sum_{i=1}^n \mu_{\lambda,i}(x) \quad (2.22)$$

La idea es encontrar el ancho de banda λ_{CV} que minimice la expresión de CV $((\mu_\lambda(x)))$.

Según *Eubank (1999)* es posible utilizar el criterio *del riesgo de predicción* o el criterio de *generalización de Cross-Validación* para seleccionar un valor óptimo para el

parámetro de suavización λ . Recordemos que $P((\mu_\lambda(x))) = n^{-1} \sum_{i=1}^n E(y_i^* - \mu_\lambda(x_i))^2$. Para

utilizar este criterio debemos estimar $E(y_i^* - \mu_\lambda(x_i))^2$. El estimador de $E(y_i^* - \mu_\lambda(x_i))^2$ en este caso es el cuadrado del residual $(y_i - \mu_\lambda(x_i))$. Una buena

estimación para $P(\boldsymbol{\mu}_\lambda(x))$ es el promedio de los residuales al cuadrado por medio de $n^{-1}RSS(\boldsymbol{\mu}_\lambda(x))$, donde $RSS(\boldsymbol{\mu}_\lambda(x)) = \sum_{i=1}^n (y_i - \mu_\lambda(x_i))^2$ es la suma de cuadrados de los residuales asociado con valores ajustados para el estimador. Teniendo en cuenta el estimador Kernel de *Gasser-Müller* (1.7), el vector de valores estimados se puede representar como,

$$\boldsymbol{\mu}_\lambda(x) = (\boldsymbol{\mu}_\lambda(x_1), \dots, \boldsymbol{\mu}_\lambda(x_n))^T = S_\lambda \mathbf{Y}$$

De acuerdo a esto, encontramos una matriz S_λ para cada λ , donde S_λ es asumida a ser simétrica y positiva semidefinida. S_λ tiene como elemento típico,

$$\lambda^{-1} \int_{z_{i-1}}^{z_i} K(\lambda^{-1}(x_i - z)) dz$$

Para encontrar una representación aceptable de $n^{-1}RSS(\boldsymbol{\mu}_\lambda(x))$ debemos calcular su valor esperado, es decir $E(RSS(\boldsymbol{\mu}_\lambda(x)))$. Note que se puede expresar,

$$RSS(\boldsymbol{\mu}_\lambda(x)) = (\mathbf{Y} - \boldsymbol{\mu}_\lambda(x))^T (\mathbf{Y} - \boldsymbol{\mu}_\lambda(x)) = \mathbf{Y}^T (\mathbf{I} - S_\lambda)^2 \mathbf{Y}$$

Ahora,

$$\begin{aligned} E(RSS(\boldsymbol{\mu}_\lambda(x))) &= \boldsymbol{\mu}(x)^T (\mathbf{I} - S_\lambda)^2 \boldsymbol{\mu}(x) + \sigma^2 \text{tr}[(\mathbf{I} - S_\lambda)^2] \\ &= \boldsymbol{\mu}(x)^T (\mathbf{I} - S_\lambda)^2 \boldsymbol{\mu}(x) + n\sigma^2 + \sigma^2 \text{tr}[S_\lambda^2] - 2\sigma^2 \text{tr}[S_\lambda] \end{aligned}$$

Sabemos que,

$$\begin{aligned} P(\boldsymbol{\mu}_\lambda(x)) &= \sigma^2 + R(\boldsymbol{\mu}_\lambda(x)) \\ &= \sigma^2 + n^{-1} \sum_{i=1}^n E(\mu(x) - \mu_\lambda(x_i))^2 \\ &= \sigma^2 + n^{-1} E(\boldsymbol{\mu}(x) - \boldsymbol{\mu}_\lambda(x))^T (\boldsymbol{\mu}(x) - \boldsymbol{\mu}_\lambda(x)) \\ &= \sigma^2 + n^{-1} \boldsymbol{\mu}(x)^T (\mathbf{I} - S_\lambda)^2 \boldsymbol{\mu}(x) + n^{-1} \sigma^2 \text{tr}[(\mathbf{I} - S_\lambda)^2] \end{aligned}$$

Comparando $E(\text{RSS}(\boldsymbol{\mu}_\lambda(x)))$ con $P(\boldsymbol{\mu}_\lambda(x))$ se puede observar que $n^{-1} \text{RSS}(\boldsymbol{\mu}_\lambda(x))$ es un estimador sesgado de $P(\boldsymbol{\mu}_\lambda(x))$. El componente del sesgo es igual a $-2n^{-1}\sigma^2 \text{tr}[S_\lambda]$.

Un estimador insesgado de $P(\boldsymbol{\mu}_\lambda(x))$ es dado por,

$$\hat{P}(\boldsymbol{\mu}_\lambda(x)) = n^{-1} \text{RSS}(\boldsymbol{\mu}_\lambda(x)) + 2n^{-1}\sigma^2 \text{tr}[S_\lambda].$$

En muchas aplicaciones prácticas no conocemos σ^2 , por lo que, usualmente debe ser estimado. Luego,

$$\hat{P}(\boldsymbol{\mu}_\lambda(x)) = n^{-1} \sum_{i=1}^n (y_i - \mu_\lambda(x_i))^2 + 2n^{-1}\hat{\sigma}^2 \text{tr}[S_\lambda]. \quad (2.23)$$

Otro método útil para seleccionar λ es la *Generalización de Cros-Validación* GCV ($\boldsymbol{\mu}_\lambda(x)$). GCV ($\boldsymbol{\mu}_\lambda(x)$) tiende a ser un estimador insesgado de $P(\boldsymbol{\mu}_\lambda(x))$. Asumiendo que $\text{tr}[S_\lambda] < n$, el criterio es definido por

$$\text{GCV}(\boldsymbol{\mu}_\lambda(x)) = n^{-1} \text{RSS}(\boldsymbol{\mu}_\lambda(x)) / (n^{-1} \text{tr}[\mathbf{I} - S_\lambda])^2 \quad (2.24)$$

Según *Eubank (1999)*, el criterio se puede definir por

$$\text{GCV}(\boldsymbol{\mu}_\lambda(x)) = n^{-1} \sum_{i=1}^n (y_i - \mu_{\lambda,i}(x_i))^2 \left\{ \frac{1 - s_i}{n^{-1} \text{tr}[\mathbf{I} - S_\lambda]} \right\}^2 \quad (2.25)$$

Donde s_i es el *i-esimo* elemento de la diagonal de la matriz S_λ .

De acuerdo a *Gasser y Müller (1979)*, la *Generalización de Cros-Validación* es definida por

$$\text{GCV}(\boldsymbol{\mu}_\lambda(x)) = n^{-1} \sum_{i=1}^n (Y_i - \mu_\lambda(x_i))^2 / (1 - n^{-1} \text{tr}[S_\lambda])^2 \quad (2.26)$$

La idea es encontrar el ancho de banda λ_{GCV} o λ_P que minimice la expresión de GCV ($\boldsymbol{\mu}_\lambda(x)$) o $\hat{P}(\boldsymbol{\mu}_\lambda(x))$. Para evaluar cada uno de estos criterios debemos de tener en cuenta que $\boldsymbol{\mu}_\lambda(x)$ es el estimador de la función de regresión $\boldsymbol{\mu}(x)$, donde $\boldsymbol{\mu}_\lambda(x)$ es un estimador lineal de la forma dada en (1.2).

Una de las estrategias utilizadas para escoger un valor óptimo del parámetro de suavización λ es elegir uno o más de estos criterios y usar el valor $\lambda_{\text{Mínimo}}$ como un estimador en la minimización de $L(\mu_\lambda(x))$ o $R(\mu_\lambda(x))$. La elección de cual criterio usar depende en muchos casos del conocimiento o no del valor de σ^2 , si se puede realizar una fácil estimación y los recursos computacionales disponibles.

Si σ^2 es desconocido y no es disponible una razonable estimación, $GCV((\mu_\lambda(x)))$ o $CV((\mu_\lambda(x)))$ pueden ser usados, además estos criterios no requieren estimación de la varianza del error. Cuando todos los estimadores del *Riesgo* son minimizados por algún valor de λ esto confirma la creencia que una buena elección para λ ha sido encontrada. En este contexto, consideramos que λ es una variable determinística o no aleatoria.

Así, $R(\mu_\lambda(x))$ y $P(\mu_\lambda(x))$ son funciones no aleatorias de la variable no aleatoria λ , mientras $L(\mu_\lambda(x))$, $RSS(\mu_\lambda(x))$, $\hat{P}(\mu_\lambda(x))$, $GCV((\mu_\lambda(x)))$ y $CV((\mu_\lambda(x)))$ son funciones aleatorias de una variable determinística λ .

2.3 SELECCIÓN DEL PARÁMETRO DE SUAVIZACIÓN EN LA COMPARACIÓN DE DOS CURVAS DE REGRESIÓN

La comparación de funciones de regresión para dos o más poblaciones es en la actualidad un problema en discusión. Métodos no-paramétricos para tales comparaciones son utilizados en algunos problemas debido a dificultades tales como la especificación del modelo, la aplicación de procedimientos paramétricos. Usando métodos no-paramétricos para realizar un análisis inferencial en la comparación de curvas de regresión se utiliza un parámetro de suavización que es especificado por el investigador.

2.3.1 Selección del valor óptimo para el parámetro de suavización que maximiza la potencia de la prueba El funcionamiento de una prueba respecto a la potencia puede ser significativamente afectada por el parámetro de suavización. En lo siguiente, examinaremos el efecto del parámetro de suavización en la potencia de

pruebas de hipótesis no-paramétricas. En nuestro caso particular, pruebas de hipótesis para comparar dos curvas de regresión.

En el caso $r = 2$ el modelo es $Y = \mu(\mathbf{x}_1, \mathbf{x}_2) + \varepsilon$, en el problema de selección de variables se asume que \mathbf{x}_1 juega un papel importante en el modelo, luego se pretende comprobar si \mathbf{x}_2 se debe de incluir en el modelo, es decir, queremos probar si \mathbf{x}_2 es una variable significativa en la función $\mu(\mathbf{x})$.

Recordemos que para el caso $r = 2$, las hipótesis de prueba son:

$$H_0: \mu(\mathbf{x}_1, \mathbf{x}_2) = \mu(\mathbf{x}_1)$$

$$H_a: \mu(\mathbf{x}_1, \mathbf{x}_2) \neq \mu(\mathbf{x}_1)$$

Para realizar este tipo de pruebas se utiliza el método de *Cuasi-Residuales* propuesto por *Kulasekera* (1995). Después de aplicar el método de *Cuasi-Residuales*, las hipótesis de prueba quedan de la siguiente forma:

$$H_0: \mu_Z(\mathbf{x}_1) = \mu_Z^c(\mathbf{x}_1)$$

$$H_a: \mu_Z(\mathbf{x}_1) \neq \mu_Z^c(\mathbf{x}_1)$$

Cuando observamos datos en la forma $\{\mathbf{x}_{1i}, \mathbf{y}_i\}, i = 1, \dots, n_1$ y $\{\mathbf{x}_{1l}, \mathbf{y}_l\}, l = 1, \dots, n_2$

Donde,

$$Y_Z = \mu_Z(\mathbf{x}_1) + \varepsilon, \text{ para aquellos valores de } \mathbf{x}_2 \in Z$$

$$Y_Z^c = \mu_Z^c(\mathbf{x}_1) + \varepsilon, \text{ para aquellos valores de } \mathbf{x}_2 \in Z^c$$

Donde Z^c es el complemento del conjunto Z , ε es una variable aleatoria no observable para los dos grupos. Además, se asume que en cada grupo los errores están idénticamente distribuidos con media 0 y $\text{Var}(\varepsilon) = \sigma^2$. Para este tipo de pruebas las dos funciones $\mu_Z(\mathbf{x}_1)$ y $\mu_Z^c(\mathbf{x}_1)$ cumplen algunas condiciones de suavización. El dominio de la covariable $\mathbf{X} \in [0, 1]$. El diseño de puntos para las dos muestras y los tamaños de muestra pueden ser diferentes.

Kulasekera (1995) propone tres tipos de pruebas W_1 , W_2 y W_3 para un diseño común de puntos y tamaños de muestra iguales. Las tres pruebas utilizan estimadores lineales, tales como estimadores Kernel para las funciones de regresión desconocidas, el cual influye en la potencia de la prueba por medio del parámetro de suavización del estimador lineal. Aunque la significancia de la prueba en la selección de un parámetro de suavización óptimo al realizar una prueba de comparación de curvas se ha observado, no se han aplicado métodos para seleccionar el valor apropiado del parámetro de suavización basado en los datos maximizando la potencia de la prueba comparada con una alternativa dada.

Para las pruebas W_1 y W_2 en *Kulasekera* y *Wang* (1997), la potencia depende

$$\text{principalmente del parámetro } \Delta = \int_0^1 \left[\int_0^t ((\mu_Z - \mu_{Z^c})(u)) du \right]^2 dt. \quad (2.27)$$

Proponemos seleccionar el parámetro de suavización para las pruebas W_1 y W_2 en *Kulasekera* y *Wang* (1997) maximizando los estimadores de la correspondiente función de Δ , así optimizando la potencia de una prueba para un conjunto de datos.

Los siguientes supuestos pueden ser necesarios,

- ∂_1 . Las funciones μ_Z y μ_{Z^c} tienen dos derivadas continuas.
- ∂_2 . El parámetro de suavización λ usado en la estimación de μ_Z y μ_{Z^c} satisface $\min(n_1, n_2)\lambda \rightarrow \infty$ y $\min(n_1, n_2)\lambda^4 \rightarrow 0$ cuando $\min(n_1, n_2) \rightarrow \infty$.
- ∂_3 . Los diseños de puntos $\{\mathbf{x}_{1i}, i = 1, \dots, n_1\}$ y $\{\mathbf{x}_{1l}, l = 1, \dots, n_2\}$ se hacen densos en $[-1, 1]$ cuando $\min(n_1, n_2) \rightarrow \infty$.
- ∂_4 . El parámetro de suavización λ usado en la estimación de μ_Z y μ_{Z^c} satisface $\min(n_1, n_2)\lambda \rightarrow \infty$ y $\min(n_1, n_2)\lambda^5 \rightarrow 0$ cuando $\min(n_1, n_2) \rightarrow \infty$.
- ∂_5 . La función Kernel es una función de densidad de probabilidad conocida simétrica en 0, definida en $[-1, 1]$.

Este procedimiento de selección es asintóticamente óptimo en el sentido que la razón del parámetro de suavización que maximiza la función Δ con respecto al método de selección del parámetro de suavización basado en los datos tiende a 1 en probabilidad cuando el tamaño de muestra incrementa. Sin embargo, el parámetro de suavización basado en los datos puede influir en el tamaño de la prueba, y la diferencia del tamaño real de la prueba depende en mayor parte de la variabilidad de las respuestas y el tamaño de la muestra. Se puede mostrar que estas diferencias disminuyen rápidamente cuando se incrementa el tamaño de muestra. En la construcción de las pruebas W_1 y W_2 para comprobar hipótesis, en todos los casos las estadísticas T_i (2.11) y (2.12) divergen a ∞ a razón de n bajo la hipótesis alterna.

Por ejemplo, en el caso $\mathbf{r} = \mathbf{2}$ el modelo se define por $Y = \mu(\mathbf{x}_1, \mathbf{x}_2) + \varepsilon$, entonces para una diferencia $h = \mu_Z(X_1) - \mu_Z^c(X_1)$, observamos que

$$E(T_1) \approx \frac{1}{n_1^2} \sum_{j=1}^{n_1} \left[\sum_{i=1}^j h_{1i} \right]^2 + \frac{\kappa_2 \sigma^2}{n_2 \lambda_2} \quad (2.28)$$

Donde,

$$h_{1i} = \mu_z(x_{1i}) - \sum_{l=1}^{n_2} \mu_{z^c}(x_{1l}) w_{2l}(x_{1i}),$$

y

$$E(T'_1) \approx \frac{1}{n_2^2} \sum_{j=1}^{n_2} \left[\sum_{l=1}^j h_{2l} \right]^2 + \frac{\kappa_2 \sigma^2}{n_1 \lambda_1} \quad (2.29)$$

Donde,

$$h_{2l} = \mu_{z^c}(x_{1l}) - \sum_{i=1}^{n_1} \mu_z(x_{1i}) w_{1i}(x_{1l}),$$

Observemos que λ_1 , λ_2 y $w_{1i}, i = 1, \dots, n_1, w_{2l}, l = 1, \dots, n_2$ son los anchos de banda y los pesos en la función Kernel para estimar $\mu_Z(x)$ y $\mu_Z^c(x)$ usando $\mathbf{y}_i, i = 1, \dots, n_1$ y $\mathbf{y}_l, l = 1, \dots, n_2$. En ambos casos la divergencia de los estadísticos de prueba bajo la hipótesis alterna depende del primer término del lado derecho de cada expresión. En vista de estas aproximaciones, proponemos la elección de los anchos de banda λ_1 y λ_2 para maximizar:

$$\Delta(\lambda_1, \lambda_2) = \frac{1}{n_1^3} \sum_{l=1}^{n_1} \left[\sum_{i=1}^l h_i \right]^2 \quad (2.30)$$

Donde,

$$h_i = h(x_{1l}) = \frac{1}{2} \left[\sum_{l=1}^{n_1} (\mu_z(x_{1l}) - \mu_{z^c}(x_{1l})) w_{1l}(x_{1l}) + \sum_{j=1}^{n_2} (\mu_z(x_{1j}) - \mu_{z^c}(x_{1j})) w_{2j}(x_{1l}) \right]$$

En esta formulación \mathbf{x}_{1l} son los datos que pertenecen al conjunto Z , \mathbf{x}_{1j} son los datos que pertenecen al conjunto Z^c , los pesos $w_{1l}, l = 1, \dots, n_1, w_{2j}, j = 1, \dots, n_2$ son funciones de $\mathbf{x}_{1l}, \mathbf{x}_{1j}$ y λ_1, λ_2 respectivamente. También se requiere que la suma de los pesos sea igual a 1. Sea $\lambda = (\lambda_1, \lambda_2)$ los valores del parámetro de suavización que maximiza la función Δ .

Definimos la versión muestral de Δ como:

$$\hat{\Delta}(\lambda_1, \lambda_2) = \frac{1}{n_1^3} \sum_{l=1}^{n_1} \left[\sum_{i=1}^l \hat{h}_i \right]^2 \quad (2.31)$$

Donde,

$$\hat{h}_i = \frac{1}{2} \left[\sum_{l=1}^{n_1} \left(\hat{\mu}_z(x_{1l}) - \hat{\mu}_{z^c}(x_{1l}) \right) w_{1l}(x_{1l}) + \sum_{j=1}^{n_2} \left(\hat{\mu}_z(x_{1j}) - \hat{\mu}_{z^c}(x_{1j}) \right) w_{2j}(x_{1l}) \right] \quad (2.32)$$

Donde $\hat{\mu}_z(x)$ y $\hat{\mu}_{z^c}(x)$ son los estimadores lineales de *Gasser y Müller* (1979) dados en (1.7). $\hat{\mu}_z(x_{1l})$ y $\hat{\mu}_{z^c}(x_{1l})$ son valores estimados de las funciones de regresión correspondientes, donde X_{1l} son los valores de X_1 que pertenecen al conjunto Z . $\hat{\mu}_z(x_{1j})$ y $\hat{\mu}_{z^c}(x_{1j})$ son valores estimados de las funciones de regresión correspondientes, donde X_{1j} son los valores de X_1 que pertenecen al conjunto Z^c . Las estimaciones del valor óptimo de los anchos de banda $\hat{\lambda}_1$ y $\hat{\lambda}_2$ se obtienen maximizando la función $\hat{\Delta}$.

Sea $\hat{\lambda} = (\hat{\lambda}_1, \hat{\lambda}_2)$ los valores estimados del parámetro de suavización que maximiza la función $\hat{\Delta}$.

Teorema 4

Si los supuestos $\partial_3 - \partial_5$ de la página 55 se satisfacen. Entonces,

$\hat{\lambda} / \lambda \rightarrow 1$ en probabilidad cuando $\min(n_1, n_2) \rightarrow \infty$. El anterior Teorema es demostrado en *Kulasekera y Wang (1997)*.

En el caso $r > 2$ el modelo es $Y = \mu(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_r) + \varepsilon$, en el problema de selección de variables se asume que $\mu(\mathbf{x})$ es una función propia de $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{r-1}$. Se desea probar si \mathbf{x}_r debe incluirse en el modelo, es decir, queremos probar si \mathbf{x}_r es una variable significativa en la función $\mu(\mathbf{x})$. Por ejemplo, si $r = 3$ el modelo sería $Y = \mu(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) + \varepsilon$.

Entonces las hipótesis quedan de la siguiente forma:

$$H_0: \mu(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) = \mu(\mathbf{x}_1, \mathbf{x}_2)$$

$$H_a: \mu(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) \neq \mu(\mathbf{x}_1, \mathbf{x}_2)$$

Se aplica el método de *Cuasi-Residuales* propuesto por *Kulasekera (1995)* para resolver esta prueba. Después de aplicar el método de *Cuasi-Residuales*, las hipótesis de prueba quedan de la siguiente forma:

$$H_0: \mu_Z(\mathbf{U}) = \mu_Z^c(\mathbf{U})$$

$$H_a: \mu_Z(\mathbf{U}) \neq \mu_Z^c(\mathbf{U})$$

Para alguna función μ propia de \mathbf{U} . Donde $\mu_Z(\mathbf{U})$ y $\mu_Z^c(\mathbf{U})$ son las curvas de regresión de la variable de respuesta \mathbf{Y} contra \mathbf{U} en los dos subconjuntos.

De acuerdo a la Definición anterior la variable \mathbf{U} es una combinación lineal de la forma:

$$\mathbf{U} = \mathbf{c}_1 \mathbf{x}_1 + \mathbf{c}_2 \mathbf{x}_2.$$

Ahora, definimos la versión muestral de $\hat{\Delta}$ como:

$$\hat{\Delta}(\lambda_1, \lambda_2) = \frac{1}{n_1^3} \sum_{l=1}^{n_1} \left[\sum_{i=1}^l \hat{h}_i \right]^2 \quad (2.33)$$

Donde,

$$\hat{h}_i = \frac{1}{2} \left[\sum_{l=1}^{n_1} \left(\hat{\mu}_z(U_l) - \hat{\mu}_{z^c}(U_l) \right) w_{1l}(U_l) + \sum_{j=1}^{n_2} \left(\hat{\mu}_z(U_j) - \hat{\mu}_{z^c}(U_j) \right) w_{2j}(U_l) \right] \quad (2.34)$$

En esta formulación $\mathbf{U}_l = \mathbf{c}_1 \mathbf{x}_{1l} + \mathbf{c}_2 \mathbf{x}_{2l}$, donde \mathbf{x}_{1l} y \mathbf{x}_{2l} pertenecen al conjunto Z ; $\mathbf{U}_j = \mathbf{c}_1 \mathbf{x}_{1j} + \mathbf{c}_2 \mathbf{x}_{2j}$, donde \mathbf{x}_{1j} y \mathbf{x}_{2j} pertenecen al conjunto Z^c ; los pesos $w_{1l}, l=1, \dots, n_1, w_{2j}, j=1, \dots, n_2$ son funciones de $\mathbf{U}_l, \mathbf{U}_j$ y λ_1, λ_2 respectivamente. También se requiere que la suma de los pesos sea igual a 1. Donde $\hat{\mu}_z(\mathbf{U})$ y $\hat{\mu}_{z^c}(\mathbf{U})$ son los estimadores lineales de Gasser y Müller (1979) dados en (1.7). Las estimaciones de los anchos de banda $\hat{\lambda}_1$ y $\hat{\lambda}_2$ se obtienen maximizando la función $\hat{\Delta}$.

Para nuestro problema en cuestión, se debe aclarar que $\hat{\lambda}_1 = \hat{\lambda}_2$ (usamos el mismo valor del parámetro de suavización λ para estimar las funciones de regresión $\hat{\mu}_z$ y $\hat{\mu}_{z^c}$). Tendríamos así una especie de *Corolario* que se deduce del Teorema

2.3.1.1 y que garantiza que el uso del Teorema es válido para nuestro caso. Entonces, la

$$\text{función } \hat{\Delta}(\lambda) = \hat{\Delta}(\lambda_1, \lambda_2) = \frac{1}{n_1^3} \sum_{l=1}^{n_1} \left[\sum_{i=1}^l \hat{h}_i \right]^2. \quad (2.35)$$

3. RESULTADOS DE ALGUNAS SIMULACIONES

Todas las simulaciones presentadas en este capítulo fueron realizadas por el autor de este proyecto de investigación siguiendo los lineamientos establecidos en el capítulo 4.

Inicialmente mostraremos resultados de algunas simulaciones respecto al procedimiento propuesto para estimar los valores de la variable de respuesta Y .

3.1 ESTIMACIÓN DE LA FUNCIÓN DE REGRESIÓN $\mu(x)$

El método propuesto es el estimador lineal de *Gasser y Müller (1979)* (1.7)

$$\mu_{\lambda}(x) = \sum_{i=1}^n (\lambda^{-1} \int_{S_{i-1}}^{S_i} K(\lambda^{-1}(x-s)) ds) y_i$$

con $S_0 = 0$, $S_n = 1$ y $S_i = (x_{i+1} + x_i) / 2$, $i = 1, \dots, n-1$. La función de pesos utilizada es el *Kernel cuadrático* $K(u) = \frac{3}{4}(1-u^2)I_{[-1,1]}(u)$ la cual es una *función Kernel de segundo orden*.

Resolviendo la expresión del lado derecho en la estimación de $\mu(x)$ dado en (1.7) tenemos:

$$\mu_{\lambda}(x) = y_1 \lambda^{-1} \int_{S_0}^{S_1} K(\lambda^{-1}(x-s)) ds + y_2 \lambda^{-1} \int_{S_1}^{S_2} K(\lambda^{-1}(x-s)) ds + \dots + y_n \lambda^{-1} \int_{S_{n-1}}^{S_n} K(\lambda^{-1}(x-s)) ds$$

Sea $u = \frac{x-s}{\lambda}$, entonces $du = -\frac{ds}{\lambda}$, ahora tenemos que $-\lambda du = ds$

$$\mu_{\lambda}(x) = y_1 \lambda^{-1} \int_{S_0}^{S_1} K(u)(-\lambda du) + y_2 \lambda^{-1} \int_{S_1}^{S_2} K(u)(-\lambda du) + \dots + y_n \lambda^{-1} \int_{S_{n-1}}^{S_n} K(u)(-\lambda du)$$

$$\mu_{\lambda}(x) = y_1 \lambda^{-1} \int_{S_0}^{S_1} (-\lambda) K(u) du + y_2 \lambda^{-1} \int_{S_1}^{S_2} (-\lambda) K(u) du + \dots + y_n \lambda^{-1} \int_{S_{n-1}}^{S_n} (-\lambda) K(u) du$$

$$\mu_{\lambda}(x) = (-y_1 \int_{S_0}^{S_1} K(u) du) + (-y_2 \int_{S_1}^{S_2} K(u) du) + \dots + (-y_n \int_{S_{n-1}}^{S_n} K(u) du)$$

$$\mu_{\lambda}(x) = (-y_1 \int_{s_0}^{s_1} \frac{3}{4}(1-u^2)du) + (-y_2 \int_{s_1}^{s_2} \frac{3}{4}(1-u^2)du) + \dots + (-y_n \int_{s_{n-1}}^{s_n} \frac{3}{4}(1-u^2)du)$$

$$\mu_{\lambda}(x) = (-\frac{3}{4} y_1 \int_{s_0}^{s_1} (1-u^2)du) + (-\frac{3}{4} y_2 \int_{s_1}^{s_2} (1-u^2)du) + \dots + (-\frac{3}{4} y_n \int_{s_{n-1}}^{s_n} (1-u^2)du)$$

$$\mu_{\lambda}(x) = y_1(-\frac{3}{4} \int_{s_0}^{s_1} (1-u^2)du) + y_2(-\frac{3}{4} \int_{s_1}^{s_2} (1-u^2)du) + \dots + y_n(-\frac{3}{4} \int_{s_{n-1}}^{s_n} (1-u^2)du)$$

$$\mu_{\lambda}(x) = y_1 \left(-\frac{3}{4} \left(u - \frac{u^3}{3} \right) \Big|_{S_0}^{S_1} \right) + y_2 \left(-\frac{3}{4} \left(u - \frac{u^3}{3} \right) \Big|_{S_1}^{S_2} \right) + \dots + y_n \left(-\frac{3}{4} \left(u - \frac{u^3}{3} \right) \Big|_{S_{n-1}}^{S_n} \right)$$

Como resultado final, vemos que se puede estimar $\mu(\mathbf{x})$ para un valor \mathbf{x}_i y un ancho de banda (λ) dado de la siguiente manera:

$$\mu_{\lambda}(x) = y_1 \left(-\frac{3}{4} \left(\left(\frac{x-s}{\lambda} \right) - \frac{1}{3} \left(\frac{x-s}{\lambda} \right)^3 \right) \Big|_{S_0}^{S_1} \right) + \dots + y_n \left(-\frac{3}{4} \left(\left(\frac{x-s}{\lambda} \right) - \frac{1}{3} \left(\frac{x-s}{\lambda} \right)^3 \right) \Big|_{S_{n-1}}^{S_n} \right)$$

Se puede observar que la estimación de la función de regresión $\mu(\mathbf{x})$ tiene en cuenta todas las observaciones de la variable de respuesta \mathbf{Y} .

3.1.1 Algunos aspectos computacionales de la simulación. Para estimar la función de regresión $\mu(\mathbf{x})$, seleccionamos muestras de tamaño 100. En cada muestra los datos corresponden a un diseño de puntos equidistantes con valores en el intervalo [0,1], los valores de la variable de respuesta \mathbf{Y} fueron generados aplicando una función $f(x)$

agregando un error aleatorio. Para la distribución del error consideramos dos tipos de errores Normales en la simulación:

Errores tipo 1: $\{0.3N\}$ y

Errores tipo 2: $\{0.5N\}$

Donde $\text{Var}(\varepsilon_1) = 0.3^2$, $\text{Var}(\varepsilon_2) = 0.5^2$ y N es una variable aleatoria Normal estándar.

En la simulación usamos un conjunto de valores para el ancho de banda en el rango 0,01 hasta 0,9. La función de regresión usada en la simulación es:

$f(x) = \sin(4x)e^x + 3$. En nuestro caso $f(x) = \mu(\mathbf{x})$.

3.1.2 Resultados de la simulación. La curva ajustada usando el estimador lineal de *Gasser y Müller (1979)* se muestra en las figuras 1, 2 y 3.

Figura 1. Un ejemplo simulado para una muestra de tamaño 100. Un ancho de banda de 0,01 con errores tipo 1. Los valores estimados de Y(estimación de $\mu(x)$) aparecen en una línea continua. Los valores de la función $\mu(x)$ aparecen en círculos consecutivos. Los valores observados de Y aparecen en círculos dispersos.

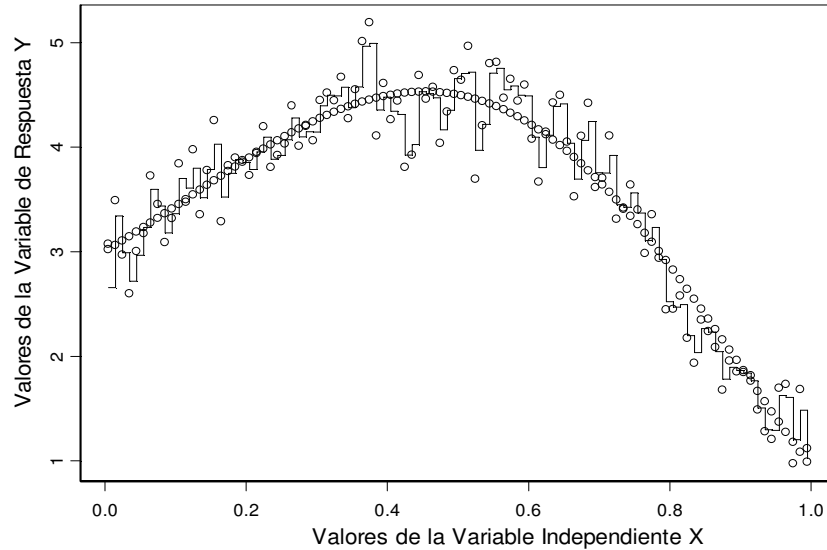


Figura 2. Un ejemplo simulado para una muestra de tamaño 100. Un ancho de banda de 0,05 con errores tipo 1. Los valores estimados de Y(estimación de $\mu(x)$) aparecen en una línea continua. Los valores de la función $\mu(x)$ aparecen en círculos consecutivos. Los valores observados de Y aparecen en círculos dispersos.

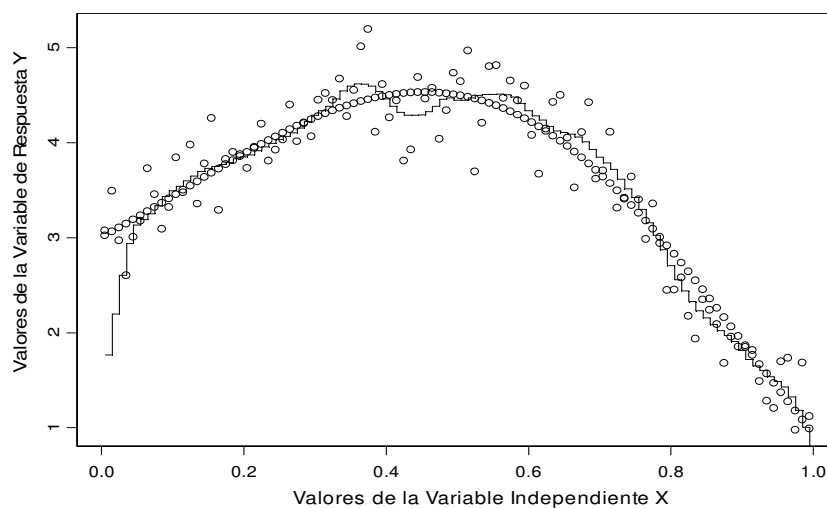
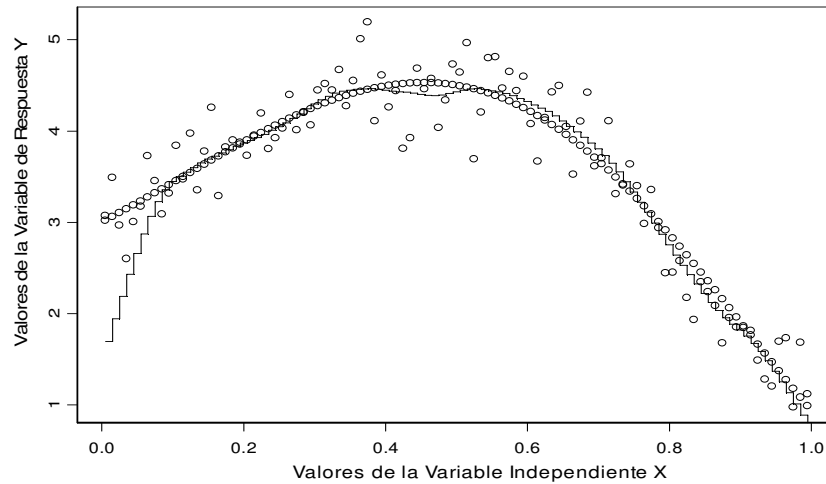


Figura 3. Un ejemplo simulado para una muestra de tamaño 100. Un ancho de banda de 0,1 con errores tipo 1. Los valores estimados de Y(estimación de $\mu(x)$) aparecen en una línea continua. Los valores de la función $\mu(x)$ aparecen en círculos consecutivos. Los valores observados de Y aparecen en círculos dispersos.



Se observa que a medida que aumenta el ancho de banda (λ) entre 0,01 y 0,1 se encontró una buena estimación de la función $\mu(x)$, es decir, existe una mejor suavización de la curva de regresión ajustada. También es notorio un alto sesgo de estimación en valores de X cercanos a cero, esto es típico en estimadores Kernel.

3.2 SELECCIÓN DE UN VALOR ÓPTIMO DEL PARÁMETRO SUAVIZACIÓN PARA ESTIMAR LA FUNCIÓN DE REGRESIÓN $\mu(x)$

Se puede encontrar una mejor precisión de la curva estimada de $\mu(x)$ seleccionando un valor apropiado del parámetro de suavización λ . En general, aplicaremos algunos procedimientos para seleccionar λ que busca optimizar la medida del error cuadrático para la curva de regresión y sus derivadas. Presentaremos algunas técnicas para obtener una razonable elección del parámetro λ cuando no conocemos la función de regresión $\mu(x)$.

3.2.1 Algunos aspectos computacionales de la simulación. Por medio de una Simulación seleccionamos muestras de tamaño 100. En cada muestra los datos corresponden a un diseño de puntos equidistantes con valores en el intervalo [0,1], los valores de la variable de respuesta Y fueron generados aplicando una función $f(x) = \text{sen}(4x)e^x + 3$ agregando un error aleatorio. Para estimar la función de regresión $\mu(x)$ se utilizó el estimador lineal de *Gasser y Müller* (1979) (1.7)

Una de las técnicas utilizadas en la simulación para determinar el ancho de banda óptimo fue *Generalización del criterio de validación cruzada GCV* ($\mu_\lambda(x)$) según *Gasser y Müller* (1988). Utilizamos un conjunto de valores para el ancho de banda en el rango 0,01 hasta 0,9. Es decir, se calculó $GCV(\mu_\lambda(x))$ utilizando muestras de tamaño 100 para cada valor del ancho de banda (λ). En la simulación se determinó el valor mínimo de $GCV(\mu_\lambda(x))$ teniendo en cuenta el valor del ancho de banda (λ) asociado. El procedimiento se repitió 1000 veces y se contó el número de veces en que cada ancho de banda utilizado en la simulación produjo el $GCV(\mu_\lambda(x))$ mínimo, considerando que el mínimo valor de GCV está asociado con el ancho de banda más adecuado para estimar la función de regresión $\mu(x)$. Ver tablas 1 y 2.

3.2.2 Resultados de la simulación.

Figura 4. Un ejemplo simulado para una muestra de tamaño 100. Con errores tipo 1. Los círculos representan los valores *del criterio de validación cruzada Generalizada* según *Gasser y Müller* (1988).

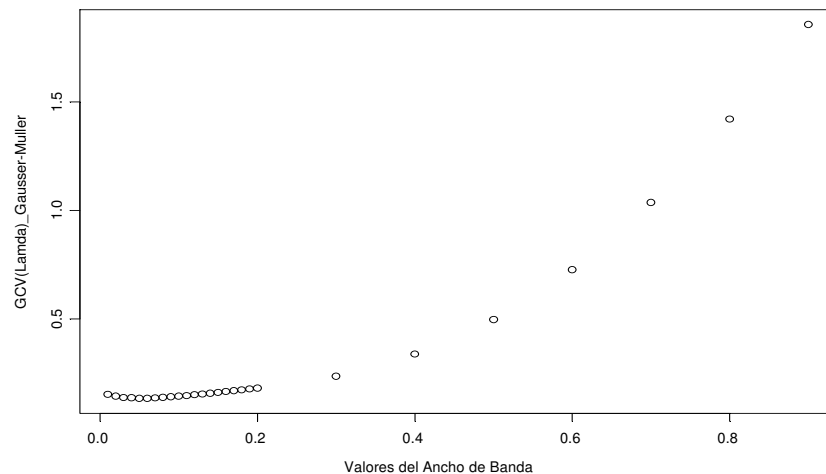
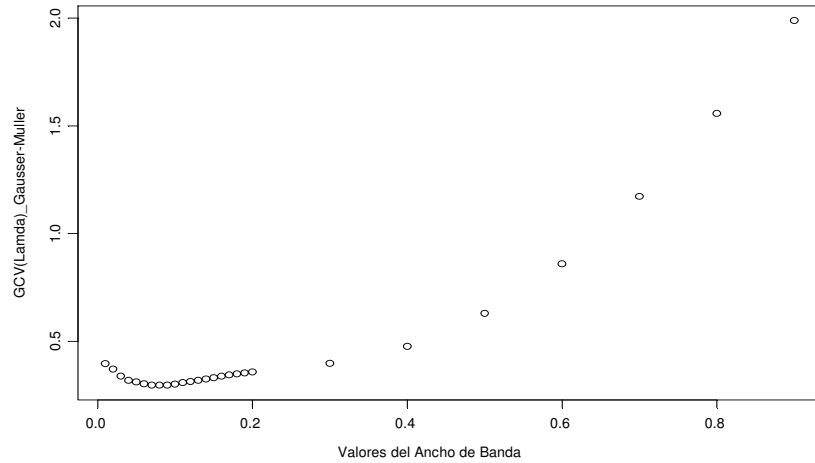


Figura 5. Un ejemplo simulado para una muestra de tamaño 100. Con errores tipo 2. Los círculos representan los valores *del criterio de validación cruzada Generalizada* según *Gasser y Müller (1988)*.



En la *figura 4* podemos observar que el valor *del criterio de validación cruzada Generalizada* aumenta con mayor rapidez para valores del ancho de banda entre 0,2 y 0,9. Valores del ancho de banda entre 0 y 0,1 permiten encontrar un valor mínimo para el *criterio de validación cruzada Generalizada* según *Gasser y Müller (1988)*.

Comparando los resultados encontrados en la *figura 5* con respecto a la *figura 4* observamos que el comportamiento *del criterio de validación cruzada Generalizada* es similar cuando se emplean errores tipo 1 y 2. Valores del ancho de banda entre 0 y 0,1 permiten encontrar un valor mínimo para el *criterio de validación cruzada Generalizada* según *Gasser y Müller (1988)*.

Tabla 1. (1000 repeticiones) Con errores normales tipo 1. Tamaño de muestra igual 100. *Criterio de validación cruzada Generalizada* según Gasser y Müller (1988).

Ancho de banda(λ)	Número de veces que λ produjo el valor mínimo de $GCV(\mu_\lambda(x))$
0,01	32
0,02	95
0,03	245
0,04	243
0,05	217
0,06	112
0,07	40
0,08	13
0,09	3

De acuerdo a los resultados encontrados en la *tabla 1* se puede observar que el valor mínimo del *Criterio de validación cruzada Generalizada* según Gasser y Müller (1988) se obtiene para valores del ancho de banda entre 0,03 y 0,05.

Tabla 2. (1000 repeticiones) Con errores normales tipo 2. Tamaño de muestra igual 100. *Criterio de validación cruzada Generalizada* según Gasser y Müller (1988).

Ancho de banda(λ)	Número de veces que λ produjo el valor mínimo de $GCV(\mu_\lambda(x))$
0,01	6
0,02	25
0,03	67
0,04	125
0,05	156
0,06	139
0,07	152
0,08	126
0,09	80
0,10	45
0,11	37
0,12	17
0,13	14
0,14	5
0,15	5
0,16	1

En la *tabla 2* se puede observar que el valor mínimo del *Criterio de validación cruzada Generalizada* según Gasser y Müller (1988) se encuentra para posibles valores del ancho de banda entre 0,04 y 0,08.

3.3 SELECCIÓN DE UN VALOR ÓPTIMO DEL PARÁMETRO DE SUAVIZACIÓN PARA MAXIMIZAR LA POTENCIA DE UNA PRUEBA

Ahora, aplicaremos un estudio de simulación para seleccionar el parámetro de suavización apropiado para maximizar la correspondiente función de $\hat{\Delta}(\lambda) = \hat{\Delta}(\lambda_1, \lambda_2)$ (2.35), aplicando las pruebas W_1 y W_2 en *Kulasekera y Wang* (1997). Buscamos optimizar la potencia de una prueba para un conjunto de datos.

3.3.1 Algunos aspectos computacionales de la simulación. Por medio de una simulación seleccionamos dos muestras n_1 y n_2 de tamaño 50. En cada muestra los datos corresponden a un diseño de puntos equidistantes con valores en el intervalo [0,1], los valores de la variable de respuesta Y fueron generados aplicando una o dos funciones $f(x)$ y $g(x)$ agregando un error aleatorio. Utilizamos un conjunto de valores para el ancho de banda en el rango 0,01 hasta 0,9. Es decir, se calculó

$$\hat{\Delta}(\lambda) = \hat{\Delta}(\lambda_1, \lambda_2) = \frac{1}{n_1^3} \sum_{l=1}^{n_1} \left[\sum_{i=1}^l \hat{h}_i \right]^2$$

para cada valor del ancho de banda (λ). En nuestro caso de interés recordemos que $\lambda_1 = \lambda_2$. En la simulación se determinó el valor máximo de $\hat{\Delta}(\lambda)$ teniendo en cuenta el valor del ancho de banda (λ) asociado. El procedimiento se repitió 1000 veces y se contó el número de veces en que cada ancho de banda utilizado en la simulación produjo el valor máximo de la función $\hat{\Delta}(\lambda)$, considerando que el valor máximo de la función $\hat{\Delta}(\lambda)$ está asociado con el ancho de banda más adecuado para maximizar la potencia de una prueba.

3.3.2 Resultados de la simulación

- **Caso donde se aplico una función $f(x)$, diferentes diseños de puntos en las muestras n_1 y n_2 .** En la generación de los valores de Y se aplico solamente la función $f(x) = \text{sen}(4x)e^x + 3$ para cada una de las muestras n_1 y n_2 de tamaño 50. El diseño de puntos en cada muestra es diferente. Para estimar la función de regresión $\mu(x)$ se utilizó el estimador lineal de *Gasser y Müller (1979)* (1.7). Ver tablas 3 y 4.

Figura 6. Un ejemplo simulado para dos muestras n_1 y n_2 de tamaño 50. Diseño de puntos diferentes para cada muestra. Se utilizo una sola función $f(x)$ para generar los valores observados de Y . Errores tipo 1. Los círculos representan los valores de la función $\hat{\Delta}(\lambda)$.

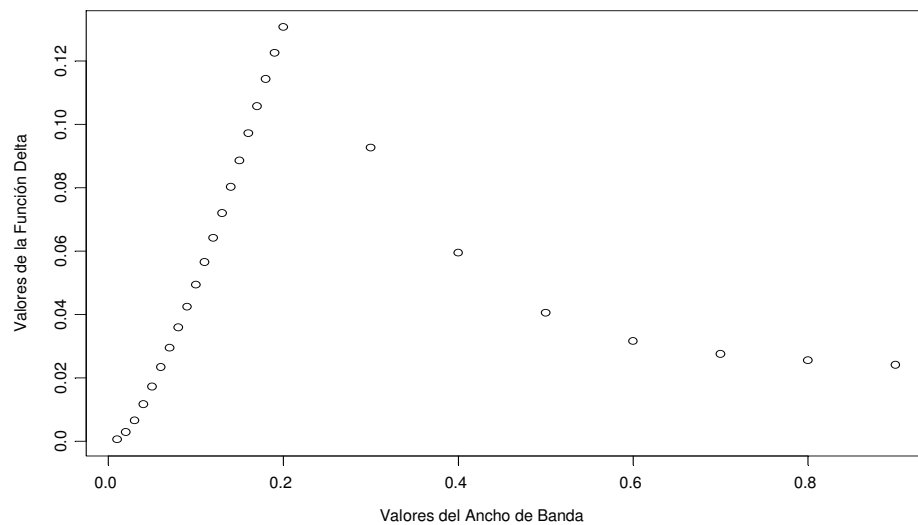
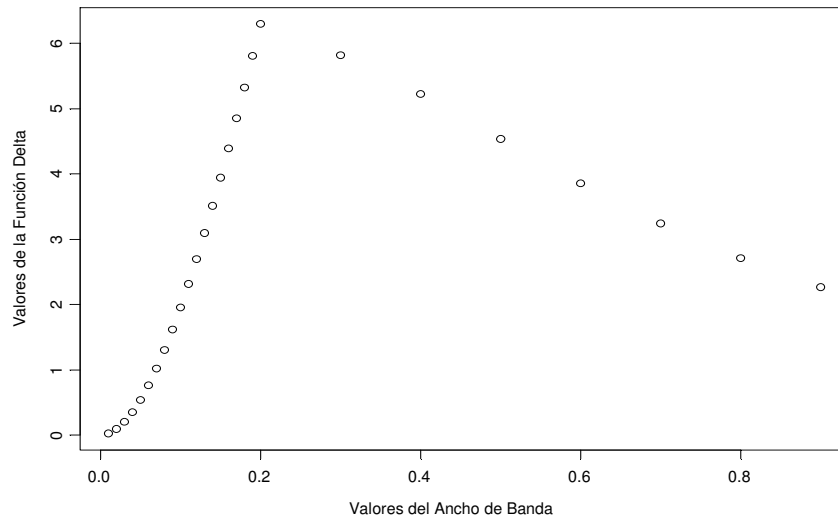


Figura 7. Un ejemplo simulado para dos muestras n_1 y n_2 de tamaño 50. Diseño de puntos diferentes para cada muestra. Se utilizo una sola función $f(x)$ para generar los valores observados de Y . Errores tipo 2. Los círculos representan los valores *de la función* $\hat{\Delta}(\lambda)$.



En la *figura 6* podemos observar que el valor *de la función* $\hat{\Delta}(\lambda)$ aumenta con mayor rapidez para valores del ancho de banda entre 0 y 0,2. Para valores del ancho de banda entre 0,2 y 0,9 decrece casi de igual manera. Valores del ancho de banda muy cercanos a 0,2 permiten encontrar un valor máximo *de la función* $\hat{\Delta}(\lambda)$.

Comparando los resultados encontrados en la *figura 7* con respecto a la *figura 6* observamos que el comportamiento *de la función* $\hat{\Delta}(\lambda)$ es muy similar cuando se emplean errores tipo 1 en la generación de los valores de la variable Y . Valores del ancho de banda muy cercanos a 0,2 permiten encontrar un valor máximo *de la función* $\hat{\Delta}(\lambda)$.

Tabla 3. Potencia empírica (1000 repeticiones) Con errores normales tipo 1. Se generaron dos muestras n_1 y n_2 de tamaño 50. Diseño de puntos diferente para cada muestra. Se utilizó la función $f(x)$ para generar los valores observados de Y .

Ancho de banda(λ)	Número de veces que λ produjo el valor máximo de la función $\hat{\Delta}(\lambda)$
0,12	2
0,15	3
0,16	3
0,17	2
0,18	4
0,19	5
0,20	680
0,30	90
0,40	72
0,50	42
0,60	34
0,70	40
0,80	11
0,90	12

De acuerdo a los resultados encontrados en la *tabla 3* se puede observar que el valor máximo de la función $\hat{\Delta}(\lambda)$ se encuentra asociado al valor 0,2 del ancho de banda.

Tabla 4. Potencia empírica (1000 repeticiones) Con errores normales tipo 2. Se generaron dos muestras n_1 y n_2 de tamaño 50. Diseño de puntos diferentes para cada muestra. Se utilizó la función $f(x)$ para generar los valores observados de Y .

Ancho de banda(λ)	Número de veces que λ produjo el valor máximo de la función $\hat{\Delta}(\lambda)$
0,10	1
0,11	1
0,12	2
0,14	1
0,16	3
0,17	1
0,18	2
0,19	1
0,20	700
0,30	96
0,40	78
0,50	36
0,60	20
0,70	19
0,80	14
0,90	25

Comparando los resultados encontrados en las *tablas 3 y 4* se observa que el comportamiento del valor máximo de la función $\hat{\Delta}(\lambda_1, \lambda_2)$ es similar cuando se emplean errores tipo 1 y 2 con respecto al valor del ancho de banda asociado.

- **Caso donde se aplicaron dos funciones $f(x)$ y $g(x)$, igual diseño de puntos en las muestras n_1 y n_2 .** En la generación de los valores de la variable de respuesta

Y se aplico la función $f(x)$ agregando un error aleatorio para los datos de la muestra n_1 y una función $g(x)$ agregando un error aleatorio para los datos de la muestra n_2 , cada una de tamaño 50. El diseño de puntos en cada muestra es igual. Para estimar la función de regresión $\mu(x)$ en cada muestra se utilizó el estimador lineal de *Gasser y Müller* (1979) (1.7)

Las funciones de regresión usadas en la simulación son:

$$f(x) = \text{sen}(4x)e^x + 3$$

$$g(x) = \cos(4x). \text{ Ver tablas 5 y 6.}$$

Figura 8. Un ejemplo simulado para dos muestras n_1 y n_2 de tamaño 50. Diseño de puntos igual para cada muestra. Se utilizaron dos funciones $f(x)$ y $g(x)$ para generar los valores observados de Y . Errores tipo 1. Los círculos representan los valores de la función $\hat{\Delta}(\lambda)$.

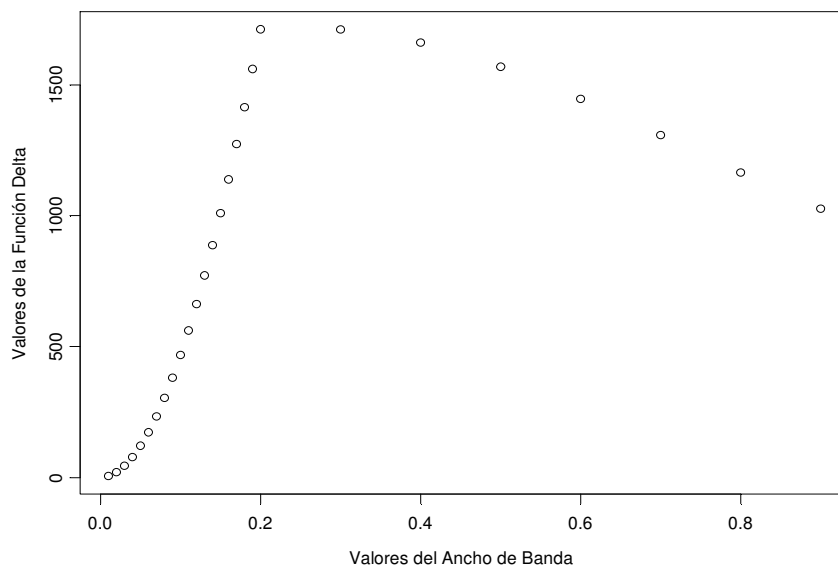
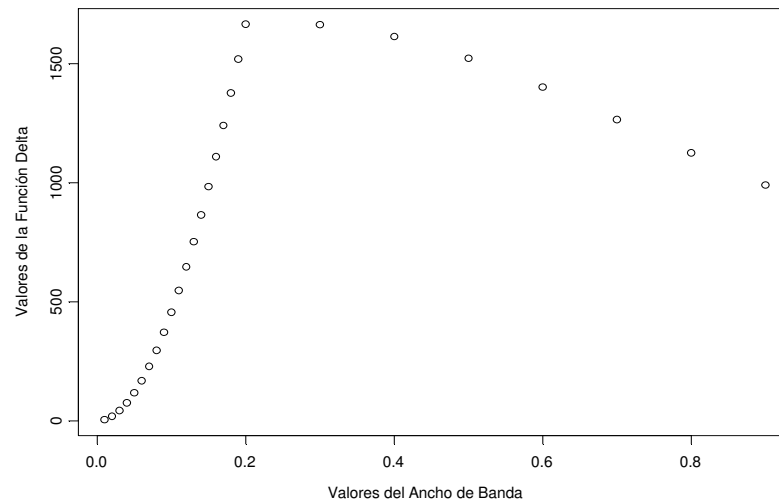


Figura 9. Un ejemplo simulado para dos muestras n_1 y n_2 de tamaño 50. Diseño de puntos igual para cada muestra. Se utilizaron dos funciones $f(x)$ y $g(x)$ para generar los valores observados de Y . Errores tipo 2. Los círculos representan los valores de la función $\hat{\Delta}(\lambda)$.



En la *figura 8* podemos observar que el valor de la función $\hat{\Delta}(\lambda)$ se incrementa a medida que el ancho de banda aumenta entre 0 y 0,2. Para valores del ancho de banda entre 0,3 y 0,9 decrece la función. Valores del ancho de banda entre 0,2 y 0,3 permiten encontrar un valor máximo de la función $\hat{\Delta}(\lambda)$.

Se puede observar en la *figura 9* que el comportamiento de la función $\hat{\Delta}(\lambda)$ es muy similar cuando se utilizan errores tipo 1 y 2 en la generación de la variable de respuesta Y .

Tabla 5. Potencia empírica (1000 repeticiones) Con errores normales tipo 1. Se generaron dos muestras n_1 y n_2 de tamaño 50. Diseño de puntos igual para cada muestra. Se utilizaron dos funciones $f(x)$ y $g(x)$ para generar los valores observados de Y .

Ancho de banda(λ)	Número de veces que λ produjo el valor máximo de la función $\hat{\Delta}(\lambda)$
0,20	854
0,30	146

De acuerdo a los resultados encontrados en la *tabla 5* se puede observar que el valor máximo de la función $\hat{\Delta}(\lambda)$ se encuentra asociado al valor 0,2 del ancho de banda.

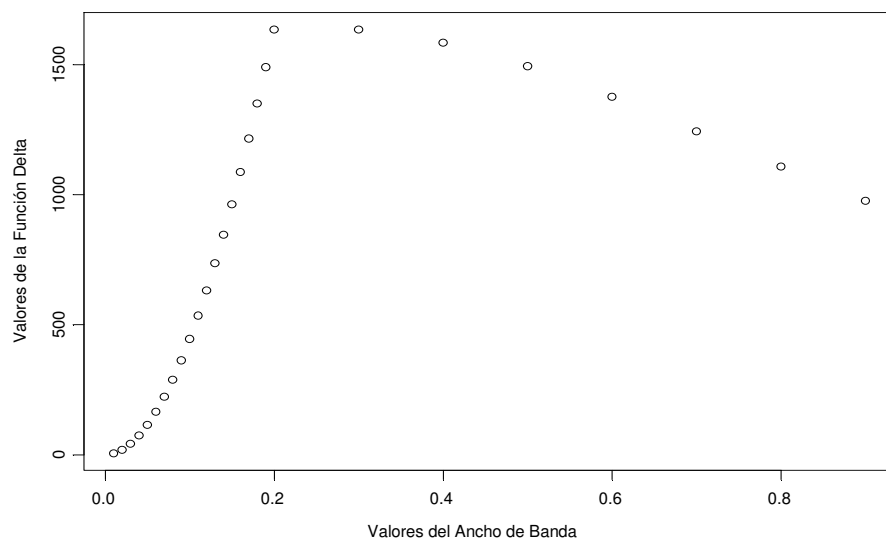
Tabla 6. Potencia empírica (1000 repeticiones) Con errores normales tipo 2. Se generaron dos muestras n_1 y n_2 de tamaño 50. Diseño de puntos igual para cada muestra. Se utilizaron dos funciones $f(x)$ y $g(x)$ para generar los valores observados de Y .

Ancho de banda(λ)	Número de veces que λ produjo el valor máximo de la función $\hat{\Delta}(\lambda)$
0,20	721
0,30	279

Comparando los resultados encontrados en las *tablas 5 y 6* se observa que el comportamiento del valor máximo de la función $\hat{\Delta}(\lambda)$ es similar cuando se emplean errores tipo 1 y 2 con respecto al valor del ancho de banda asociado.

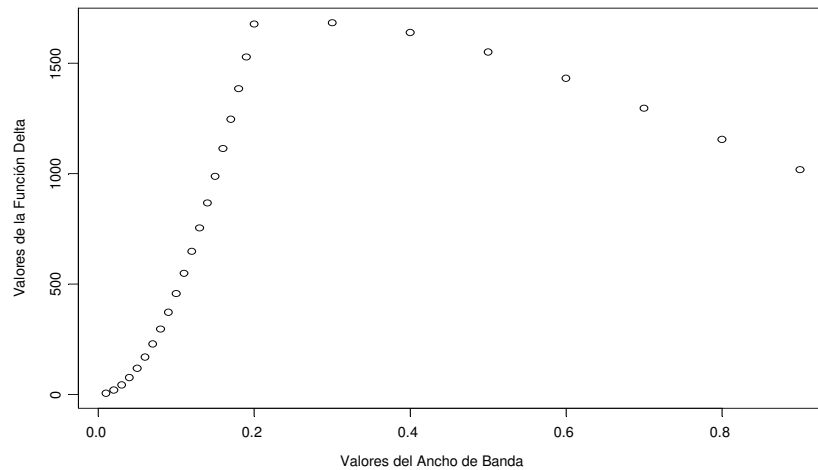
- **Caso donde se aplicaron dos funciones $f(x)$ y $g(x)$, diferentes diseños de puntos en las muestras n_1 y n_2 .** En la generación de los valores de la variable de respuesta Y se aplicó la función $f(x) = \sin(4x)e^x + 3$ para los datos de la muestra n_1 y la función $g(x) = \cos(4x)$ para los datos de la muestra n_2 , cada una de tamaño 50. El diseño de puntos en cada muestra es diferente. Para estimar la función de regresión $\mu(x)$ en cada muestra se utilizó el estimador lineal de *Gasser y Müller* (1979) (1.7). Ver tablas 7 y 8.

Figura 10. Un ejemplo simulado para dos muestras n_1 y n_2 de tamaño 50. Diseño de puntos diferente para cada muestra. Se utilizaron dos funciones $f(x)$ y $g(x)$ para generar los valores observados de Y . Errores tipo 1. Los círculos representan los valores de la función $\hat{\Delta}(\lambda)$.



Se observa en la *figura 10* que el comportamiento de la función $\hat{\Delta}(\lambda)$ es muy similar cuando se utiliza un diseño de puntos igual para cada muestra con errores tipo 1 (*figura 8*) en la generación de la variable de respuesta Y .

Figura 11. Un ejemplo simulado para dos muestras n_1 y n_2 de tamaño 50. Diseño de puntos diferente para cada muestra. Se utilizaron dos funciones $f(x)$ y $g(x)$ para generar los valores observados de Y . Errores tipo 2. Los círculos representan los valores de la función $\hat{\Delta}(\lambda)$.



Se observa en la *figura 11* que el comportamiento de la *función* $\hat{\Delta}(\lambda)$ es muy similar cuando se utiliza un diseño de puntos igual para cada muestra con errores tipo 2 (*figura 9*) en la generación de la variable de respuesta Y .

Tabla 7. Potencia empírica (1000 repeticiones) Con errores normales tipo 1. Se generaron dos muestras n_1 y n_2 de tamaño 50. Diseño de puntos diferente para cada muestra. Se utilizaron dos funciones $f(x)$ y $g(x)$ para generar los valores observados de Y .

Ancho de banda(λ)	Número de veces que λ produjo el valor máximo de la función $\hat{\Delta}(\lambda)$
0,20	663
0,30	337

De acuerdo a los resultados encontrados en la *tabla 7* se puede observar que el valor máximo de la función $\hat{\Delta}(\lambda)$ se encuentra asociado al valor 0,2 del ancho de banda.

Tabla 8. Potencia empírica (1000 repeticiones) Con errores normales tipo 2. Se generaron dos muestras n_1 y n_2 de tamaño 50. Diseño de puntos diferente para cada muestra. Se utilizaron dos funciones $f(x)$ y $g(x)$ para generar los valores observados de Y .

Ancho de banda(λ)	Número de veces que λ produjo el valor máximo de la función $\hat{\Delta}(\lambda)$
0,20	610
0,30	390

Comparando los resultados encontrados en las *tablas 7 y 8* se observa que el comportamiento del valor máximo de la función $\hat{\Delta}(\lambda)$ es similar cuando se emplean errores tipo 1 y 2 con respecto al valor del ancho de banda asociado.

3.4 SELECCIÓN DE UN VALOR ÓPTIMO DEL PARÁMETRO DE SUAVIZACIÓN EN EL PROBLEMA DE SELECCIÓN DE VARIABLES EN REGRESIÓN NO-PARAMETRICA

Finalmente, después de encontrar algunos resultados con respecto al valor óptimo del parámetro de suavización en la estimación de la función de regresión $\mu(x)$ y en la búsqueda para maximizar la función $\hat{\Delta}(\lambda) = \hat{\Delta}(\lambda_1, \lambda_2)$ (2.35) teniendo en cuenta solamente una variable independiente X_i , centraremos nuestro estudio para seleccionar un valor óptimo del parámetro de suavización en el problema de selección de variables en el marco no-paramétrico propuesto por *Olaya (2000)*.

Ahora, aplicaremos un estudio de simulación para seleccionar el parámetro de suavización óptimo en el problema de selección de variables. Recordemos que el método propuesto por *Olaya* (2000) permite seleccionar variables significativas en el modelo.

En el proceso de selección de variables se aplica una prueba formal de hipótesis de comparación de dos curvas de regresión en cada etapa del proceso de selección, donde las variables seleccionadas tienen un orden de importancia.

Cuando realizamos la prueba de comparación de dos curvas de regresión seleccionamos un valor apropiado del parámetro de suavización el cual permite maximizar la correspondiente función de $\hat{\Delta}(\lambda) = \hat{\Delta}(\lambda_1, \lambda_2)$ (2.35), aplicando las pruebas W_1 y W_2 en *Kulasekera y Wang* (1997). Buscamos optimizar la potencia de la prueba para un conjunto de datos. Aplicando este método encontraremos un valor óptimo del parámetro de suavización.

3.4.1 Caso donde se tiene en el modelo dos variables explicativas independientes X_1 y X_2 . Supondremos que el modelo de regresión involucra dos variables explicativas independientes X_1 , X_2 y la variable de respuesta Y . En el caso $r = 2$ el modelo es, $Y = \mu(X_1, X_2) + \varepsilon$, asumiremos que $\mu(X_1, X_2) = X_1 + X_2$, por lo que $Y = X_1 + X_2 + \varepsilon$.

Este es un modelo lineal que podría ser estimado por medio de un análisis de regresión paramétrico. Para estimar la función de regresión $\mu(\mathbf{x})$ se utilizó el estimador lineal de *Gasser y Müller* (1979) (1.7)

Diseño de puntos aleatorios

♦ **Algunos aspectos computacionales de la simulación** Por medio de una simulación generamos una muestra aleatoria de tamaño 100. Para las variables X_1 y X_2 se generaron aleatoriamente 10 valores, el diseño de puntos fue generado usando una distribución uniforme con valores en el intervalo $[0,1]$. Los valores de la variable de

respuesta Y fueron generados aplicando la función $f(x) = X_1 + X_2$ agregando un error aleatorio. Se quiere probar si X_2 es una variable significativa en la función $f(x)$.

Se divide el conjunto de datos (X_1, X_2) acorde a los valores de X_2 . Sea $0,5$ algún valor de X_2 y definimos los conjuntos $Z = \{X_2: X_2 \leq 0,5\}$, $Z^c = \{X_2: X_2 > 0,5\}$.

Ahora, tenemos dos subconjuntos de observaciones, es decir, dos muestras n_1 y n_2 de tamaño 50. Donde $\{X_{1j}, Y_j\}$, $j = 1, \dots, n_1$ para aquellos valores de $x_2 \in Z$ y $\{X_{1m}, Y_m\}$, $m = 1, \dots, n_2$ para aquellos valores de $x_2 \in Z^c$.

En estos dos subconjuntos de puntos calculamos los estimadores $\mu_z(\hat{X}_1)$ y $\mu_{z^c}(\hat{X}_1)$ de $\mu_z(X_1)$ y $\mu_{z^c}(X_1)$ respectivamente, teniendo en cuenta algunos valores del ancho de banda. Después de obtener las estimaciones anteriores procedemos a evaluar la potencia de la prueba de hipótesis. Utilizamos un conjunto de valores para el ancho de banda en el rango 0,01 hasta 0,9. Es decir, se

calculó $\hat{\Delta}(\lambda) = \hat{\Delta}(\lambda_1, \lambda_2) = \frac{1}{n_1^3} \sum_{l=1}^{n_1} \left[\sum_{i=1}^l \hat{h}_i \right]^2$ utilizando dos muestras n_1 y n_2 de tamaño

50 para cada valor del ancho de banda (λ) . En nuestro caso de interés recordemos que $\lambda_1 = \lambda_2$. En la simulación se determinó el valor máximo de $\hat{\Delta}(\lambda)$ teniendo en cuenta el valor del ancho de banda (λ) asociado. El procedimiento se repitió 1000 veces y se contó el número de veces en que cada ancho de banda utilizado en la simulación produjo el valor máximo de la función $\hat{\Delta}(\lambda)$, considerando que el valor máximo de la función $\hat{\Delta}(\lambda)$ está asociado con el ancho de banda más adecuado para maximizar la potencia de una prueba. Ver tablas 9 y 10.

- Resultados de la simulación

Figura 12. Un ejemplo simulado para dos submuestras n_1 y n_2 de tamaño 50. Diseño de puntos aleatorios de distribución uniforme con valores en el intervalo $[0,1]$. Se utilizó una función $f(x)$ para generar los valores observados de Y . Errores tipo 1. Los círculos representan los valores *de la función* $\hat{\Delta}(\lambda)$.

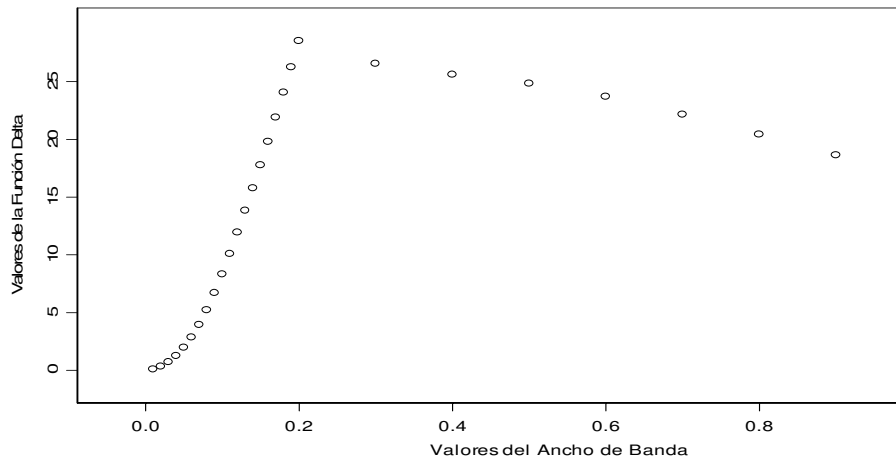
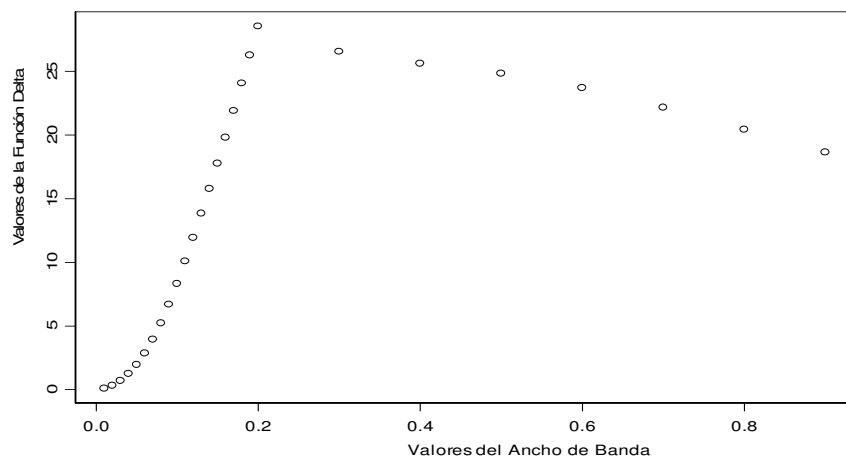


Figura 13. Un ejemplo simulado para dos submuestras n_1 y n_2 de tamaño 50. Diseño de puntos aleatorios de distribución uniforme con valores en el intervalo $[0,1]$. Se utilizó una función $f(x)$ para generar los valores observados de Y . Errores tipo 2. Los círculos representan los valores *de la función* $\hat{\Delta}(\lambda)$.



En la *figura 12* podemos observar que el valor *de la función* $\hat{\Delta}(\lambda)$ se incrementa a medida que el ancho de banda aumenta entre 0 y 0,2. Para valores del ancho de banda

entre 0,3 y 0,9 decrece la función. Valores del ancho de banda entre 0,2 y 0,3 permiten encontrar un valor máximo de la función $\hat{\Delta}(\lambda)$.

Se observa en la *figura 13* que el comportamiento de la función $\hat{\Delta}(\lambda)$ es muy similar cuando se utilizan errores tipo 1 y 2 en la generación de la variable de respuesta **Y**.

Tabla 9. Potencia empírica (1000 repeticiones) Con errores normales tipo 1. Se generaron dos submuestras n_1 y n_2 de tamaño 50. Diseño de puntos aleatorios de distribución uniforme con valores en el intervalo [0,1]. Se utilizó una función $f(x)$ para generar los valores observados de **Y**.

Ancho de banda(λ)	Número de veces que λ produjo el valor máximo de la función $\hat{\Delta}(\lambda)$
0,20	741
0,30	79
0,40	45
0,5	62
0,60	45
0,70	20
0,80	8

De acuerdo a los resultados encontrados en la *tabla 9* se puede observar que el valor máximo de la función $\hat{\Delta}(\lambda)$ se encuentra asociado al valor 0,2 del ancho de banda.

Tabla 10. Potencia empírica (1000 repeticiones) Con errores normales tipo 2. Se generaron dos submuestras n_1 y n_2 de tamaño 50. Diseño de puntos aleatorios de distribución uniforme con valores en el intervalo [0,1]. Se utilizó una función $f(x)$ para generar los valores observados de Y .

Ancho de banda(λ)	Número de veces que λ produjo el valor máximo de la función $\hat{\Delta}(\lambda)$
0,10	1
0,15	1
0,18	2
0,19	1
0,20	560
0,30	163
0,40	124
0,50	60
0,60	46
0,70	24
0,80	16
0,90	2

Comparando los resultados encontrados en las *tablas 9 y 10* se observa que el comportamiento del valor máximo de la función $\hat{\Delta}(\lambda)$ es similar cuando se emplean errores tipo 1 y 2 con respecto al valor del ancho de banda asociado. Es decir, el valor del ancho de banda correspondiente es 0,2.

◆ **Diseño de puntos no aleatorios**

- **Algunos aspectos computacionales de la simulación** Por medio de una Simulación generamos una muestra de tamaño 100. Para las variables X_1 y X_2 se generaron 10 valores no aleatorios, los datos corresponden a un diseño de puntos equidistantes con valores en el intervalo [0,1]. Los valores de la variable de respuesta Y fueron generados aplicando la función $f(x) = X_1 + X_2$ agregando un

error aleatorio. Se quiere probar si X_2 es una variable significativa en la función $f(x)$. Ver tablas 11 y 12.

- **Resultados de la simulación** La *figura 14* permite observar que el valor *de la función* $\hat{\Delta}(\lambda)$ se incrementa a medida que el ancho de banda aumenta entre 0 y 0,2. Para valores del ancho de banda entre 0,3 y 0,9 decrece la función. Valores del ancho de banda entre 0,2 y 0,3 permiten encontrar un valor máximo *de la función* $\hat{\Delta}(\lambda)$.

Figura 14. Un ejemplo simulado para dos submuestras n_1 y n_2 de tamaño 50. Diseño de puntos no aleatorios de distribución uniforme con valores en el intervalo $[0,1]$. Se utilizó una función $f(x)$ para generar los valores observados de Y . Errores tipo 1. Los círculos representan los valores *de la función* $\hat{\Delta}(\lambda)$.

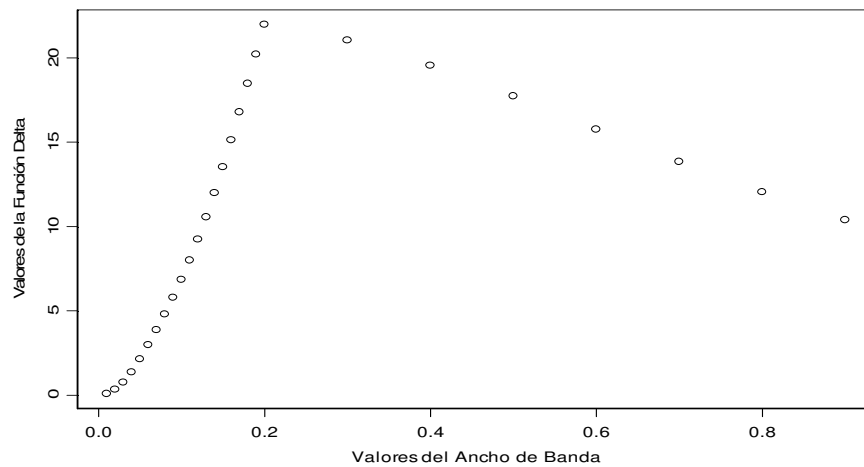
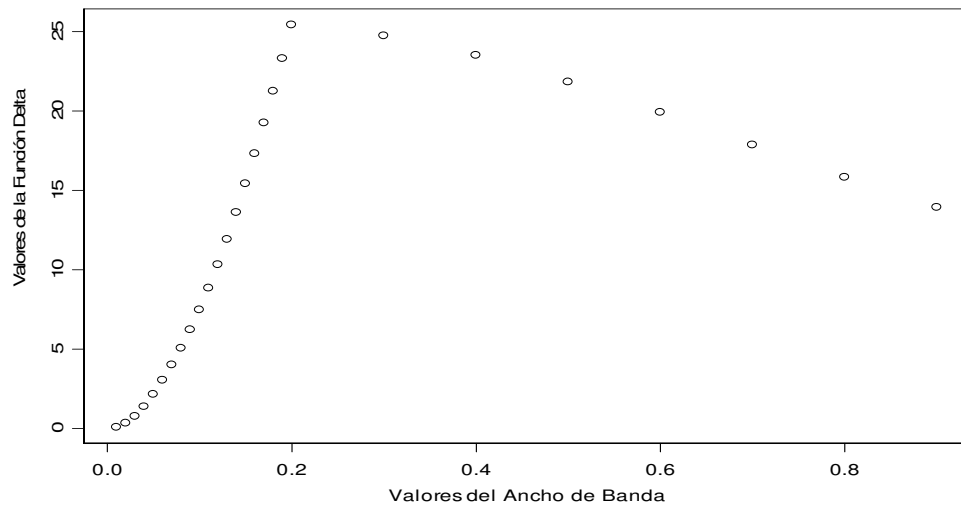


Figura 15. Un ejemplo simulado para dos submuestras n_1 y n_2 de tamaño 50. Diseño de puntos no aleatorios de distribución uniforme con valores en el intervalo $[0,1]$. Se utilizó una función $f(x)$ para generar los valores observados de Y . Errores tipo 2. Los círculos representan los valores de la función $\hat{\Delta}(\lambda)$.



Se observa en la *figura 15* que el comportamiento de la *función* $\hat{\Delta}(\lambda)$ es muy similar cuando se utilizan errores tipo 1 y 2 en la generación de la variable de respuesta Y .

Tabla 11. Potencia empírica (1000 repeticiones) Con errores normales tipo 1. Se generaron dos submuestras n_1 y n_2 de tamaño 50. Diseño de puntos no aleatorios de distribución uniforme con valores en el intervalo $[0,1]$. Se utilizó una función $f(x)$ para generar los valores observados de Y .

Ancho de banda(λ)	Número de veces que λ produjo el valor máximo de la función $\hat{\Delta}(\lambda)$
0.20	772
0.30	193
0.40	27
0.50	6
0.60	1
0.70	1

Teniendo en cuenta los resultados encontrados en la *tabla 11* se puede observar que el valor máximo de la función $\hat{\Delta}(\lambda)$ se encuentra asociado al valor 0,2 del ancho de banda.

Tabla 12. Potencia empírica (1000 repeticiones) Con errores normales tipo 2. Se generaron dos submuestras n_1 y n_2 de tamaño 50. Diseño de puntos no aleatorios de distribución uniforme con valores en el intervalo [0,1]. Se utilizó una función $f(x)$ para generar los valores observados de Y .

Ancho de banda(λ)	Número de veces que λ produjo el valor máximo de la función $\hat{\Delta}(\lambda)$
0,18	1
0,19	1
0,20	690
0,30	183
0,40	81
0,50	27
0,60	8
0,70	5
0,80	4

Al comparar los resultados encontrados en las *tablas 11 y 12* se observa que el comportamiento del valor máximo de la función $\hat{\Delta}(\lambda)$ es similar cuando se emplean errores tipo 1 y 2 con respecto al valor del ancho de banda asociado.

3.4.2 Caso donde se tiene en el modelo tres variables explicativas independientes X_1 , X_2 y X_3 . Supondremos que el modelo de regresión involucra tres variables explicativas independientes X_1 , X_2 , X_3 y la variable de respuesta Y . En el caso $r = 3$ el modelo es, $Y = \mu(X_1, X_2, X_3) + \varepsilon$, asumiremos que

$\mu(X_1, X_2, X_3) = X_1 + X_2 + X_3$, por lo que $Y = X_1 + X_2 + X_3 + \varepsilon$. Para estimar la función de regresión $\mu(\mathbf{x})$ se utilizó el estimador lineal de *Gasser y Müller* (1979) (1.7)

♦ **Diseño de puntos aleatorios**

- **Algunos aspectos computacionales de la simulación** Por medio de una simulación generamos aleatoriamente 6 valores para las variables X_1, X_2 y X_3 , así generamos una muestra aleatoria de tamaño $n = 6*6*6 = 216$. El diseño de puntos fue generado usando una distribución uniforme con valores en el intervalo $[0,1]$. Los valores de la variable de respuesta Y fueron generados aplicando la función $f(\mathbf{x}) = X_1 + X_2 + X_3$ agregando un error aleatorio. Se quiere probar si X_3 es una variable significativa en la función $f(\mathbf{x})$.

Se divide el conjunto de datos (X_1, X_2, X_3) acorde a los valores de X_3 . Sea $0,5$ algún valor de X_3 y definimos los conjuntos $Z = \{X_3: X_3 \leq 0,5\}$, $Z^C = \{X_3: X_3 > 0,5\}$.

Ahora, tenemos dos subconjuntos de observaciones, es decir, dos muestras n_1 y n_2 de tamaño 108. Donde $\{X_{1j}, X_{2j}, Y_j\}$, $j = 1, \dots, n_1$ para aquellos valores de $x_3 \in Z$ y $\{X_{1m}, X_{2m}, Y_m\}$, $m = 1, \dots, n_2$ para aquellos valores de $x_3 \in Z^C$.

Luego, generamos un vector $\mathbf{c} = (c_1, c_2)^T$ de números reales utilizando el primer paso del procedimiento PPR (*Projection Pursuit Regression*), porque PPR selecciona estas constantes reales teniendo en cuenta tanto las respuestas Y como la covariable $\mathbf{X} = (X_1, X_2)^T$ (el vector de variables aleatorias que están incluidas en el modelo).

Basado en el vector \mathbf{c} y la covariable \mathbf{X} generamos la variable aleatoria $\mathbf{U} = \mathbf{c}^T \mathbf{X}$. Específicamente, encontramos una combinación lineal de la forma:

$$U_i = \sum_{k=1}^{r-1} c_k X_{ki} = c_1 X_{1i} + c_2 X_{2i},$$

es el valor de \mathbf{U} asociado con la i -ésima observación de X_1 y X_2 . Nuestros subconjuntos de datos son $\{U_j, Y_j\}$, $j = 1, \dots, n_1$ y $\{U_m, Y_m\}$, $m = 1, \dots, n_2$.

En estos dos subconjuntos de puntos calculamos los estimadores $\hat{\mu}_z(U)$ y $\hat{\mu}_{z^c}(U)$ de $\mu_z(U)$ y $\mu_{z^c}(U)$ respectivamente, teniendo en cuenta algunos valores del ancho de banda. Después de obtener las estimaciones anteriores procedemos a evaluar la potencia de la prueba de hipótesis.

Se Utilizaron un conjunto de valores para el ancho de banda en el rango 0,01 hasta 0,9. Se calculó $\hat{\Delta}(\lambda) = \hat{\Delta}(\lambda_1, \lambda_2)$ utilizando dos muestras n_1 y n_2 de tamaño 108 para cada valor del ancho de banda (λ). En la simulación se determinó el valor máximo de $\hat{\Delta}(\lambda)$ teniendo en cuenta el valor del ancho de banda (λ) asociado. El procedimiento se repitió 1000 veces y se contó el número de veces en que cada ancho de banda utilizado en la simulación produjo el valor máximo de la función $\hat{\Delta}(\lambda)$, considerando que el valor máximo de la función $\hat{\Delta}(\lambda)$ está asociado con el ancho de banda más adecuado para maximizar la potencia de una prueba. Ver tablas 13 y 14.

- Resultados de la simulación

Figura 16. Un ejemplo simulado para dos submuestras n_1 y n_2 de tamaño 108. Diseño de puntos aleatorios de distribución uniforme con valores en el intervalo [0,1]. Se utilizó una función $f(x)$ para generar los valores observados de Y. Errores tipo 1. Los círculos representan los valores de la función $\hat{\Delta}(\lambda)$.

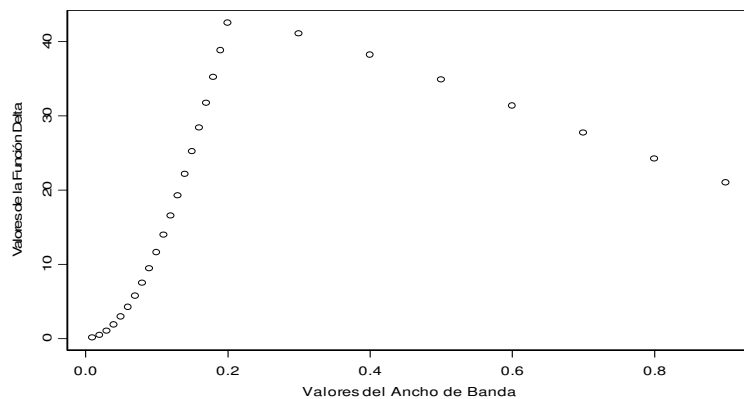
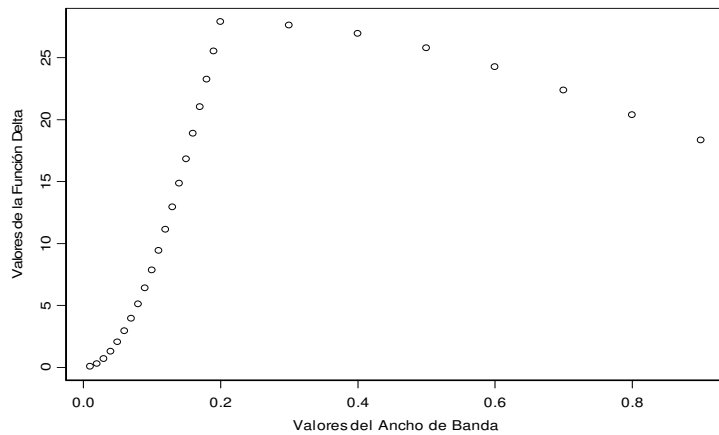


Figura 17. Un ejemplo simulado para dos submuestras n_1 y n_2 de tamaño 108. Diseño de puntos aleatorios de distribución uniforme con valores en el intervalo $[0,1]$. Se utilizó una función $f(x)$ para generar los valores observados de Y . Errores tipo 2. Los círculos representan los valores de la función $\hat{\Delta}(\lambda)$.



En la *figura 16* podemos observar que el valor de la función $\hat{\Delta}(\lambda)$ se incrementa a medida que el ancho de banda aumenta entre 0 y 0,2. Para valores del ancho de banda entre 0,3 y 0,9 decrece la función. Valores del ancho de banda entre 0,2 y 0,3 permiten encontrar un valor máximo de la función $\hat{\Delta}(\lambda)$.

Se observa en la *figura 17* que el comportamiento de la función $\hat{\Delta}(\lambda)$ es muy similar cuando se utilizan errores tipo 1 y 2 en la generación de la variable de respuesta Y .

Tabla 13. Potencia empírica (1000 repeticiones) Con errores normales tipo 1. Se generaron dos submuestras n_1 y n_2 de tamaño 108. Diseño de puntos aleatorios de distribución uniforme con valores en el intervalo [0,1]. Se utilizó una función $f(x)$ para generar los valores observados de Y .

Ancho de banda(λ)	Número de veces que λ produjo el valor máximo de la función $\hat{\Delta}(\lambda)$
0,20	316
0,30	357
0,40	268
0,50	57
0,60	2

De acuerdo a los resultados encontrados en la *tabla 13* se puede observar que el valor máximo de la función $\hat{\Delta}(\lambda)$ se encuentra asociado para valores del ancho de banda entre 0,2 y 0,3.

Tabla 14. Potencia empírica (1000 repeticiones) Con errores normales tipo 2. Se generaron dos submuestras n_1 y n_2 de tamaño 108. Diseño de puntos aleatorios de distribución uniforme con valores en el intervalo [0,1]. Se utilizó una función $f(x)$ para generar los valores observados de Y .

Ancho de banda(λ)	Número de veces que λ produjo el valor máximo de la función $\hat{\Delta}(\lambda)$
0,17	1
0,20	525
0,30	278
0,40	150
0,50	30
0,60	9
0,70	6
0,80	1

Comparando los resultados encontrados en las *tablas 13 y 14* se observa que el comportamiento del valor máximo de la función $\hat{\Delta}(\lambda)$ es similar cuando se emplean errores tipo 1 y 2 con respecto al valor del ancho de banda asociado.

◆ **Diseño de puntos no aleatorios**

- **Algunos aspectos computacionales de la simulación** Por medio de una simulación generamos 6 valores no aleatorios para las variables X_1 , X_2 y X_3 , así generamos una muestra de tamaño $n = 6*6*6 = 216$. Los datos corresponden a un diseño de puntos equidistantes con valores en el intervalo [0,1]. Los valores de la variable de respuesta Y fueron generados aplicando la función $f(x) = X_1 + X_2 + X_3$ agregando un error aleatorio. Se quiere probar si X_3 es una variable significativa en la función $f(x)$. Ver tablas 15 y 16.

- **Resultados de la simulación** Mirando la *figura 18* podemos ver que el valor de la función $\hat{\Delta}(\lambda)$ se incrementa a medida que el ancho de banda aumenta entre 0 y 0,2. Para valores del ancho de banda entre 0,4 y 0,9 decrece la función. Valores del ancho de banda entre 0,2 y 0,4 permiten encontrar un valor máximo de la función $\hat{\Delta}(\lambda)$

Figura 18. Un ejemplo simulado para dos submuestras n_1 y n_2 de tamaño 108. Diseño de puntos no aleatorios de distribución uniforme con valores en el intervalo $[0,1]$. Se utilizó una función $f(x)$ para generar los valores observados de Y . Errores tipo 1. Los círculos representan los valores de la función $\hat{\Delta}(\lambda)$.

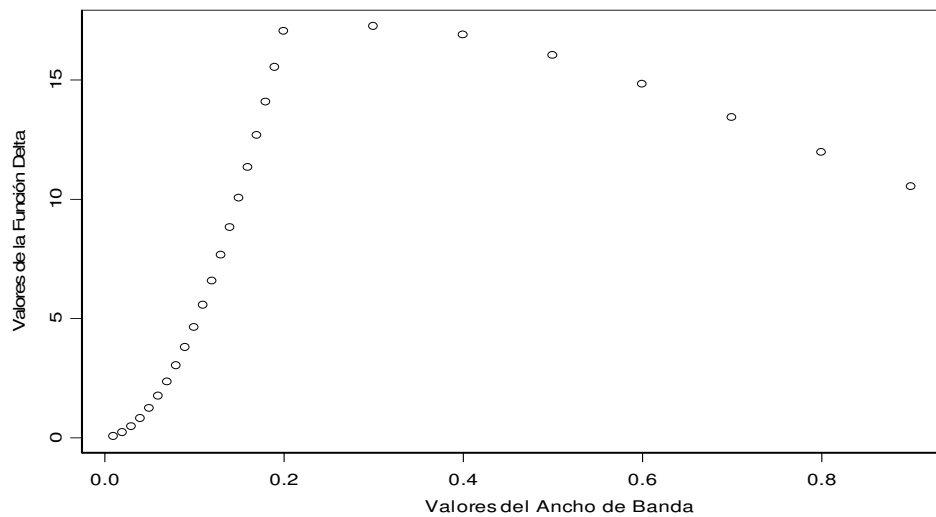
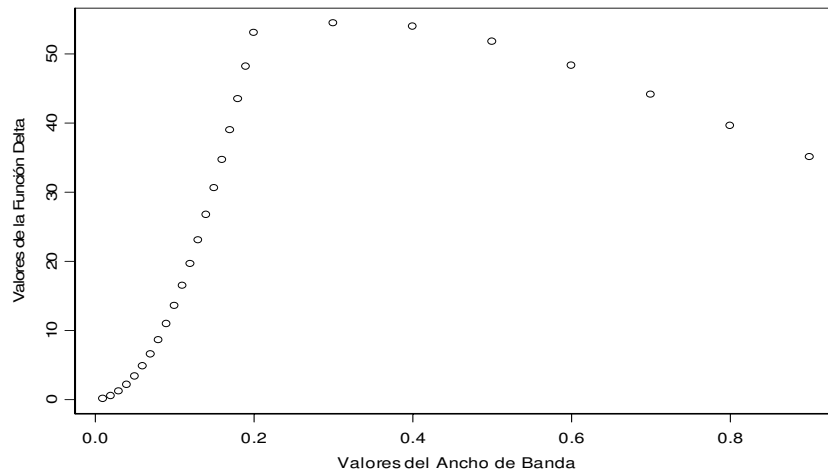


Figura 19. Un ejemplo simulado para dos submuestras n_1 y n_2 de tamaño 108. Diseño de puntos no aleatorios de distribución uniforme con valores en el intervalo $[0,1]$. Se utilizó una función $f(x)$ para generar los valores observados de Y . Errores tipo 2. Los círculos representan los valores de la función $\hat{\Delta}(\lambda)$.



Se observa en la *figura 19* que el comportamiento de la *función* $\hat{\Delta}(\lambda)$ es muy similar cuando se utilizan errores tipo 1 y 2 en la generación de la variable de respuesta Y .

Tabla 15. Potencia empírica (1000 repeticiones) Con errores normales tipo 1. Se generaron dos submuestras n_1 y n_2 de tamaño 108. Diseño de puntos no aleatorios de distribución uniforme con valores en el intervalo $[0,1]$. Se utilizó una función $f(x)$ para generar los valores observados de Y .

Ancho de banda(λ)	Número de veces que λ produjo el valor máximo de la función $\hat{\Delta}(\lambda)$
0,20	223
0,30	654
0,40	122
0,50	1

Observando los resultados encontrados en la *tabla 15* se puede ver que el valor máximo de la función $\hat{\Delta}(\lambda)$ se encuentra asociado para un valor del ancho de banda de 0,3.

Tabla 16. Potencia empírica (1000 repeticiones) Con errores normales tipo 2. Se generaron dos submuestras n_1 y n_2 de tamaño 108. Diseño de puntos no aleatorios de distribución uniforme con valores en el intervalo [0,1]. Se utilizó una función $f(x)$ para generar los valores observados de Y .

Ancho de banda(λ)	Número de veces que λ produjo el valor máximo de la función $\hat{\Delta}(\lambda)$
0,20	351
0,30	401
0,40	206
0,50	33
0,60	6
0,70	1
0,90	2

Comparando los resultados encontrados en las *tablas 15 y 16* se observa que el comportamiento del valor máximo de la función $\hat{\Delta}(\lambda)$ es similar cuando se emplean errores tipo 1 y 2 con respecto al valor del ancho de banda asociado.

4. DOCUMENTACIÓN DE PROGRAMAS EN C++ Y EN LENGUAJE S

Para realizar los procedimientos y programas siguientes se utilizó el lenguaje C++ y el paquete estadístico *S-PLUS 6.1 Professional for Windows* el cual contiene el lenguaje de programación S.

4.1 PROGRAMA NEWK1.SSC (Procedimiento en Lenguaje S)

Mediante el lenguaje S, se genera una muestra de Tamaño 100, se crea la variable independiente **X** y la variable de respuesta **Y**; se exportan datos para que sean tomados por el programa desarrollado en C++ (NEWK).

- **Descripción del Procedimiento**

1. `n<-100` // Se le Asigna 100 a la variable n (Tamaño de la Muestra).
2. `x<-seq(1, n, 1)` // Se crea la variable independiente con valores secuenciales desde 1 hasta 100. `x<-((2*x)-1)/(2*n)` // Se generan 100 valores equidistantes para la Variable Independiente entre 0 Y 1.
3. `muestra = length(x)` // Se asigna el tamaño de la muestra a la variable “muestra” que se exportara al programa desarrollado en C++.
4. `fx<-(sin(4*x)*exp(x))+3` // Se crea la variable f(x) (Función Diferenciable Continua).
5. `e<-rnorm(n, 0, .3)` // Se generan 100 errores aleatorios con Distribución Normal Estándar con media cero y $\text{Var}(\epsilon) = 0,3^2$.
6. `y<-f(x)+e` // Se crea la variable de respuesta, compuesta por la función f(x) agregando un error aleatorio.
7. `datos<-data.frame(x,y)` // Se genera un conjunto de datos que se exportaran al programa desarrollado en C++.
8. `exportData(datos,file="datos.fsc",type="FASCII",format="%20.10%20.10",colNames=F, rowNames=F)` // Exporta el Conjunto de Datos (x,y), generando un archivo ASCII con formato 10 Enteros, 10 Decimales.

9. `exportData(muestra,file="muestra.fsc",type="FASCII", colNames=F,rowNames=F)`
// Exporta el tamaño de la muestra, generando un archivo ASCII.
10. `system("newk.exe",multi=F,minimized=F, on.exec.status="stop",on.exit.status="")`
// Se ejecuta el programa desarrollado en C++.
11. `datosimport<-read.table("datestim.fsc", sep=c(1,21,41,61),`
`col.names=c("S","K","Yest","Zest"))` // Los datos obtenidos mediante el programa
en C++ son importados al S-PLUS.
12. `plot(x,y)` // Grafica de la Variable Independiente X .vs. la variable de respuesta Y.
13. `lines(x,datosimport$Yest,type="s")` // Grafica de la Variable Independiente X .vs.
valores estimados de Y.
14. `lines(x,fx,type="b")` // Grafica de la Variable Independiente X .vs. valores de la
Función $f(x)$.

4.2 PROGRAMA NEWK (Programa en C++)

- **Descripción del Programa NEWK**

1. Importa el Conjunto de Datos (X, Y).
2. Importa el tamaño de la Muestra.
3. Captura el Ancho de Banda (λ).
4. Ordena Ascendentemente los valores de la Variable Independiente X.
5. Crea los Limites de la Integral para evaluar la Función de Pesos en la Estimación de Y mediante el método de *Gasser y Müller*.
6. Calcula el Valor Estimado de Y.
7. Calcula el Valor de Z Estimado = Y_i estimado sin incluir la observación (X_i, Y_i) .
8. Calcula ECM (Error Cuadrático Medio).
9. Calcula $CV(\lambda)$ Criterio de Validación Cruzada.
10. Calcula $GCV(\lambda)$ Criterio de Validación Cruzada Generalizada según *Eubank*.
11. Calcula $GCV_{gm}(\lambda)$ Criterio de Validación Cruzada Generalizada de *Gasser y Müller*.
12. Exporta Datos para el S-PLUS:
 - a. $s[n]$ = Limites de la Integral para evaluar la Función de Pesos en la estimación de la variable Y utilizando el Método de *Gasser y Müller*.

- b. $k[n,n]$ = Matriz de Función de Pesos(Función Kernel Cuadrático).
- c. $yest[n]$ = Estimación de la variable Y.
- d. $zest[n]$ = Y_i estimado sin incluir la observación (X_i, Y_i) .
- e. λ = Ancho de Banda (λ).
- f. tr_s_lambda = Traza de la Matriz de Función de Pesos.
- g. ECM (Error Cuadrático Medio).
- h. $CV(\lambda)$ = Criterio de Validación Cruzada.
- i. $GCV(\lambda)$ = Criterio de Validación Cruzada Generalizada según *Eubank*.
- j. $GCV_gm(\lambda)$ = Criterio de Validación Cruzada Generalizada de *Gasser y Müller*.

4.3 PROGRAMA NEWK1000.SSC (Procedimiento en Lenguaje S)

Mediante el lenguaje S, se genera una muestra de Tamaño 100, se crea la variable independiente X , se genera un ciclo en el cual se crea la variable de respuesta Y agregando un error aleatorio. Este procedimiento se realiza 1000 veces; se exportan los datos para que sean tomados por el programa desarrollado en C++ (NEWK1000)

Su objetivo principal es poder determinar el valor mínimo para el Criterio de Validación Cruzada Generalizada según *Gasser y Müller* teniendo en cuenta el valor del Parámetro de Suavización (λ) asociado.

- **Descripción del Procedimiento**

1. `n<-100` // Se le Asigna 100 a la variable n (Tamaño de la Muestra).
2. `x<-seq(1,n,1)` // Se crea la variable independiente con valores secuenciales desde 1 hasta 100.
3. `x<-((2*x)-1)/(2*n)` // Se generan 100 valores equidistantes para la Variable Independiente X entre 0 y 1.
4. `muestra=length(x)` // Se asigna el tamaño de la muestra a la variable “muestra” que se exportara al programa desarrollado en C++.
5. `exportData(muestra,file="muestra.fsc",type="FASCII", colNames=F,rowNames=F)` // Exporta el tamaño de la muestra, generando un archivo ASCII.

6. `fx<-(sin(4*x)*exp(x))+3` // Se crea la variable $f(x)$ (Función Diferenciable Continua).
7. `for(i in 1:1000)` // Ciclo para generar la Variable de respuesta agregando un error aleatorio por cada repetición.
 - a. `e<-rnorm(n,0,.3)` // Se generan 100 errores aleatorios con Distribución Normal Estándar con media cero y $\text{Var}(\epsilon) = 0,3^2$ por cada repetición.
 - b. `y<-fx+e` // Se crea la variable de respuesta, compuesta por la función $f(x)$ agregando un error aleatorio.
 - c. `datos<- data.frame(x,y)` // Se genera un conjunto de datos que se exportaran al programa desarrollado en C++.
 - d. `exportData(datos,file="datos.fsc",type="FASCII",format="%20.10%20.10",colNames=F,rowNames=F)` // Exporta el Conjunto de Datos (x,y) , generando un archivo ASCII con formato 10 Enteros, 10 Decimales.
 - e. `system("newk1000.exe",multi=F,minimized=T, on.exec.status="stop", on.exit.status="")`
 // Se ejecuta el programa desarrollado en C++.

4.4 PROGRAMA NEWK1000 (Programa en C++)

- **Descripción del Programa NEWK1000**

1. Importa el Conjunto de Datos (X, Y) .
2. Importa el tamaño de la Muestra.
3. Fija 27 valores distintos para el Ancho de Banda (λ) .
4. Ordena Ascendentemente los valores de la Variable Independiente X .
5. Crea los Limites de la Integral para evaluar la Función de Pesos en la Estimación de Y mediante el método de Gasser y Müller.
6. Calcula el Valor Estimado de Y .
7. Calcula el Valor Estimado de $Z = Y_i$ estimado sin incluir la observación (X_i, Y_i)
8. Calcula ECM (Error Cuadrático Medio)
9. Calcula $CV(\lambda) = \text{Criterio de Validación Cruzada}$
10. Calcula $GCV(\lambda) = \text{Criterio de Validación Cruzada Generalizada según Eubank}$

11. Calcula $GCV_gm(\lambda)$ = Criterio de Validación Cruzada Generalizada de Gasser y Müller.
12. Exporta Datos para la Hoja Electrónica, donde se almacenan las frecuencias correspondientes a los Valores Mínimos del Criterio de Validación Cruzada Generalizada por Gasser y Müller asociados al valor de Lambda (λ).
Por *Gasser y Müller*:
 - a. $lambda_gm_min$ = ancho de banda.
 - b. $tr_s_gm_min$ = Traza de la Matriz de Función de Pesos.
 - c. gcv_gm_min = Criterio de Validación Cruzada Generalizada de *Gasser y Müller*.
 - d. ecm_gm_min = ECM (Error Cuadrático Medio).
 - e. cv_gm_min = $CV(\lambda)$ = Criterio de Validación Cruzada.
 - f. gcv_min_gm = $GCV(\lambda)$ = Criterio de Validación Cruzada Generalizada según *Eubank*.

4.5 PROGRAMA NEWK1001.SSC (Procedimiento en Lenguaje S)

Mediante el lenguaje S, se genera una muestra de Tamaño 100, se crea la variable independiente **X**, se genera un ciclo en el cual se crea la variable de respuesta **Y** agregando un error aleatorio. Este procedimiento se realiza 1000 veces; se exportan los datos para que sean tomados por el programa desarrollado en C++ (NEWK1001).

Su objetivo principal es poder determinar el valor máximo de la Función

$$\hat{\Delta}(\lambda_1, \lambda_2) = \hat{\Delta}(\lambda) \text{ para el caso } \lambda_1 = \lambda_2, \text{ y una sola función } f(x) \text{ en las dos muestras.}$$

- **Descripción del Procedimiento**

1. $n < 100$ // Se le Asigna 100 a la variable n (Tamaño de la Muestra).
2. $x < \text{seq}(1, n, 1)$ // Se crea la variable independiente con valores secuenciales desde 1 hasta 100.
3. $x < ((2 * x) - 1) / (2 * n)$ // Se generan 100 valores equidistantes para la Variable Independiente X entre 0 y 1.

4. `muestra=length(x)` // Se asigna el tamaño de la muestra a la variable “muestra” que se exportara al programa desarrollado en C++.
5. `exportData(muestra,file="muestra.fsc",type="FASCII",
colNames=F,rowNames=F)` // Exporta el tamaño de la muestra, generando un archivo ASCII.
6. `fx<-(sin(4*x)*exp(x))+3` // Se crea la variable $f(x)$ (Función Diferenciable Continua).
7. `for (i in 1:1000)` // Ciclo para generar la Variable de respuesta agregando un error aleatorio por cada repetición.
 - a. `e<-rnorm(n,0,.3)` // Se generan 100 errores aleatorios con Distribución Normal Estándar con media cero y $\text{Var}(\epsilon) = 0,3^2$ por cada repetición.
 - b. `y<-fx+e` // Se crea la variable de respuesta, compuesta por la función $f(x)$ agregando un error aleatorio.
 - c. `datos<- data.frame(x,y)` // Se genera un conjunto de datos que se exportaran al programa desarrollado en C++.
 - d. `exportData(datos,file="datos.fsc",type="FASCII",format="%20.10%20.10",
colNames=F,rowNames=F)`// Exporta el Conjunto de Datos (x,y) , generando un archivo ASCII con formato 10 Enteros, 10 Decimales.
 - e. `system("newk1001.exe",multi=F,minimized=T, on.exec.status="stop",
on.exit.status="")`
// Se ejecuta el programa desarrollado en C++.

4.6 PROGRAMA NEWK1001 (Programa en C++)

- **Descripción del Programa NEWK1001**

1. Importa el Conjunto de Datos (X, Y)
2. Importa el tamaño de la Muestra.
3. Fija 27 valores distintos para el Ancho de Banda (λ).
4. Ordena Ascendentemente los valores de la Variable Independiente X.
5. Genera las Muestras de tamaño $n/2$ con las variables independientes $X_{n_1}[i]$, $X_{n_2}[i]$; y las variables de respuesta $Y_{n_1}[i]$, $Y_{n_2}[i]$ teniendo en cuenta el subíndice.

- a. Datos para el Subíndice Impar corresponden a $X_{n_1}[i]$, $Y_{n_1}[i]$ y $Yest_{n_1}[i]$.
- b. Datos para el Subíndice Par corresponden a $X_{n_2}[i]$, $Y_{n_2}[i]$ y $Yest_{n_2}[i]$.
- 6. Crea los Limites de la Integral para evaluar la Función de Pesos para cada Muestra.
- 7. Genera Ciclo (27 Repeticiones) para determinar el Valor Máximo de la Función Delta $(\lambda_1, \lambda_2) = \hat{\Delta}(\lambda)$ de acuerdo para algunos valores del ancho de banda (λ) .
 - a. Calcula los valores estimados de Y para la Muestra 1.
 - b. Calcula los valores estimados de Y para la Muestra 2.
 - c. Calcula la Matriz de Pesos (Función Kernel Cuadrático).
 - d. Calcula Nuevos Valores de Estimación para la variable de respuesta Y para cada una de las Muestras.
 - e. Calcula la función $\hat{\Delta}(\lambda)$
- 8. Exporta Datos para la Hoja Electrónica, donde se almacenan las frecuencias correspondientes a los Valores Máximos de la función $\hat{\Delta}(\lambda)$ asociados al valor de Lambda (λ) .
 - a. λ_max = Valor del Ancho de Banda asociado al valor máximo de la Función $\hat{\Delta}(\lambda)$.
 - b. δ_max = Valor Máximo de la Función $\hat{\Delta}(\lambda)$.

4.7 PROGRAMA NEWK1002.SSC (Procedimiento en Lenguaje S)

Mediante el lenguaje S, se genera una muestra de Tamaño 50, se crea la variable independiente X , se genera un ciclo en el cual se crea la variable de respuesta Y agregando un error aleatorio. Este procedimiento se realiza 1000 veces; se exportan los datos para que sean tomados por el programa desarrollado en C++ (NEWK1001).

Su objetivo principal es poder determinar el valor máximo de la Función Delta $(\lambda_1, \lambda_2) = \hat{\Delta}(\lambda_1, \lambda_2) = \hat{\Delta}(\lambda)$, para el caso $\lambda_1 = \lambda_2$.

Este Procedimiento emplea dos funciones para generar los valores observados de la variable de respuesta Y: $f(x)$ y $g(x)$

- $f(x)$ para calcular los valores observados de Y en la Muestra 1
- $g(x)$ para calcular los valores observados de Y en la Muestra 2

• **Descripción del Procedimiento**

1. `n<-50` // Se le Asigna 50 a la variable n (Tamaño de la Muestra).
2. `x<-seq(1,n,1)` // Se crea la variable independiente con valores secuenciales desde 1 hasta 50.
3. `x<-((2*x)-1)/(2*n)` // Se generan 50 valores equidistantes para la Variable Independiente X entre 0 y 1 .
4. `muestra=length(x)` // Se asigna el tamaño de la muestra a la variable “muestra” que se exportara al programa desarrollado en C++.
5. `exportData(muestra,file="muestra.fsc",type="FASCII", colNames=F,rowNames=F)` // Exporta el tamaño de la muestra, generando un archivo ASCII.
6. `fx<-(sin(4*x)*exp(x))+3` // Se crea la variable $f(x)$ (Función Diferenciable Continua).
7. `gx<-cos(4*x)` // Se crea la variable $g(x)$ (Función Diferenciable Continua).
8. `for (i in 1:1000)` // Ciclo para generar la Variable de respuesta agregando un error aleatorio por cada repetición.
 - a. `e<-rnorm(n,0,.3)` // Se generan 50 errores aleatorios con Distribución Normal Estándar con media cero y $\text{Var}(\epsilon) = 0,3^2$ por cada repetición.
 - b. `Y1<-fx+e` // Se crea la variable de respuesta, compuesta por la función $f(x)$ agregando un error aleatorio para la muestra n_1 .
 - c. `e<-rnorm(n,0,.3)` // Se generan 50 errores aleatorios con Distribución Normal Estándar con media cero y $\text{Var}(\epsilon) = 0,3^2$ por cada repetición.
 - d. `Y2<-gx+e` // Se crea la variable de respuesta, compuesta por la función $g(x)$ agregando un error aleatorio para la muestra n_2 .

- e. `datos<- data.frame(x,Y1,Y2) // Se genera un conjunto de datos que se exportaran al programa desarrollado en C++.`
- f. `exportData(datos,file="datos.fsc",type="FASCII",format="%20.10%20.10%20.10",colNames=F, rowNames=F) // Exporta el Conjunto de Datos (x,Y1,Y2), generando un archivo ASCII con formato 10 Enteros, 10 Decimales.`
- g. `system("newk1002.exe",multi=F,minimized=T, on.exec.status="stop", on.exit.status="")`
// Se ejecuta el programa desarrollado en C++.

4.8 PROGRAMA NEWK1002 (Programa en C++)

- **Descripción del Programa NEWK1002**

1. Importa el Conjunto de Datos (X, Y₁, Y₂).
2. Importa el tamaño de la Muestra.
3. Fija 27 valores distintos para el Ancho de Banda (λ).
4. Ordena Ascendentemente los valores de la Variable Independiente X.
5. Genera las Muestras de tamaño n con las variables independientes X_{n1}[i], X_{n2}[i]; y las variables de respuesta Y_{n1}[i], Y_{n2}[i].
6. Crea los Limites de la Integral para evaluar la Función de Pesos para cada Muestra.
7. Genera Ciclo (27 Repeticiones) para determinar el Valor Máximo de la Función $\Delta = \hat{\Delta}(\lambda_1, \lambda_2) = \hat{\Delta}(\lambda)$ de acuerdo para algunos valores del ancho de banda (λ).
 - a. Calcula los valores estimados de Y para la Muestra 1.
 - b. Calcula los valores estimados de Y para la Muestra 2.
 - c. Calcula la Matriz de Pesos (Función Kernel Cuadrático).
 - d. Calcula Nuevos Valores de Estimación para la variable de respuesta Y para cada una de las Muestras.
 - e. Calcula la función $\hat{\Delta}(\lambda)$.

8. Exporta Datos para la Hoja Electrónica, donde se almacenan las frecuencias correspondientes a los Valores Máximos de la función $\hat{\Delta}(\lambda)$ asociados al valor de Lambda (λ).
- lambda_max = Valor del Ancho de Banda asociado al valor máximo de la Función $\hat{\Delta}(\lambda)$.
 - delta_max = Valor Máximo de la Función $\hat{\Delta}(\lambda)$.

4.9 PROGRAMA NEWK1003.SSC (Procedimiento en Lenguaje S)

Este procedimiento es equivalente al **NEWK1002.SSC**, la diferencia reside en el tamaño de la muestra y en la ejecución de otro programa desarrollado en C++.

Mediante el lenguaje S, se genera una muestra de Tamaño 100, se crea la variable independiente **X**, se genera un ciclo en el cual se crea la variable de respuesta **Y** agregando un error aleatorio. Este procedimiento se realiza 1000 veces; se exportan los datos para que sean tomados por el programa desarrollado en C++ (NEWK1003).

Su objetivo principal es poder determinar el valor máximo de la Función Delta (λ_1, λ_2) = $\hat{\Delta}(\lambda)$, para el caso $\lambda_1 = \lambda_2$.

Este Procedimiento emplea dos funciones para generar los valores observados de la variable de respuesta Y: $f(x)$ y $g(x)$

- $f(x)$ para calcular los valores observados de Y en la Muestra 1
- $g(x)$ para calcular los valores observados de Y en la Muestra 2

- **Descripción del Procedimiento**

1. `n<-100` // Se le Asigna 100 a la variable n (Tamaño de la Muestra).
2. `x<-seq(1,n,1)` // Se crea la variable independiente con valores secuenciales desde 1 hasta 100.
3. `x<-((2*x)-1)/(2*n)` // Se generan 100 valores equidistantes para la Variable Independiente X entre 0 y 1.

4. `muestra=length(x)` // Se asigna el tamaño de la muestra a la variable “muestra” que se exportara al programa desarrollado en C++.
5. `exportData(muestra,file="muestra.fsc",type="FASCII", colNames=F,rowNames=F)`
// Exporta el tamaño de la muestra, generando un archivo ASCII.
6. `fx<-(sin(4*x)*exp(x))+3` // Se crea la variable $f(x)$ (Función Diferenciable Continua).
7. `gx<-(cos(4*x))` // Se crea la variable $g(x)$ (Función Diferenciable Continua).
8. `for (i in 1:1000)` // Ciclo para generar la Variable de respuesta agregando un error aleatorio por cada repetición.
 - a. `e<-rnorm(n,0,.3)` // Se generan 100 errores aleatorios con Distribución Normal Estándar con media cero y $\text{Var}(\epsilon) = 0,3^2$ por cada repetición.
 - b. `Y1<-fx+e` // Se crea la variable de respuesta y_1 , compuesta por la Función $f(x)$ agregando un error aleatorio para la muestra n_1 .
 - c. `e<-rnorm(n,0,.3)` // Se generan 100 errores aleatorios con Distribución Normal Estándar con media cero y $\text{Var}(\epsilon) = 0,3^2$ por cada repetición.
 - d. `Y2<-gx+e` // Se crea la variable de respuesta y_2 , compuesta por la Función $g(x)$ agregando un error aleatorio para la muestra n_2 .
 - e. `datos<-data.frame(x,y1,y2)` // Se genera un conjunto de datos que se exportaran al programa desarrollado en C++.
 - f. `exportData(datos,file="datos.fsc",type="FASCII",format="%20.10%20.10%20.10",colNames=F, rowNames=F)` // Exporta el Conjunto de Datos (x,y_1,y_2) , generando un archivo ASCII con formato 10 Enteros, 10 Decimales.
 - g. `system("newk1003.exe",multi=F,minimized=T, on.exec.status="stop", on.exit.status="")`
// Se ejecuta el programa desarrollado en C++.

4.10 PROGRAMA NEWK1003 (Programa en C++)

Este programa es equivalente al **NEWK1002.CPP**, la diferencia reside en el tamaño de la muestra y en la creación de dos submuestras n_1 y n_2 según valores pares e impares del subíndice para la variable independiente **X**.

- **Descripción del Programa NEWK1003**

1. Importa el Conjunto de Datos (X, Y_1, Y_2).
2. Importa el tamaño de la Muestra.
3. Fija 27 valores distintos para el Ancho de Banda (λ).
4. Ordena Ascendentemente los valores de la Variable Independiente **X**.
5. Genera las Muestras de tamaño $n/2$ con las variables independientes $X_{n_1}[i]$, $X_{n_2}[i]$; y las variables de respuesta $Y_{n_1}[i]$, $Y_{n_2}[i]$ teniendo en cuenta el subíndice.
 - a. Datos para el Subíndice Impar corresponden a $X_{n_1}[i]$, $Y_{n_1}[i]$
 - b. Datos para el Subíndice Par corresponden a $X_{n_2}[i]$, $Y_{n_2}[i]$
6. Crea los Limites de la Integral para evaluar la Función de Pesos para cada Muestra.
7. Genera Ciclo (27 Repeticiones) para determinar el Valor Máximo de la Función $\Delta(\lambda)$ de acuerdo para algunos valores del ancho de banda (λ)
 - a. Calcula los valores estimados de **Y** para la Muestra 1.
 - b. Calcula los valores estimados de **Y** para la Muestra 2.
 - c. Calcula la Matriz de Pesos (Función Kernel Cuadrático).
 - d. Calcula Nuevos Valores de Estimación para la variable de respuesta **Y** para cada una de las Muestras.
 - e. Calcula la función $\hat{\Delta}(\lambda)$.
8. Exporta Datos para la Hoja Electrónica, donde se almacenan las frecuencias correspondientes a los Valores Máximos de la función $\hat{\Delta}(\lambda)$ asociados al valor de Lambda (λ).

- a. λ_{max} = Valor del Ancho de Banda asociado al valor máximo de la Función $\hat{\Delta}(\lambda)$.
- b. δ_{max} = Valor Máximo de la Función $\hat{\Delta}(\lambda)$.

4.11 PROGRAMA NEWK1004.SSC (Procedimiento en Lenguaje S)

Mediante el lenguaje S, se crean dos variables independientes no aleatorias X_1 y X_2 , para cada variable se generan 10 valores equidistantes en el intervalo [0, 1]. Se generan dos submuestras de Tamaño 50. Se genera un ciclo en el cual se crea la variable de respuesta Y agregando en cuenta un error aleatorio. Este procedimiento se realiza 1000 veces; se exportan los datos para que sean tomados por el programa desarrollado en C++ (NEWK1002). Ver descripción del procedimiento en la Pág. 104.

Su objetivo principal es poder determinar el valor máximo de la Función $\Delta(\lambda_1, \lambda_2) = \hat{\Delta}(\lambda)$, para el caso $\lambda_1 = \lambda_2$.

Este Procedimiento emplea dos funciones similares para generar los valores observados de la variable de respuesta Y : $f(x)$ y $g(x)$.

- $f(x) = X_1 + X_2$ permite calcular los valores observados de Y en la Muestra 1 para valores de X_2 menores o iguales a 0,5.
- $g(x) = X_1 + X_2$ permite calcular los valores observados de Y en la Muestra 2 para valores de X_2 mayores a 0,5.

Se puede observar que realmente se emplea una función $f(x) = X_1 + X_2$ para generar los valores de Y .

- **Descripción del Procedimiento**

1. # Valores de la Variable Independiente $X_1 \rightarrow [0,1]$
2. $n < 10$ // Se le Asigna 10 a la variable n (Número de datos a generar en la Muestra por cada Variable).
3. $X_1 < 0$ // Inicializa la Variable Independiente X_1

4. $X_1 \leftarrow \text{seq}(1, n, 1)$ // Se crea la variable independiente con valores secuenciales desde 1 hasta 10.
5. $X_1 \leftarrow ((2 * X_1) - 1) / (2 * n)$ // Se generan 10 valores equidistantes para la Variable Independiente X_1 entre 0 y 1.
6. # Valores de la Variable Independiente $X_2 \rightarrow [0, 1]$
7. $X_2 \leftarrow 0$ // Inicializa la Variable Independiente X_2
8. $X_2 \leftarrow X_1$ // Asigna los valores de la Variable Independiente X_1 a la Variable Independiente X_2
9. # Subconjunto Z de la Variable Independiente $X_2 \Rightarrow$
 - a. # $X_{2a} \rightarrow [0, 1]$ para la muestra n_1
10. # Criterio $X_{2a} \leq .5$
11. $X_{2a} \leftarrow 0$ // Inicializa Subconjunto Z de la Variable Independiente X_2
12. $X_{2a} \leftarrow X_2 [X_2 \leq .5]$ // Crea el Subconjunto Z de la Variable Independiente X_2 para aquellos valores menores o iguales a .5
13. # Subconjunto Z^C (Complemento de Z) de la Variable Independiente $X_2 \Rightarrow$ # $X_{2b} \rightarrow [0, 1]$ para la muestra n_2
14. # Criterio $X_{2b} > .5$
15. $X_{2b} \leftarrow 0$ // Inicializa Subconjunto Z^C de la Variable Independiente X_2
16. $X_{2b} \leftarrow X_2 [X_2 > .5]$ // Crea el Subconjunto Z^C de la Variable Independiente X_2 para aquellos valores mayores a .5
17. $x \leftarrow 0$ // Inicializa la Variable Independiente X Resultante
18. $f_x \leftarrow 0$ // Inicializa Variable $f(x)$ (Función Diferenciable Continua).
19. $g_x \leftarrow 0$ // Inicializa Variable $g(x)$ (Función Diferenciable Continua).
20. $k \leftarrow 1$ // Inicializa Subíndice para la Variable Independiente X
21. # Genera las Variables $f(x)$, $g(x)$ mediante ciclos anidados para el manejo del plano cartesiano (X_1 , $\#X_{2a}$) ó (X_1 , X_{2b})
22. for (i in 1:10)
 - {
 - for (j in 1:5)
 - {
 - a.x[k] ← $X_1[i]$ // Crea la Variable Independiente X.

```

    b.fx[k]<- X1[i]+X2a[j] // Crea la Variable f(x) (Función Diferenciable
        Continua).
    c.gx[k]<- X1[i]+X2b[j] // Crea la Variable g(x) (Función Diferenciable
        Continua).
    d.k<-k+1 // Incrementa Subíndice
    }
}

```

23. #Define tamaño de cada submuestra Resultante

24. n<-50

25. #Inicializa las Variables de Respuesta temporales

26. y1<-0

27. y2<-0

28. muestra=length(x) // Se asigna el tamaño de la submuestra a la variable “muestra” que se exportara al programa desarrollado en C++.

29. exportData(muestra,file="muestra.fsc",type="FASCII",colNames=F,rowNames=F)
// Exporta el tamaño de la submuestra, generando un archivo ASCII.

30. for (i in 1:1000) // Ciclo para generar la Variable de respuesta agregando un error aleatorio por cada repetición.

a. e<-rnorm(n,0,.3) // Se generan 50 errores aleatorios con Distribución Normal Estándar con media cero y $\text{Var}(\epsilon) = 0,3^2$ por cada repetición.

b. y1<-fx+e // Se crea la variable de respuesta y1 compuesta por la Función f(x) agregando un error aleatorio para la muestra n1.

c. e<-rnorm(n,0,.3) // Se generan 50 errores aleatorios con Distribución Normal Estándar con media cero y $\text{Var}(\epsilon) = 0,3^2$ por cada repetición.

d. Y2<-gx+e // Se crea la variable de respuesta y2, compuesta por la Función g(x) agregando un error aleatorio para la muestra n2.

e. datos<- data.frame(x,y1,y2) // Se genera un conjunto de datos que se exportaran al programa desarrollado en C++.

f. exportData(datos,file="datos.fsc",type="FASCII",format="%20.10%20.10%20.10",colNames=F, rowNames=F) // Exporta el Conjunto de Datos (x,y1,y2), generando un archivo ASCII con formato 10 Enteros, 10 Decimales.

```
g. system("newk1002.exe",multi=F,minimized=T, on.exec.status="stop",
on.exit.status="")
```

// Se ejecuta el programa desarrollado en C++.

4.12 PROGRAMA NEWK1004_RANDOM.SSC (Procedimiento en lenguaje S)

Mediante el lenguaje S, se generan dos variables independientes aleatorias X_1 y X_2 , para cada variable se generan aleatoriamente 10 valores uniformemente distribuidos en el intervalo $[0, 1]$. Se generan dos submuestras de Tamaño 50. Se genera un ciclo en el cual se crea la variable de respuesta Y agregando un error aleatorio. Este procedimiento se realiza 1000 veces; se exportan los datos para que sean tomados por el programa desarrollado en C++ (NEWK1002). Ver descripción del procedimiento en la Pág. 104.

Su objetivo principal es poder determinar el valor máximo de la Función

Delta $(\lambda_1, \lambda_2) = \hat{\Delta}(\lambda)$, para el caso $\lambda_1 = \lambda_2$

Este Procedimiento emplea dos funciones similares para generar los valores observados de la variable de respuesta Y : $f(x)$ y $g(x)$.

- $f(x) = X_1 + X_2$ permite calcular los valores observados de Y en la Muestra 1 para valores de X_2 menores o iguales a 0,5.
- $g(x) = X_1 + X_2$ permite calcular los valores observados de Y en la Muestra 2 para valores de X_2 mayores a 0,5.

- **Descripción del Procedimiento**

1. # Valores de la Variable Independiente $X_1 \rightarrow [0,1]$
2. $n < 10$ // Se le Asigna 10 a la variable n (Número de datos a generar en la Muestra por cada Variable).
3. $X_1 < 0$ // Inicializa la Variable Independiente X_1
4. $X_1 < -c(\text{sort}(\text{runif}(10,0,1)))$ // Se crea la variable independiente con 10 valores aleatorios entre 0 y 1.
5. # Valores Variable Independiente $X_2 \rightarrow [0,1]$
6. $X_2 < 0$ // Inicializa la Variable Independiente X_2

7. $X_2 \leftarrow c(\text{runif}(1,0,.5), \text{runif}(1,0,.5), \text{runif}(1,0,.5),$
 $\text{runif}(1,0,.5), \text{runif}(1,0,.5), \text{unif}(1,.51,1),$
 $\text{runif}(1,.51,1), \text{runif}(1,.51,1), \text{runif}(1,.51,1), \text{runif}(1,.51,1))$
 - a. // Se crea la variable independiente con 10 valores aleatorios entre 0 y 1.
8. # Subconjunto Z de la Variable Independiente $X_2 \Rightarrow$
 - a. # $X_{2a} \rightarrow [0,1]$ para la muestra n_1
9. # Criterio $X_{2a} \leq .5$
10. $X_{2a} \leftarrow 0$ // Inicializa Subconjunto Z de la Variable Independiente X_2
11. $X_{2a} \leftarrow \text{sort}(X_2[X_2 \leq .5])$ // Crea el Subconjunto Z de la Variable Independiente X_2 para aquellos valores menores o iguales a .5 en orden ascendente
12. # Subconjunto Z^C (Complemento de Z) de la Variable Independiente $X_2 \Rightarrow$ # $X_{2b} \rightarrow [0,1]$ para la muestra n_2
13. # Criterio $X_{2b} > .5$
14. $X_{2b} \leftarrow 0$ // Inicializa Subconjunto Z^C de la Variable Independiente X_2
15. $X_{2b} \leftarrow \text{sort}(X_2[X_2 > .5])$ // Crea el Subconjunto Z^C de la Variable Independiente X_2 para aquellos valores mayores a .5 en orden ascendente
16. $x \leftarrow 0$ // Inicializa la Variable Independiente x Resultante
17. $fx \leftarrow 0$ // Inicializa la Variable f(x) (Función Diferenciable Continua).
18. $gx \leftarrow 0$ // Inicializa la Variable g(x) (Función Diferenciable Continua).
19. $k \leftarrow 1$ // Inicializa Subíndice para la Variable Independiente x
20. #Genera Variables f(x) ,g(x) mediante ciclos anidados para el manejo del plano cartesiano (X_1, X_{2a}) ú (X_1, X_{2b})
21. for (i in 1:10)
 - {
 - for (j in 1:5)
 - {
 - a. $x[k] \leftarrow x_1[i]$ // Crea la Variable Independiente X.
 - b. $fx[k] \leftarrow x_1[i] + x_{2a}[j]$ // Crea la Variable f(x) (Función Diferenciable Continua).
 - c. $gx[k] \leftarrow x_1[i] + x_{2b}[j]$ // Crea la Variable g(x) (Función Diferenciable Continua).
 - d. $k \leftarrow k + 1$ // Incrementa Subíndice


```

    }
}
22. #Define tamaño de cada submuestra Resultante
23. n<-50
24. #Inicializa las Variables de Respuesta temporales
25. y1<-0
26. y2<-0
27. muestra = length(x) // Se asigna el tamaño de la submuestra a la variable "muestra"
    que se exportara al programa desarrollado en C++.
28. exportData(muestra,file="muestra.fsc",type="FASCII", colNames=F,rowNames=F)
    // Exporta el tamaño de la submuestra, generando un archivo ASCII.
29. for (i in 1:1000) // Ciclo para generar la Variable de respuesta agregando un error
    aleatorio por cada repetición.
    a. e<-rnorm(n,0,.3) // Se generan 50 errores aleatorios con Distribución Normal
        Estándar con media cero y Var ( $\epsilon$ ) =  $0,3^2$  por cada repetición.
    b. y1<-fx+e // Se crea la variable de respuesta y1, compuesta por la Función f(x)
        agregando un error aleatorio para la muestra n1.
    c. e<-rnorm(n,0,.3) // Se generan 50 errores aleatorios con Distribución Normal
        Estándar con media cero y Var ( $\epsilon$ ) =  $0,3^2$  por cada repetición.
    d. Y2<-gx+e // Se crea la variable de respuesta y2, compuesta por la Función
        g(x) agregando un error aleatorio para la muestra n2.
    e. datos<- data.frame(x,y1,y2) // Se genera un conjunto de datos que se
        exportaran al programa desarrollado en C++.
    f. exportData(datos,file="datos.fsc",type="FASCII",format="%20.10%20.10%
        20.10",colNames=F, rowNames=F) // Exporta el Conjunto de Datos
        (x,y1,y2), generando un archivo ASCII con formato 10 Enteros, 10
        Decimales.
    g. system("newk1002.exe",multi=F,minimized=T, on.exec.status="stop",
        on.exit.status="")
    // Se ejecuta el programa desarrollado en C++.

```

4.13 PROGRAMA NEWK1005.SSC (Procedimiento en Lenguaje S)

Mediante el lenguaje S, se crean tres variables independientes no aleatorias X_1 , X_2 y X_3 , para cada variable se generan 6 valores equidistantes en el intervalo $[0, 1]$. Se generan dos submuestras de Tamaño 108. Se genera un ciclo en el cual se crea la variable de respuesta Y agregando un error aleatorio. Este procedimiento se realiza 1000 veces; se exportan los datos para que sean tomados por el programa desarrollado en C++ (NEWK1005).

Su objetivo principal es poder determinar el valor máximo de la Función

$$\Delta(\lambda_1, \lambda_2) = \hat{\Delta}(\lambda_1, \lambda_2) = \hat{\Delta}(\lambda), \text{ para el caso } \lambda_1 = \lambda_2$$

Este Procedimiento emplea dos funciones similares para generar los valores observados de la variable de respuesta Y : $f(x)$ y $g(x)$.

- $f(x) = X_1 + X_2 + X_3$ permite calcular los valores observados de Y en la Muestra 1 para valores de X_3 menores o iguales a 0,5.
- $g(x) = X_1 + X_2 + X_3$ permite calcular los valores observados de Y en la Muestra 2 para valores de X_3 mayores a 0,5.

Se puede observar que realmente se emplea una función $f(x) = X_1 + X_2 + X_3$ para generar los valores de Y .

• Descripción del Procedimiento

1. # Valores de la Variable Independiente $X_1 \rightarrow [0,1]$
2. $n < 6$ // Se le Asigna 6 a la variable n (Número de datos a generar en la Muestra por cada Variable).
3. $X_1 < 0$ // Inicializa la Variable Independiente X_1
4. $X_1 < \text{seq}(1, n, 1)$ // Se crea la variable independiente con valores secuenciales desde 1 hasta 6.
5. $X_1 < -((2 * x1) - 1) / (2 * n)$ // Se generan 6 valores uniformemente distribuidos para la Variable Independiente X_1 .
6. # Valores de la Variable Independiente $X_2 \rightarrow [0,1]$

7. $X_2 < -0$ // Inicializa la Variable Independiente X_2
8. $X_2 < -X_1$ // Asigna los valores de la Variable Independiente X_1 a la Variable Independiente X_2
9. # Valores Variable Independiente $X_3 \rightarrow [0,1]$
10. $X_3 < -0$ // Inicializa la Variable Independiente X_3
11. $X_3 < -X_1$ // Asigna los valores de la Variable Independiente X_1 a la Variable Independiente X_3
12. # Subconjunto Z de la Variable Independiente $X_3 \Rightarrow$
 - a. # $X_{3a} \rightarrow [0,1]$ para la muestra n_1
13. # Criterio $X_{3a} \leq .5$
14. $X_{3a} < -0$ // Inicializa Subconjunto Z de la Variable Independiente X_3
15. $X_{3a} < -x_3[x_3 \leq .5]$ // Crea el Subconjunto Z de la Variable Independiente X_3 para aquellos valores menores o iguales a $.5$
16. # Subconjunto Z^C (Complemento de Z) de la Variable Independiente $X_3 \Rightarrow$ # $X_{3b} \rightarrow [0,1]$ para la muestra n_2
17. # Criterio $X_{3b} > .5$
18. $X_{3b} < -0$ // Inicializa Subconjunto Z^C de la Variable Independiente X_3
19. $X_{3b} < -x_3[x_3 > .5]$ // Crea el Subconjunto Z^C de la Variable Independiente X_3 para aquellos valores mayores a $.5$
20. $x < -0$ // Inicializa la Variable Independiente x Resultante
21. $x_{tmp} < -0$ // Inicializa la Variable x_{tmp} Resultante
22. $f_x < -0$ // Inicializa la Variable $f(x)$ (Función Diferenciable Continua).
23. $g_x < -0$ // Inicializa la Variable $g(x)$ (Función Diferenciable Continua).
24. $k < -1$ // Inicializa Subíndice para las Variables x , x_{tmp} #Genera Variables $f(x)$, $g(x)$ mediante ciclos anidados para el manejo del plano cartesiano (X_1, X_2, X_{3a}) ú (X_1, X_2, X_{3b})
25. for (i in 1:6)
 - {
 - for (j in 1:6)
 - {
 - for (h in 1:3)
 - {

```

a. x[k]<-x1[i] // Crea la Variable Independiente x
b. xtmp[k]<-x2[2] // Crea la Variable xtmp fx[k]<-
   x1[i]+x2[j]+x3a[h] // Crea la Variable f(x)(Función
   Diferenciable Continua).
c. gx[k]<- x1[i]+x2[j]+x3b[h] // Crea la Variable g(x)(Función
   Diferenciable Continua).
d. k<-k+1 // Incrementa Subíndice
   }
}
}

```

26. #Define tamaño de cada submuestra Resultante

27. n<-108

28. #Genera la Matriz B con las variables x, xtmp

29. B <- matrix(c(x,xtmp),ncol=2)

30. #Inicializa las Variables de Respuesta temporales

31. y1<-0

32. y2<-0

33. muestra=length(x) // Se asigna el tamaño de la submuestra a la variable “muestra”
que se exportara al programa desarrollado en C++.

34. exportData(muestra,file="muestra.fsc",type="FASCII",colNames=F,rowNames=F)
// Exporta el tamaño de la submuestra, generando un archivo ASCII.

35. for (i in 1:1000) // Ciclo para generar la Variable de respuesta agregando un error
aleatorio por cada repetición.

```

a. e<-rnorm(n,0,.3) // Se generan 50 errores aleatorios con Distribución Normal
   Estándar con media cero y Var ( $\epsilon$ ) =  $0,3^2$  por cada repetición.
b. y1<-fx+e // Se crea la variable de respuesta y1, compuesta por la Función f(x)
   teniendo en cuenta un error aleatorio para la muestra n1.
c. #Ejecuta el programa ppreg (Projection Pursuit Regression)
d. saleppr<-ppreg(B,y1,1)
e. a1<-saleppr$alpha[1]
f. a2<-saleppr$alpha[2]

```

- g. #Genera la Variable Independiente U_1 mediante la multiplicación de la Matriz B por la salida del ppreg (Projection Pursuit Regression)
- h. $U_1 <- -B[,1]*a_1 + B[,2]*a_2$
- i. $e <- rnorm(n,0,.3)$ // Se generan 50 errores aleatorios con Distribución Normal Estándar con media cero y $Var(\epsilon) = 0,3^2$ por cada repetición.
- j. $Y_2 <- gx + e$ // Se crea la variable de respuesta Y_2 , compuesta por la Función $g(x)$ agregando un error aleatorio para la muestra n_2 .
- k. #Ejecuta el programa *ppreg* (Projection Pursuit Regression)
- l. $saleppr <- ppreg(B, Y_2, 1)$
- m. $a_3 <- saleppr\$alpha[1]$
- n. $a_4 <- saleppr\$alpha[2]$
- o. #Genera la Variable Independiente U_2 mediante la multiplicación de la Matriz B por la salida del ppreg (Projection Pursuit Regression)
- p. $U_2 <- -B[,1]*a_3 + B[,2]*a_4$
- q. $datos <- data.frame(U_1, U_2, y_1, y_2)$ // Se genera un conjunto de datos que se exportaran al programa desarrollado en C++.
- r. $exportData(datos, file="datos.fsc", type="FASCII",$
 $format="%20.10%20.10%20.10%20.10", colNames=F, rowNames=F)$ // Exporta el Conjunto de Datos (U_1, U_2, y_1, y_2) , generando un archivo ASCII con formato 10 Enteros, 10 Decimales.
- s. $system("newk1005.exe", multi=F, minimized=T, on.exec.status="stop",$
 $on.exit.status="")$
 // Se ejecuta el programa desarrollado en C++.

4.14 PROGRAMA NEWK1005 (Programa en C ++)

Este programa es equivalente al **NEWK1002.CPP**, la diferencia reside en el tamaño de la muestra y en la importación de un conjunto más de datos, la variable independiente U_2 para la muestra número 2.

- **Descripción del Programa NEWK1005**

1. Importa el Conjunto de Datos (U_1, U_2, Y_1, Y_2).
2. Importa el tamaño de la Muestra.
3. Fija 27 valores distintos para el Ancho de Banda (λ).
4. Ordena Ascendentemente los valores de las Variables Independientes U_1, U_2 .
5. Genera las Muestras de tamaño n con las variables independientes $X_{n1}[i], X_{n2}[i]$; y las variables de respuesta $Y_{n1}[i], Y_{n2}[i]$
6. Crea los Limites de la Integral para evaluar la Función de Pesos para cada Muestra.
7. Genera Ciclo (27 Repeticiones) para determinar el Valor Máximo de la Función $\Delta = \hat{\Delta}(\lambda_1, \lambda_2) = \hat{\Delta}(\lambda)$ de acuerdo para algunos valores del ancho de banda (λ)
 - a. Calcula los valores estimados de Y para la Muestra 1.
 - b. Calcula los valores estimados de Y para la Muestra 2.
 - c. Calcula la Matriz de Pesos (Función Kernel Cuadrático).
 - d. Calcula Nuevos Valores de Estimación para la variable de respuesta Y para cada una de las Muestras.
 - e. Calcula la función $\hat{\Delta}(\lambda)$.
8. Exporta Datos para la Hoja Electrónica, donde se almacenan las frecuencias correspondientes a los Valores Máximos de la función $\hat{\Delta}(\lambda)$ asociados al valor de Lambda (λ).
 - a. $\lambda_{max} =$ Valor del Ancho de Banda asociado al valor máximo de la Función $\hat{\Delta}(\lambda)$.

b. $\text{delta_max} = \text{Valor M\u00e1ximo de la Funci\u00f3n } \hat{\Delta}(\lambda)$.

4.15 PROGRAMA NEWK1005_RANDOM.SSC (Procedimiento en Lenguaje S)

Mediante el lenguaje S, se crean tres variables independientes aleatorias X_1 , X_2 y X_3 , para cada variable se generan 6 valores aleatorios uniformemente distribuidos en el intervalo $[0, 1]$. Se generan dos submuestras de Tama\u00f1o 108. Se genera un ciclo en el cual se crea la variable de respuesta Y agregando un error aleatorio. Este procedimiento se realiza 1000 veces; se exportan los datos para que sean tomados por el programa desarrollado en C++ (NEWK1005). Ver descripci\u00f3n del procedimiento en la P\u00e1g. 118.

Su objetivo principal es poder determinar el valor m\u00e1ximo de la Funci\u00f3n $\Delta(\lambda_1, \lambda_2) = \hat{\Delta}(\lambda_1, \lambda_2) = \hat{\Delta}(\lambda)$, para el caso $\lambda_1 = \lambda_2$

Este Procedimiento emplea dos funciones similares para generar los valores observados de la variable de respuesta Y : $f(x)$ y $g(x)$.

- $f(x) = X_1 + X_2 + X_3$ permite calcular los valores observados de Y en la Muestra 1 para valores de X_3 menores o iguales a 0,5.
- $g(x) = X_1 + X_2 + X_3$ permite calcular los valores observados de Y en la Muestra 2 para valores de X_3 mayores a 0,5.

Se puede observar que realmente se emplea una funci\u00f3n $f(x) = X_1 + X_2 + X_3$ para generar los valores de Y .

- **Descripci\u00f3n del Procedimiento**

1. #Genera 6 n\u00fameros aleatorios en X
2. $x < 0$ // Inicializa la Variable x
3. $x < -c(\text{runif}(1,0,.5), \text{runif}(1,0,.5), \text{runif}(1,0,.5),$
 $\text{runif}(1,.51,1), \text{runif}(1,.51,1), \text{runif}(1,.51,1))$
 - a. // Se crea la variable X con 6 valores aleatorios entre 0 y 1.
4. # Subconjunto Z de la Variable Independiente $X_3 \Rightarrow$

```

a. # X3a -> [0,1] para la muestra n1
5. # Criterio X3a <= .5
6. X3a<-0 // Inicializa Subconjunto Z de la Variable Independiente X3
7. X3a <-sort(x[x <= .5]) // Crea el Subconjunto Z de la Variable Independiente X3
    para aquellos valores menores o iguales a .5
8. # Subconjunto ZC(Complemento de Z) de la Variable Independiente X3 => # X3b -
    > [0,1] para la muestra n2
9. # Criterio X3b > .5
10. X3b<-0 // Inicializa Subconjunto ZC de la Variable Independiente X3
11. X3b <-sort(x[x > .5]) // Crea el Subconjunto ZC de la Variable Independiente X3 para
    aquellos valores mayores a .5
12. # Valores de Variable Independiente X3 -> [0,1]
13. X3<-0 // Inicializa la Variable Independiente X3
14. X3<-c(X3a,X3b) // Asigna los valores de los Subconjuntos de datos a la Variable
    Independiente X3
15. # Valores de la Variable Independiente X1 -> [0,1]
16. X1<-0 // Inicializa la Variable Independiente X1
17. X1<-c(sort(runif(6,0,1)))// Se crea la variable independiente con valores aleatorios
    entre 0 y 1
18. # Valores de la Variable Independiente X2 -> [0,1]
19. X2<-0 // Inicializa la Variable Independiente X2
20. X2<-c(sort(runif(6,0,1)))// Se crea la variable independiente con valores aleatorios
    entre 0 y 1
21. x<-0 // Inicializa la Variable Independiente x Resultante
22. xtmp<-0 // Inicializa la Variable xtmp resultante
23. fx<-0 // Inicializa la Variable f(x) (Función Diferenciable Continua).
24. gx<-0 // Inicializa la Variable g(x) (Función Diferenciable Continua).
25. k<-1 // Inicializa Subíndice para las Variables x, xtmp
26. #Genera las Variables f(x) ,g(x) mediante ciclos anidados para el manejo del plano
    cartesiano (X1,X2,X3a) ú (X1,X2,X3b)
27. for (i in 1:6)
    {

```



```

for (j in 1:6)
{
  for (h in 1:3)
  {
    a. x[k]<-X1[i] // Crea la Variable Independiente x
    b. xtmp[k]<-X2[j] // Crea la Variable xtmp
    c. fx[k]<-X1[i]+X2[j]+X3a[h] // Crea la Variable
      f(x)(Función Diferenciable Continua).
    d. gx[k]<- X1[i]+X2[j]+X3b[h] // Crea la Variable
      g(x)(Función Diferenciable Continua).
    e. k<-k+1 // Incrementa Subíndice
  }
}
}

```

28. #Define el tamaño de cada submuestra Resultante

29. n<-108

30. #Genera la Matriz B con las variables x, xtmp

31. B <- matrix(c(x,xtmp),ncol=2)

32. #Inicializa Variables de Respuesta temporales

33. y₁<-0

34. y₂<-0

35. muestra=length(x) // Se asigna el tamaño de la submuestra a la variable “muestra” que se exportara al programa desarrollado en C++.

36. exportData(muestra,file="muestra.fsc",type="FASCII", colNames=F,rowNames=F)
// Exporta el tamaño de la submuestra, generando un archivo ASCII.

37. for (i in 1:1000) // Ciclo para generar la Variable de respuesta agregando un Error Aleatorio por cada repetición.

- e<-rnorm(n,0,.3) // Se generan 50 errores aleatorios con Distribución Normal Estándar con media cero y Var (ϵ) = $0,3^2$ por cada repetición.
- y₁<-fx+e // Se crea la variable de respuesta y₁, compuesta por la Función f(x) teniendo en cuenta un error aleatorio para la muestra n₁.
- #Ejecuta el programa *ppreg* (Projection Pursuit Regression)

```

d. saleppr<-ppreg(B,y1,1)
e. a1<-saleppr$alpha[1]
f. a2<-saleppr$alpha[2]
g. #Genera la Variable Independiente U1 mediante la multiplicación de la
    Matriz B por la salida del ppreg (Projection Pursuit Regression)
h. U1<-B[,1]*a1+B[,2]*a2
i. e<-rnorm(n,0,.3) // Se generan 50 errores aleatorios con Distribución Normal
    Estándar con media cero y Var (ε) = 0,32 por cada repetición.
j. y2<-gx+e // Se crea la variable de respuesta y2, compuesta por la Función
    g(x) agregando un error aleatorio para la muestra n2.
k. #Ejecuta el programa ppreg (Projection Pursuit Regression)
l. saleppr<-ppreg(B,y2,1)
m. a3<-saleppr$alpha[1]
n. a4<-saleppr$alpha[2]
o. #Genera la Variable Independiente U2 mediante la multiplicación de la
    Matriz B por la salida del ppreg (Projection Pursuit Regression)
p. U2<-B[,1]*a3+B[,2]*a4
q. datos<- data.frame(U1,U2,y1,y2) // Se genera un conjunto de datos que se
    exportaran al programa desarrollado en C++.
r. exportData(datos,file="datos.fsc",type="FASCII",
    format="%20.10%20.10%20.10%20.10",colNames=F, rowNames=F) //
    Exporta el Conjunto de Datos (U1,U2,y1,y2), generando un archivo ASCII
    con formato 10 Enteros, 10 Decimales.
s. system("newk1005.exe",multi=F,minimized=T, on.exec.status="stop",
    on.exit.status="")
// Se ejecuta el programa desarrollado en C++.

```

Figura 20. DIAGRAMA DE FLUJO PROGRAMAS NEWK -NEWK1.SSC

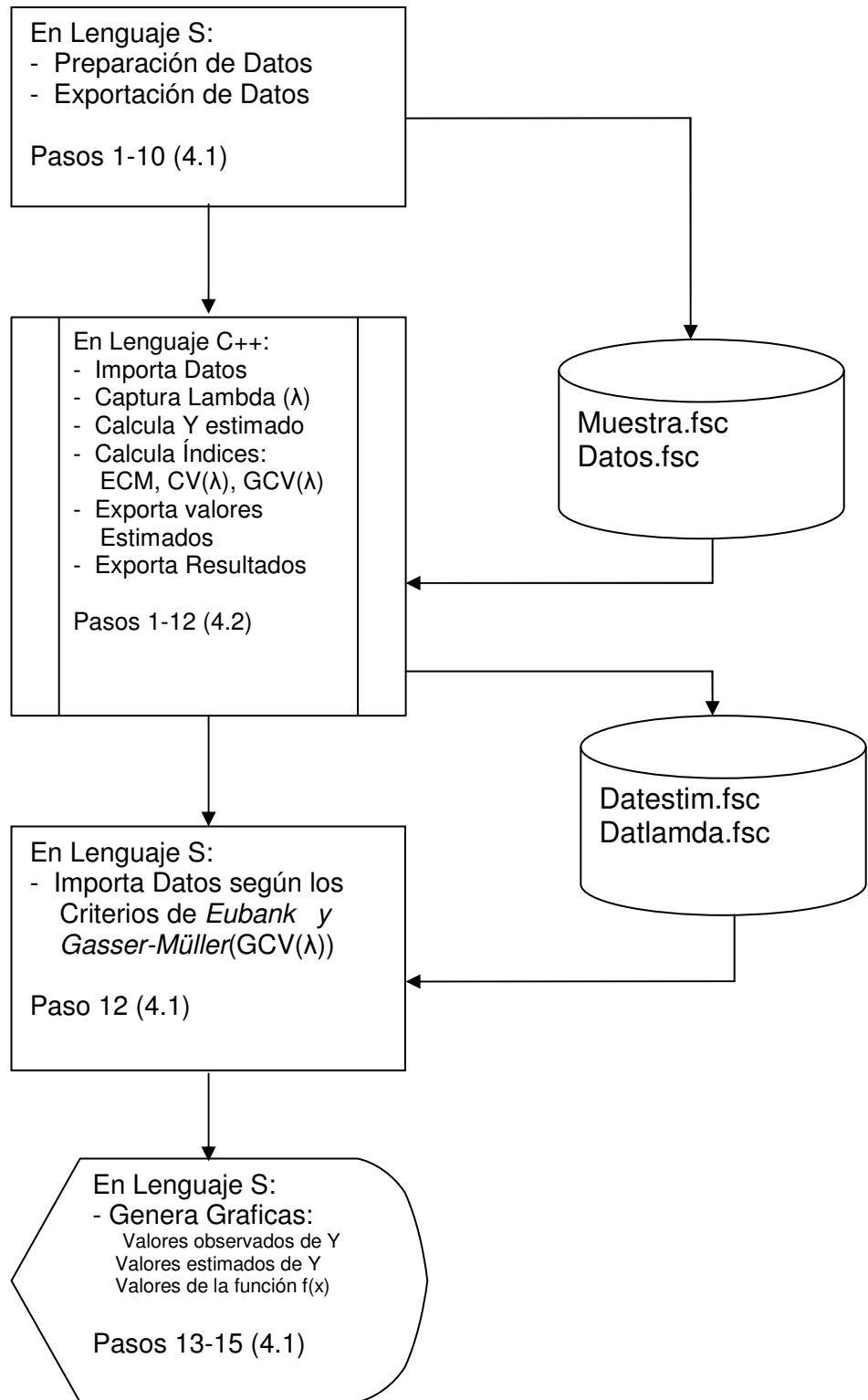


Figura 21. DIAGRAMA DE FLUJO PROGRAMAS NEWK1000 - NEWK1000.SSC

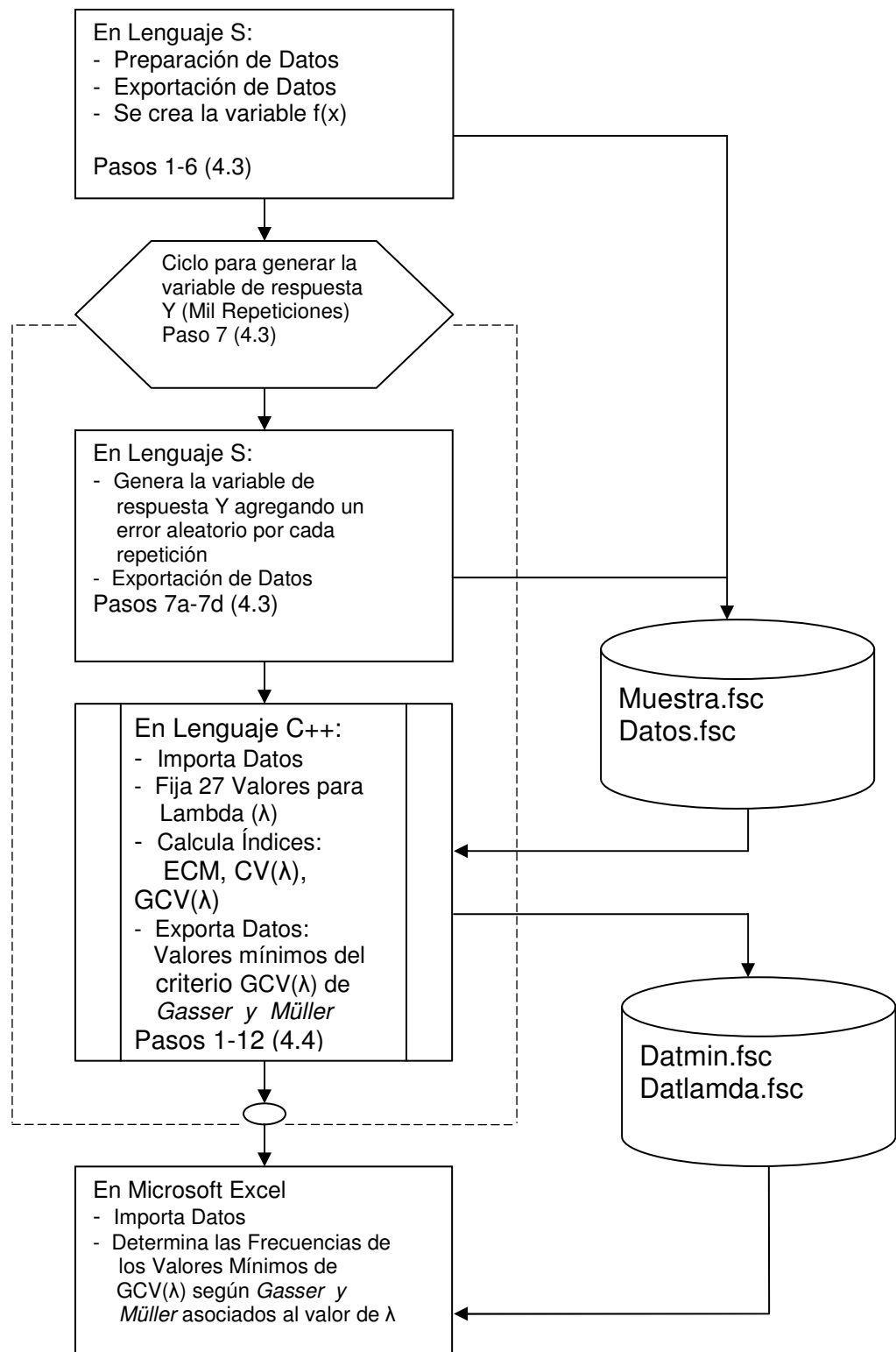


Figura 22. DIAGRAMA DE FLUJO PROGRAMAS NEWK1001 - NEWK1001.SSC

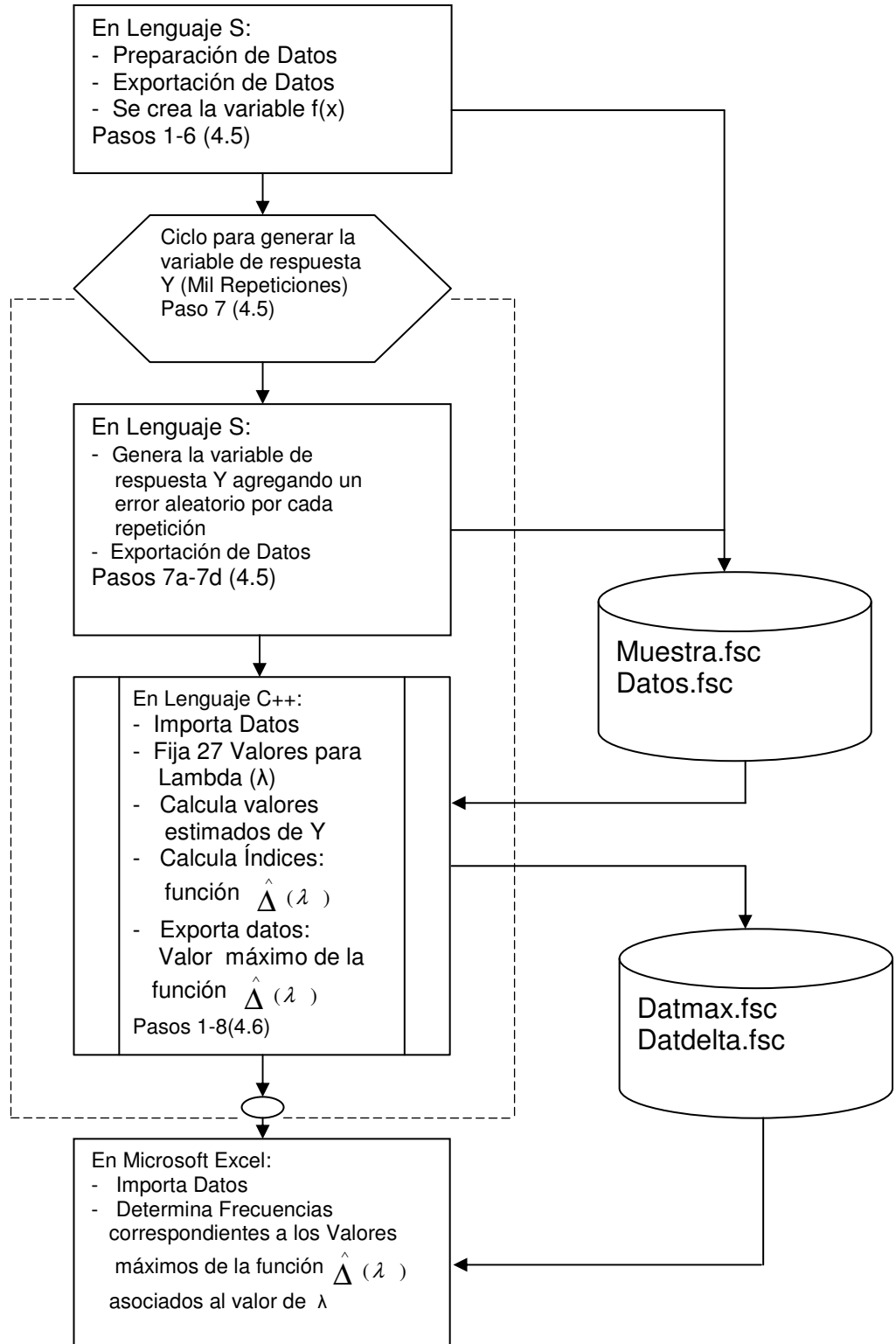


Figura 23. DIAGRAMA DE FLUJO PROGRAMAS NEWK1002 - NEWK1002.SSC

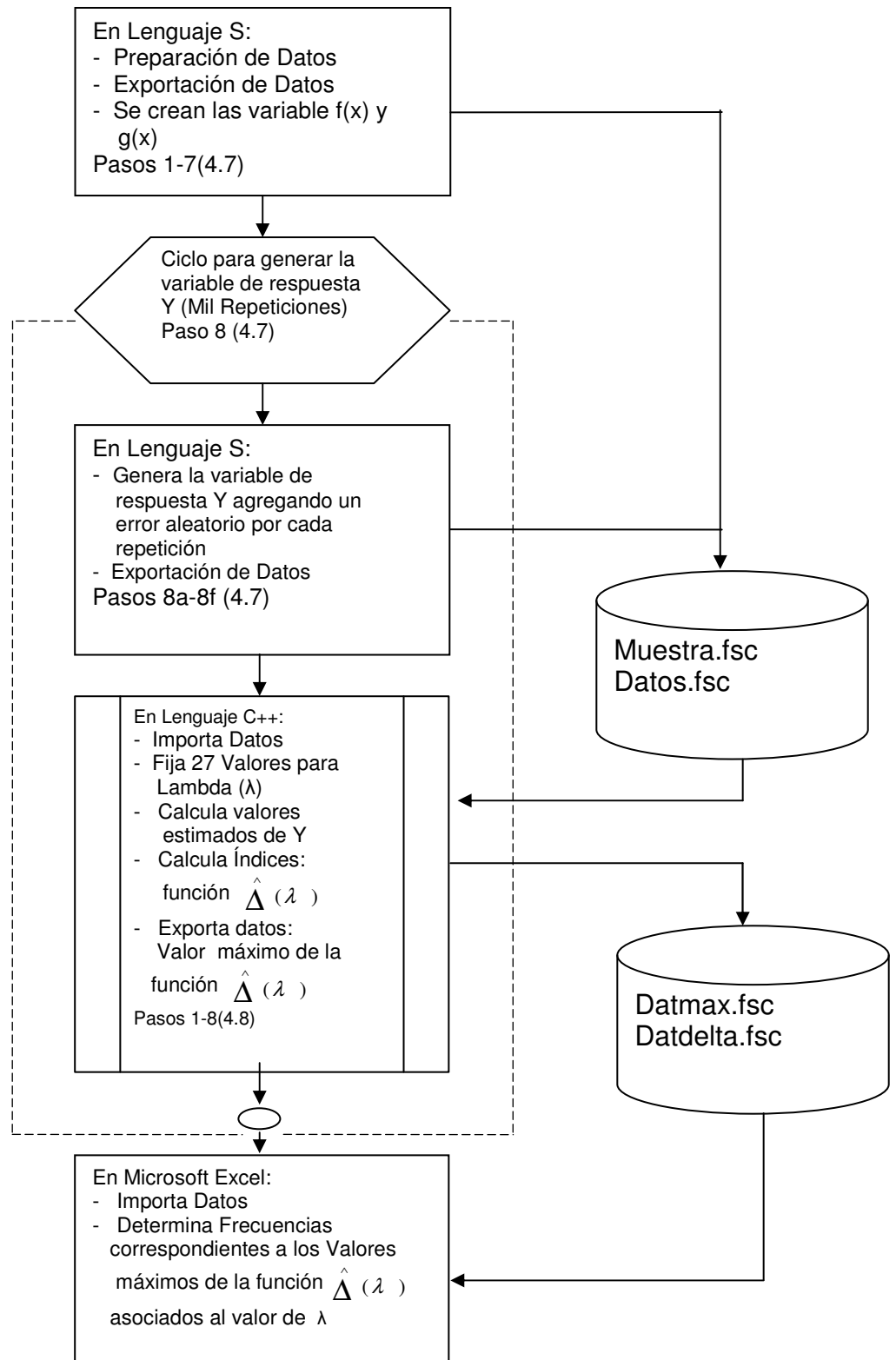


Figura 24. DIAGRAMA DE FLUJO PROGRAMAS NEWK1003 - NEWK1003.SSC

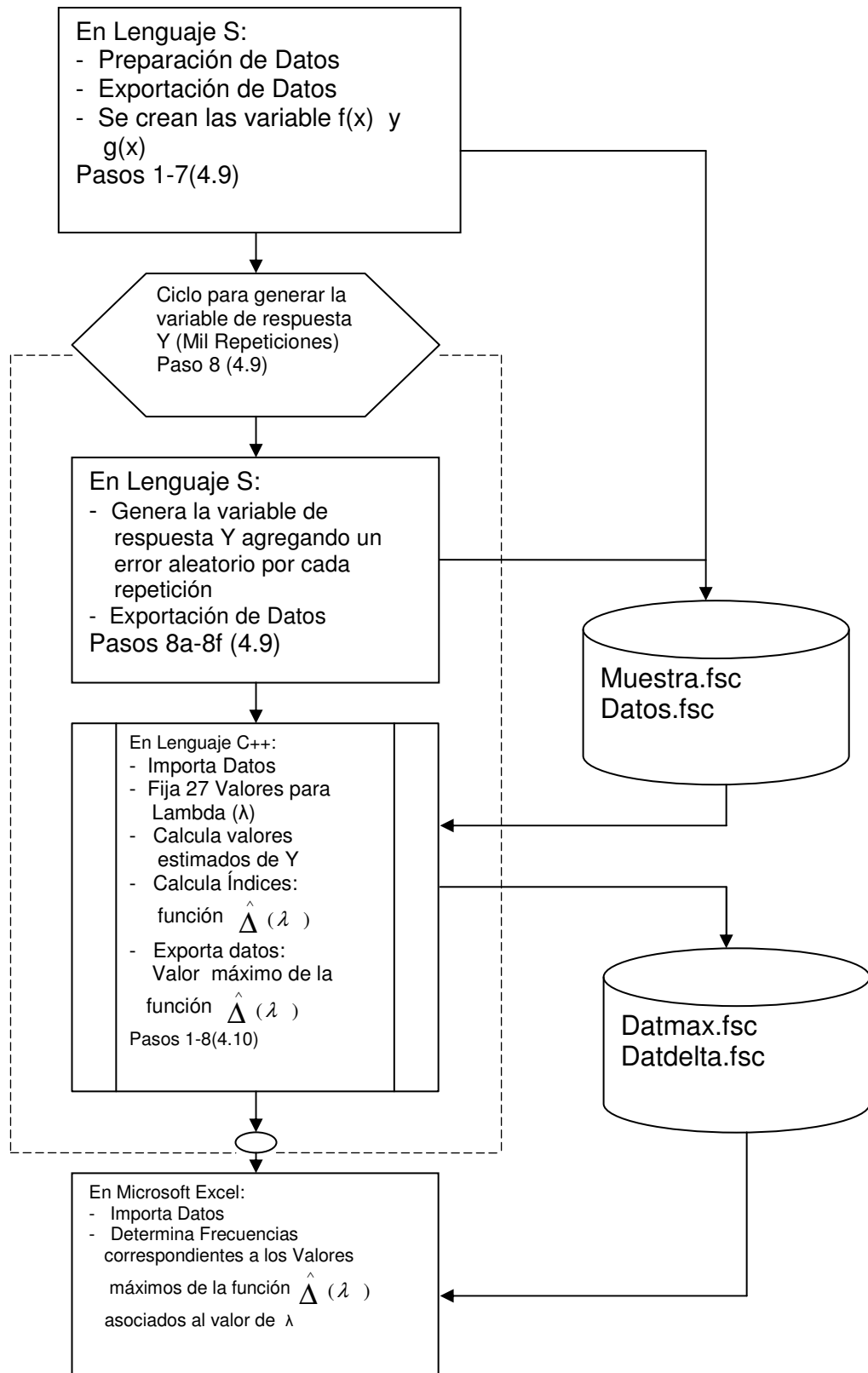


Figura 25. DIAGRAMA DE FLUJO PROGRAMAS NEWK1002 - NEWK1004.SSC

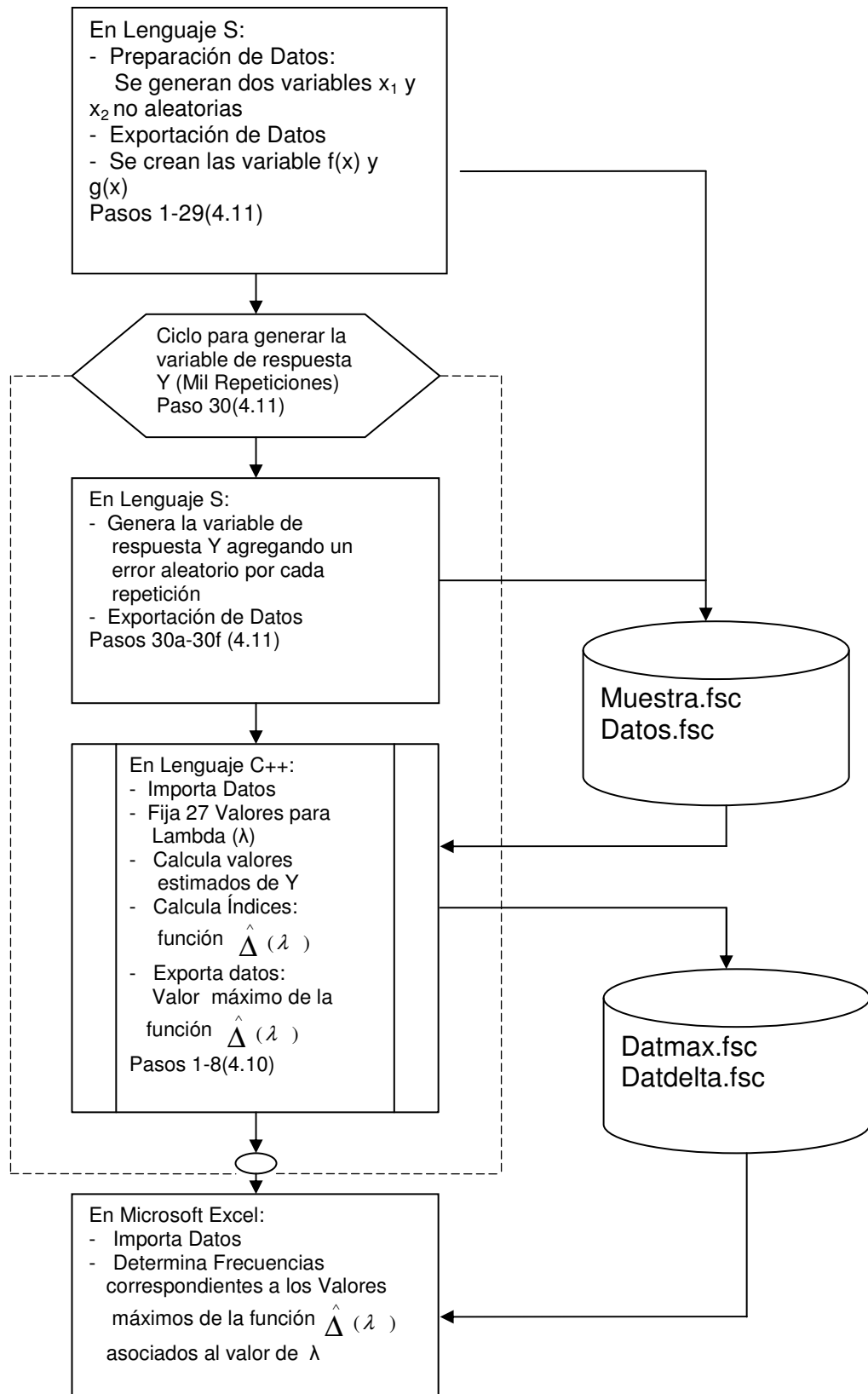


Figura 26. DIAGRAMA DE FLUJO PROGRAMAS NEWK1002 - NEWK1004_RANDOM.SSC

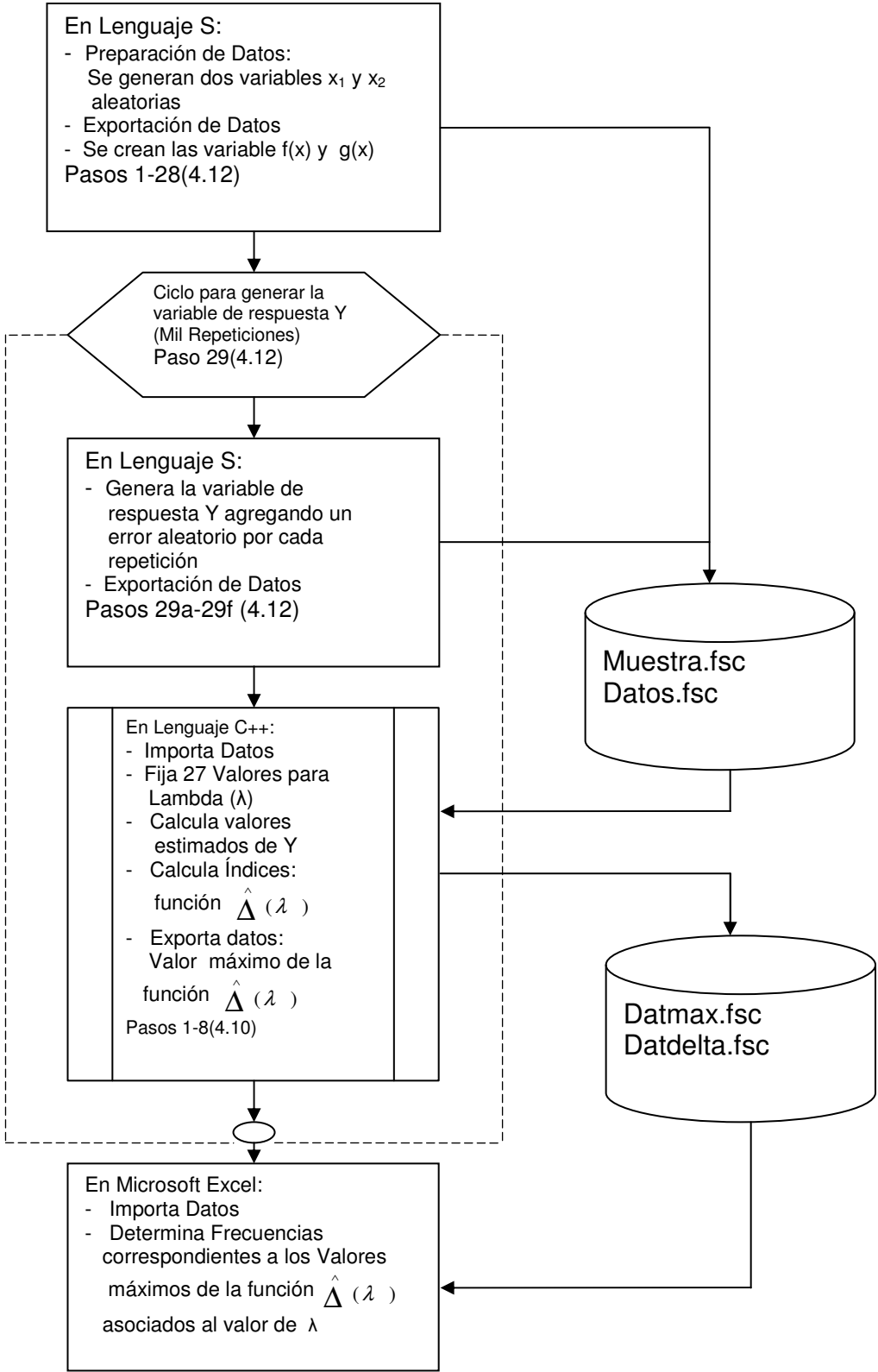


Figura 27. DIAGRAMA DE FLUJO PROGRAMAS NEWK1005 - NEWK1005.SSC

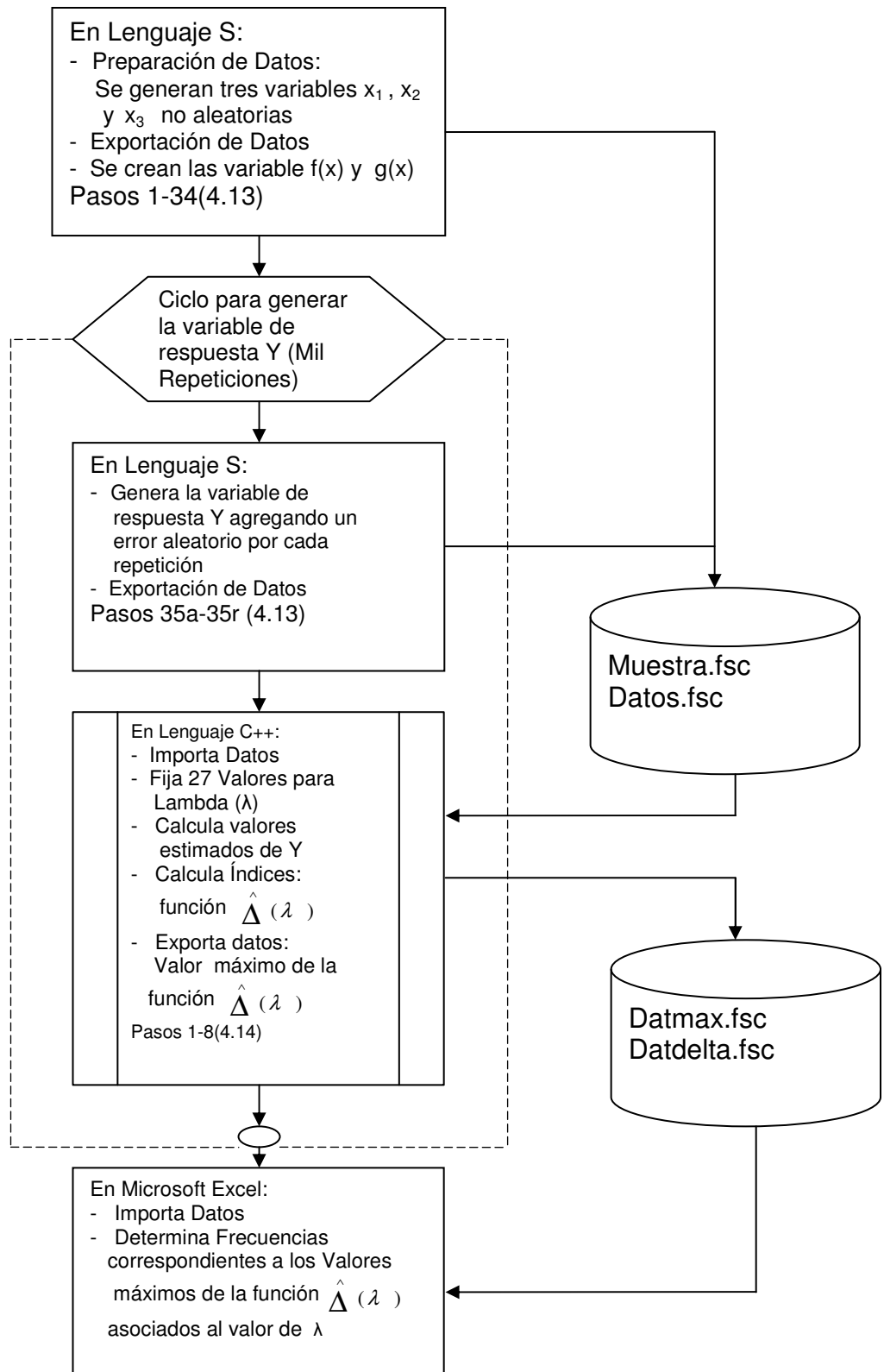
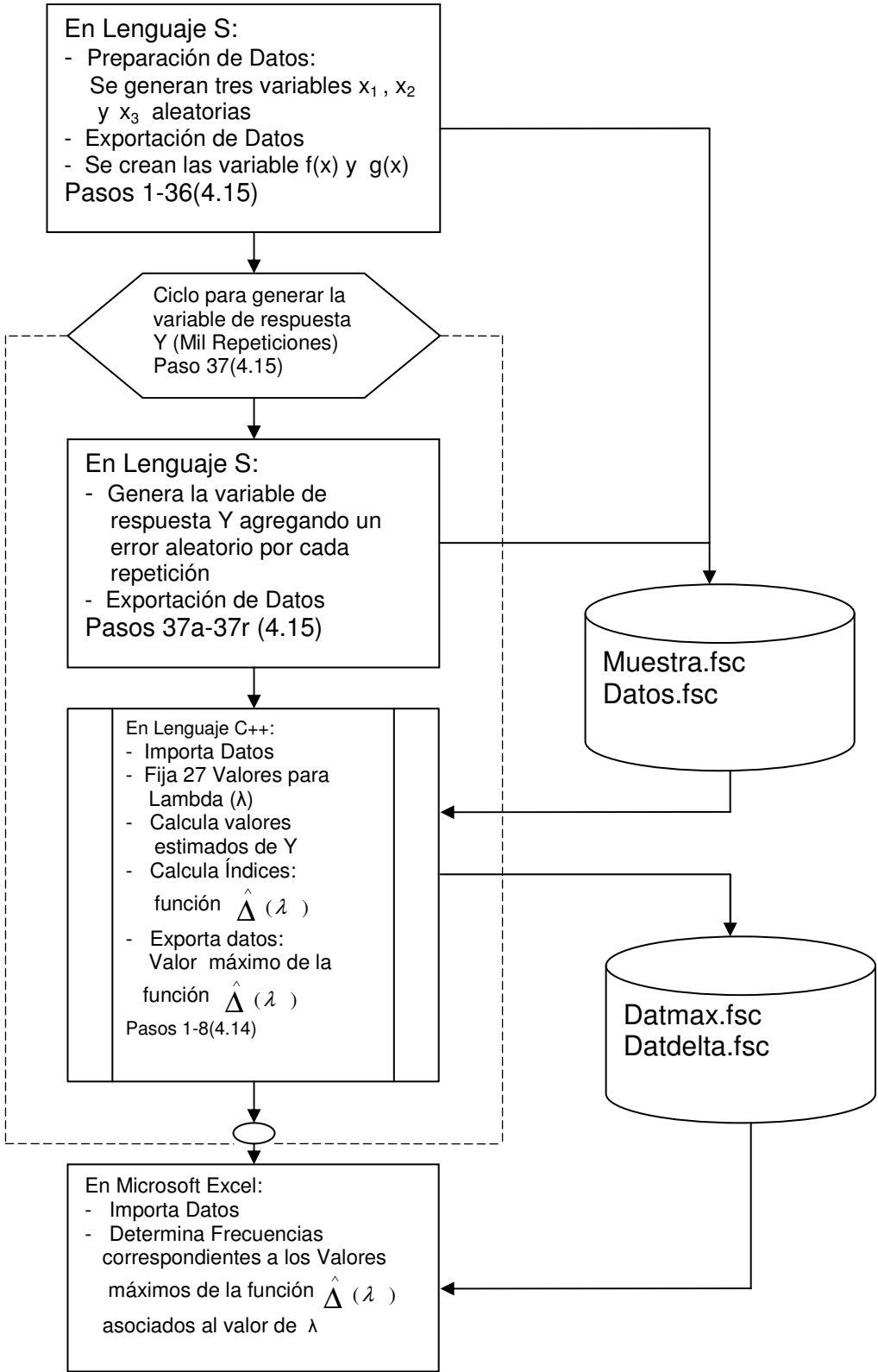


Figura 28. DIAGRAMA DE FLUJO PROGRAMAS NEWK1005 - NEWK1005_RANDOM.SSC



5. CONCLUSIONES

Se propone en este trabajo una solución numérica que permite encontrar el valor óptimo del parámetro de suavización en el problema de la selección de variables en los modelos de regresión no-paramétricos. Siguiendo la literatura, se utilizó como criterio de optimización la maximización de la potencia de una prueba estadística de comparación de curvas de regresión.

Para la optimización se utilizó el método de *Kulasekera y Wang (1997)* aplicado al método de selección de variables en regresión no-paramétrica propuesto por *Olaya (2000)*. Para efectos comparativos se simularon varias respuestas posibles basadas en funciones de regresión adecuadamente escogidas y se diseñó una herramienta computacional que permite seleccionar automáticamente el valor del parámetro de suavización óptimo en el problema de selección de variables en regresión *no-paramétrica*. En todos los casos se utilizan estimadores *Kernel* para estimar la función de regresión $\mu(x)$.

Algunos de los resultados más destacados son:

- Cuando se utiliza como función de regresión la función $f(x) = X_1 + X_2$, el valor óptimo del parámetro de suavización resulta ser 0,2, ya sea que el diseño de puntos en las covariables sea o no aleatorio.
- Para el caso de la función de regresión $f(x) = X_1 + X_2 + X_3$, se encontró un valor de 0,3 para el parámetro de suavización para ambos tipos de diseños, aleatorio y no aleatorio, de puntos en las covariables.
- Para afinar el método se compararon curvas teóricas de regresión en una sola variable. En particular se utilizaron las funciones $f(x) = \sin(4x)e^x + 3$ y $g(x) = \cos(4x)$, las cuales se estimaron inicialmente en dos conjuntos de igual

tamaño y posteriormente en conjuntos de tamaño diferente. En ambos casos el valor elegido es 0,2 para el parámetro de suavización.

El método propuesto parece muy promisorio, en cuanto provee una solución no disponible en la literatura para problemas similares, tal como se ilustra con las dos funciones reportadas.

En el Capítulo 1 se estudiaron los modelos de regresión no-paramétricos. Se estudió el problema de selección de variables y se discutieron algunos aspectos teóricos del método para seleccionar el valor óptimo del parámetro de suavización (ancho de banda) en el problema de selección de variables en regresión *no-paramétrica* en el Capítulo 2.

Se estudió la prueba de comparación de dos funciones de regresión en varios casos. La metodología propuesta puede ser aplicada con diversos diseños de puntos (igual diseño de puntos o diferentes diseños de puntos en las covariables para las dos submuestras (n_1 y n_2) y estructuras de errores con cualquier distribución. El método de escogencia del valor óptimo del parámetro de suavización permite obtener la potencia óptima en pruebas de comparación de curvas de regresión.

La función $\Delta(\lambda_1, \lambda_2)$ (2.30) que hemos estudiado, no solamente se utiliza para optimizar la potencia de una prueba, es confiable para obtener un valor apropiado del parámetro de suavización basado en los datos de la muestra. Sin embargo, este procedimiento requiere que el parámetro de suavización λ , presente algunas propiedades asintóticas de las pruebas que involucran las funciones de regresión $\mu_Z(X)$ y $\mu_{Z^c}(X)$. El método propuesto de *Kulasekera y Wang* (1997) permite estimar la potencia para una prueba dada. Este método da posibilidad en definir una función $\hat{\Delta}(\lambda)$ (2.35) que puede ser maximizada con respecto al parámetro λ . En nuestro caso y en el de *Kulasekera y Wang* (1997), la idea central fue encontrar un valor óptimo del parámetro de suavización λ el cual permite maximizar la potencia por medio de la función $\hat{\Delta}(\lambda)$ (2.35).

Aunque hemos utilizado exclusivamente en nuestro estudio de simulación el método *Kernel* (estimador lineal de *Gasser y Müller* (1979) (1.7)) en la estimación de las funciones de regresión $\mu_Z(X)$ y $\mu_{Z^c}(X)$ que fue examinado en el Capítulo 1, se puede utilizar estimadores equivalentes con propiedades asintóticas, tales como el método de suavización local lineal (Fan 1992). Además, algunas modificaciones apropiadas en los supuestos permiten el uso de estas pruebas en diseños aleatorios.

Una posible extensión de este trabajo sería utilizar la herramienta diseñada para su uso en otros problemas similares, por ejemplo en los métodos de selección propuestos por *Kulasekera* (1995) y en los problemas de comparación de curvas de *King, Hall* y *Wehrly* (1991)

BIBLIOGRAFIA

BENEDETTI, G. (1975). *Kernel estimation of regression functions*. Proceedings in Computer Science and Statistics: 8 th annual symposium on the interface, pages 405-412.

BROCKMANN, M. , GASSER, T. , y HERMANN, E. (1993). *Locally Adaptive Bandwidth Choise for Kernel Regression Estimation*, Journal of the American Statistical Association, 88, 1302-1309.

DRAPER, N. R. y SMITH, H. (1966). *Applied Regression Analysis*. John Wiley & Sons, New York, NY.

DRAPER, N. R. y SMITH, H. (1981). *Applied Regression Analysis*. John Wiley & Sons, New York, NY, second edition.

DRAPER, N. R. y SMITH, H. (1998). *Applied Regression Analysis*. John Wiley & Sons, New York, NY, third edition.

EUBANK, R. L (1999). *Nonparametric Regression and Spline Smoothing*. Segunda edición. Marcel Dekker, Inc. New York.

FAN, J. (1992). *Design-Adaptive Nonparametric Regression*. Journal of the American Statistical Association, 87, 998 -1004.

FRIEDMAN, J. H. y TUKEY, J. W. (1974). *A Projection Pursuit Algorithm for Exploratory Data Analysis*. *IEEE Trans. Comput.* C-23 881-889.

GASSER, T. y MÜLLER, H. G. (1979). *Smoothing Techniques for Curve Estimation*. Springer, Heidelberg.

GRAYBILL, F. A. (1976). *Theory and Application of the Linear Model*. Wadsworth & Brooks, Pacific Grove, California.

HALL, PETER, KAY, J. W. y TITTERINGTON, D. M. (1990). *Asymptotically Optimal Difference-Based Estimation of Variance in Nonparametric Regression* Department of Statistics, Australian National University, Canberra, ACT 2601, Australia. *Biometrika*, 77, 3, pp.521-8.

HALL, P. y HART, J. D. (1990). *Bootstrap test for difference between means in nonparametric regression*. *Journal of the American Statistical Association*, 85, 1085-1093.

HALL, P. , MARRON, J. S. , y PARK, B. U. (1992). *Smoothed Cross-Validation*, *Probability Theory and Related Fields*, 92, 1-20.

HÄRDLE, W. (1990). *Applied Nonparametric Regression*. Cambridge University Press, Cambridge, UK.

HÄRDLE, W., HALL, P., y ICHIMURA, H. (1993). *Optimal Smoothing in single-index Models*. *The Annals of Statistics*.

HÄRDLE, WOLFGANG y TSYBAKOV, A. B. (1995). *Additive nonparametric regression on principal components*. *nonparametrics Statistics*, 5:157-184.

HÄRDLE, W., y KOROSTELEV, A. (1996). *Search for Significant Variables in Nonparametric Additive Regression*. *Biometrika*. 541-549.

KING, E. C. HÄRT, J. D, and WEHRLY, T. E (1991). *Testing the Equality of Two Regression Curves Using Linear Smoothers*. *Statistics and Probability Letters*, 12. 239-247.

JEROME, H. FRIEDMAN y WERNER STUETZLE (1981). *Projection Pursuit Regression*. Article December. Volume 76, No. 376. American Statistical Association.

KOTZ, S. , JOHNSON, N. L. , y READ, C. B. (1988). *Encyclopedia of Statistical Science*, Volume VIII. John Wiley, New York.

KULASEKERA, K.B. (1995). *Comparison of Regression Curves Using Quasi-Residuals* . Article September, Volume 90, No. 431. American Statistical Association.

KULASEKERA, K.B. y WANG, J. (1997). *Smoothing Parameter Selection for Power Optimality in Testing of Regression Curves*. Article June, Volume 92, No. 438. American Statistical Association.

LI, K. C. (1991). *Sliced Inverse Regression for Dimension Reduction*. Journal of the American Statistical Association, 86, 316-327.

NADARAYA, E. A. A. y SECKLER, B. T. (1964). *On estimating regression*. Theory of Probability and its Applications.

OLAYA, J. (2000). *Selecting Variables in Nonparametric Regression*. Universidad delValle. Artículo November 26.

OLAYA, J. y KULASEKERA, K.B. (2000). *A method for selecting variables in nonparametric regression*. Dissertation Thesis, Clemson University, Clemson, SC.

TREVOR, J. , HASTIE y TIBSHIRANI, ROBERT J. (1990). *General Additive Models*. Chapman & Hall-CRC, Boca Raton, FL.

WATSON, G. (1964). *Smooth Regresión Análisis*. Sankhya, Series A.

ZHANG, P. (1990). Variable Selection in nonparametric regression with continuous covariates. The Annals of Statistics.