



COLEGIO DE POSTGRADUADOS

---

INSTITUCIÓN DE ENSEÑANZA E INVESTIGACIÓN  
EN CIENCIAS AGRÍCOLAS

**CAMPUS MONTECILLO**

POSTGRADO EN SOCIOECONOMÍA-ESTADÍSTICA E  
INFORMÁTICA-ESTADÍSTICA

**Estimación Semiparamétrica en el Análisis de  
encuestas complejas**

Luis Fernando Contreras Cruz

T E S I S

PRESENTADA COMO REQUISITO PARCIAL PARA  
OBTENER EL GRADO DE:

DOCTOR EN CIENCIAS

MONTECILLO, TEXCOCO, EDO. DE MÉXICO  
2013

---

La presente tesis titulada: **Estimación Semiparamétrica en el Análisis de encuestas complejas**, realizada por el alumno: **Luis Fernando Contreras Cruz**, bajo la dirección del Consejo Particular indicado ha sido aprobada por el mismo y aceptada como requisito parcial para obtener el grado de:

**DOCTOR EN CIENCIAS**

**SOCIOECONOMÍA, ESTADÍSTICA E INFORMÁTICA  
ESTADÍSTICA**

**CONSEJO PARTICULAR**

CONSEJERO

Dr. José A. Villaseñor Alva

DIRECTOR DE TESIS

Dr. José Elías Rodríguez Muñoz

ASESOR

Dr. Javier Suárez Espinosa

ASESOR

Dr. Ignacio Méndez Ramírez

ASESOR

Dr. Gilberto Rendón Sánchez

# Estimación Semiparamétrica en el Análisis de encuestas complejas

Luis Fernando Contreras Cruz

Colegio de Postgraduados, 2013

En la primera parte de este trabajo se muestra que no hay pérdida significativa en la precisión del estimador del total poblacional si calculamos el parámetro de suavizamiento del modelo de splines penalizados con información de la muestra en vez de utilizar toda la información de la población. El parámetro de suavizamiento es obtenido fijando los grados de libertad y resolviendo una ecuación basada únicamente en datos de la muestra. Se discuten los resultados de un estudio de simulación de la determinación del parámetro de suavizamiento. Además, se evaluó el efecto de la propuesta de determinación del parámetro de suavizamiento en la estimación del total poblacional usando una población real. Para esta población, los experimentos de simulación muestran que la determinación del parámetro de suavizamiento, ya sea con datos de la muestra o con los datos de toda la población, produce errores relativos similares en la estimación del total poblacional. En la segunda parte, se presenta y desarrolla un método semiparamétrico de estimación del total poblacional para mejorar la precisión de los estimadores mediante la incorporación de información auxiliar multivariada. El modelo de superpoblación utilizado es un modelo de coeficientes variantes, que ha demostrado ser una herramienta semiparamétrica simple y eficiente en la regresión multivariada. En condiciones de diseño estándar, los estimadores propuestos son asintóticamente insesgados, consistente y asintóticamente normales. Para ilustrar el modelo y la metodología de estimación propuesta, se estudian dos poblaciones, una población real y otra simulada, que han proporcionado fuerte evidencia que corrobora la teoría asintótica.

**Palabras clave:** Parámetro de suavizamiento, regresión semiparamétrica, modelo de coeficientes variantes, muestreo, información auxiliar multivariada.

# Semiparametric estimation in complex survey analysis

Luis Fernando Contreras Cruz

Colegio de Postgraduados, 2013

In the first part of this work, the aim is to show that there is no significant loss in accuracy of the estimator of the population total if we calculate the smoothing parameter of penalized spline model with sample information rather than using all the population information. The smoothing parameter is obtained by fixing the degrees of freedom and solving an equation based only on sample data. Results of a simulation study of the determination of smoothing parameter are presented. In addition, the effect of the proposed determination of smoothing parameter on the estimation of the population total is evaluated using a real population. For this population, simulation experiments show that determining the smoothing parameter with either sample data or with data from the entire population, produces similar relative errors on the estimate of the population total. In the second part, a model-assisted semiparametric method of estimating finite-population totals is investigated to improve the precision of survey estimators by incorporating multivariate auxiliary information. The proposed superpopulation model is a varying-coefficient model which has proven to be a simple and efficient semiparametric tool in multivariate regression. Under standard design conditions, the proposed estimators are asymptotically design-unbiased, consistent and asymptotically normal. Both simulated and real data examples are given to illustrate the model and the proposed estimation methodology, which have provided strong evidence that corroborates with the asymptotic theory.

**Key words:** Smoothing parameter, semiparametric regression, varying coefficient model, sampling, multivariate auxiliary information.

## AGRADECIMIENTOS

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico brindado durante la realización de mis estudios.

Al Colegio de Postgraduados, por haberme brindado la oportunidad de seguir mi formación académica en sus aulas.

A los integrantes de mi Consejo Particular:

Al Dr. José Elías Rodríguez Muñoz mi más sincero agradecimiento por ser un excelente guía y amigo, por sus atinadas indicaciones y consejos, así como su calidad humana, mismos que me fueron de gran utilidad en la realización del presente trabajo de tesis.

Al Dr. José A. Villaseñor Alva, por ser un guía durante mis estudios y por sus consejos tan atinados y por dedicar parte de su tiempo en la revisión de este trabajo de tesis.

A los doctores: Javier Suárez Espinosa, Gilberto Rendón Sánchez e Ignacio Méndez Ramírez por sus colaboraciones desinteresada en el presente trabajo.

Un agradecimiento muy especial al Dr. F. Jay Breidt, del Departamento de Estadística de la Universidad Estatal de Colorado E.U.A., por permitirme trabajar con él y por dedicar parte de su tiempo en la discusión y desarrollo de algunos temas de este trabajo.

A la familia Cárdenas Ramírez y en especial a la Familia Gaona Cárdenas por el apoyo recibido durante mi estancia en Estados Unidos.

A mis compañeros de clase: Raúl A. Pérez Agámez, Juan Diego Hernández Jarquín y Jaime Dionisio Cuevas Domínguez, por los momentos de discusión de algunos temas en estadística.

A todas aquellas personas que de alguna manera fueron coparticipes de esta tarea, a todos muchas gracias.

## DEDICATORIA

A mis padres: Fernando Contreras Avalos y Bélgica Cruz López, que me han enseñado que la única forma de concluir satisfactoriamente una meta es trabajar día a día.

A mis hermanos: Carlos Arturo Contreras Cruz y Marcos Contreras Cruz.

A mis abuelos: Ana María Avalos Jesús<sup>†</sup>, Rosalío Contreras García<sup>†</sup>, Catalina López Damían<sup>†</sup> y Antonio Cruz Gómez<sup>†</sup>.

En especial a mi esposa Ma. Esperanza Blanco Ramírez por animarme siempre y enseñarme a ver la vida de otra manera.

# Índice

<b>1. Introducción</b>	<b>1</b>
<b>2. Objetivos</b>	<b>5</b>
2.1. Objetivos Generales . . . . .	5
2.2. Objetivos Particulares . . . . .	5
<b>3. Muestreo de poblaciones finitas</b>	<b>6</b>
3.1. Población, muestra y selección de muestras . . . . .	6
3.2. Diseño de muestreo . . . . .	7
3.3. Probabilidades de inclusión . . . . .	8
3.4. Parámetros poblacionales y sus estimadores . . . . .	11
3.5. Distribución aproximada de un estimador . . . . .	12
3.6. Teoría asintótica en muestreo de poblaciones finitas . . . . .	13
3.7. Estimador de Horvitz-Thompson . . . . .	16
3.8. Muestreo para unidades poblacionales . . . . .	17
3.8.1. Muestreo Bernoulli . . . . .	17
3.8.2. Muestreo aleatorio simple sin reemplazo . . . . .	19
3.8.3. Muestreo con probabilidad proporcional al tamaño . . . . .	21

3.8.4. Muestreo estratificado . . . . .	24
3.8.5. Muestreo de conglomerados en una sola etapa . . . . .	26
3.9. Muestreo de poblaciones finitas asistido por modelos . . . . .	31
3.9.1. Estimación en la población finita . . . . .	32
3.9.2. Estimación en la muestra . . . . .	32
3.9.3. Estimador de regresión general . . . . .	33
<b>4. Regresión semiparamétrica en muestreo de poblaciones finitas</b>	<b>37</b>
4.1. Introducción . . . . .	37
4.2. Regresión local polinomial . . . . .	38
4.3. Regresión por Splines Penalizados . . . . .	41
4.4. Estudio de Simulación . . . . .	43
4.5. Conclusiones . . . . .	45
<b>5. El efecto sobre la estimación del total poblacional de la estimación basada en la muestra del parámetro de suavizamiento de regresión por spline penalizado en muestreo de poblaciones finitas</b>	<b>47</b>
5.1. Introducción . . . . .	47
5.2. Selección del parámetro de suavizamiento basada en la muestra . . . . .	49
5.3. Estudio de simulación . . . . .	51
5.3.1. Algoritmo . . . . .	51
5.3.2. Resultados empíricos . . . . .	53
5.4. Aplicación a datos reales . . . . .	55
5.5. Conclusiones . . . . .	57



<b>6. Modelo de coeficientes variantes en muestreo de poblaciones finitas</b>	<b>59</b>
6.1. Modelo de coeficientes variantes en muestreo de poblaciones finitas . . . . .	60
6.2. Selección de los parámetros de suavizamiento . . . . .	63
6.3. Propiedades del estimador del total poblacional . . . . .	66
6.4. Estudio de simulación . . . . .	67
6.5. Aplicación a datos reales . . . . .	71
6.6. Conclusiones . . . . .	72
<b>7. Conclusiones y trabajos futuros</b>	<b>73</b>
<b>Referencias</b>	<b>74</b>
<b>Anexos</b>	<b>79</b>
Anexo A: Códigos de Programas en R usados en el Capítulo 4 . . . . .	79
Anexo B: Códigos de Programas en R usados en el Capítulo 5 . . . . .	83
Anexo C: Códigos de Programas en R usados en el Capítulo 6 . . . . .	94
Anexo D: Demostración del Teorema 1 del Capítulo 6 . . . . .	133
Anexo E: Demostración del Teorema 2 del Capítulo 6 . . . . .	135
Anexo F: Demostración del Corolario 1 del Capítulo 6 . . . . .	137

# Índice de tablas

4.1. Razón de ECM relativo al estimador de spline penalizado, basado en 1000 réplicas de tamaños de muestra $n = 50$ de una población fija de tamaño $N = 1000$ . . . . .	46
5.1. $\sqrt{\widehat{RMSE}(\lambda_s)}$ para los 16 experimentos. . . . .	54
5.2. $\sqrt{\widehat{RMSE}(\ell_s)}$ para los 16 experimentos. . . . .	55
5.3. $\sqrt{RMSE}$ de los estimadores. . . . .	56
5.4. Errores Relativos de Estimación(ERE). . . . .	57
6.1. $\sqrt{RMSE(\hat{t}_{y,VC})}$ y $\%RB(\hat{t}_{y,VC})$ . . . . .	69
6.2. Modelos de Coeficientes Variantes: $\%RB(\hat{V}_{\lambda_{opt,i}})$ . . . . .	70
6.3. Modelos de Coeficientes Variantes. Probabilidad de Cobertura. . . . .	70

# Índice de figuras

5.1. Diagramas de dispersión de las estimaciones del total. Las estimaciones del total usando $\hat{\lambda}_s$ están sobre el eje horizontal y las estimaciones usando $\lambda_U$ están sobre el eje vertical. Las líneas horizontales y verticales indican el valor verdadero del total poblacional. . . . .	58
---	----

# Capítulo 1

## Introducción

Los recursos económicos para realizar censos de población y vivienda se tornan cada vez más difíciles de obtener. Sin embargo, las necesidades de información estadística continúan creciendo en varios sentidos. Se demanda:

1. Mayor frecuencia;
2. Mayor cobertura temática;
3. Mayor capacidad para el estudio de relaciones entre temas;
4. Mayor desagregación geográfica;
5. Mínima disminución en la precisión de la información producida.

Países con infraestructura adecuada han basado la producción de información sobre la totalidad de sus poblaciones y subpoblaciones, en los registros administrativos. Para otro conjunto de países, sus particulares condiciones políticas, económicas y sociales hacen que este enfoque no sea viable. Aunque algunos consideran que este es el camino a seguir en el futuro y ya trabajan para alcanzarlo, otros consideran impensable el disponer algún día de por lo menos un registro de población en el cuál basar tal enfoque.

Lo anterior ha conducido a algunas oficinas nacionales de estadística a explorar y, en algunos casos, a iniciar ya la instrumentación del uso combinado o alternativo de las fuentes para la generación de información estadística que recibirá el mismo uso que la censal. En particular, se tiene información de:

1. Censos y encuestas sociodemográficas (Estados Unidos);
2. Registros administrativos y encuestas (Países Bajos);

## 1. Introducción

---

3. Registros administrativos y censos (España);
4. Encuestas rotatorias (Francia).

En nuestro país el INEGI ha estado analizando e instrumentando algunas de las anteriores opciones. Fue así como se decidió que el censo mexicano del año 2000 utilizara un esquema combinado: un cuestionario básico, aplicado a la totalidad de la población (el censo propiamente dicho), y durante el mismo operativo censal una muestra de la población recibió una ampliación al anterior cuestionario (encuesta). Lo anterior puede ser ejemplo de lo señalado para lo referente a Estados Unidos, aunque éste incluye además a las encuestas intercensales. En principio, para el censo del 2010 también se hizo un cuestionario básico y al menos uno ampliado.

En este mismo sentido, parece que aquellos esquemas que recurren al uso de encuestas por muestreo suponen grandes tamaños de muestra para conseguir representatividad” (precisión adecuada) a niveles pequeños de desagregación geográfica, con lo que no se atiende a la reducción de costos, como en la encuesta levantada simultáneamente con el Censo 2000.

Adicionalmente, parece que el aspecto de reducción de costos puede ser atendido al relacionar los resultados de las encuestas con los de las otras dos fuentes: la censal y la registral (registros administrativos); tomando en cuenta las experiencias internacionales, así como los ejemplos de aplicación basados en información mexicana que pueden ser aún mejorados. En caso de contar con metodologías que permitiesen lo anterior, se podría concebir un esquema de generación de información en el que se integran el censo y tanto sus encuestas simultáneas como las levantadas durante el período intercensal para:

1. Lograr una reducción en los tamaños de muestra que implicaría ahorros sustanciales, si no para el ejercicio censal propiamente dicho, sí para los muestrales;
2. Contar con información geográficamente desagregada con mayor frecuencia;
3. Mantener la precisión alcanzada en los ejercicios muestrales.

Por lo dicho anteriormente, es conveniente contar con métodos de estimación que hagan un uso eficiente de la información generada por una muestra y la información auxiliar, proveniente de registros administrativos o información censal o ambos. Si existe alguna relación conocida entre la información de la muestra y la información auxiliar, por ejemplo una relación lineal, dicha relación podría utilizarse para obtener estimadores eficientes de los parámetros poblacionales de interés, el cual es el enfoque de estimación paramétrica. En el caso de no existir evidencia sobre alguna relación conocida, podemos utilizar métodos de estimación semiparamétricos, los cuales son independientes de la forma de la relación entre las variables de interés.

## 1. Introducción

---

Para el caso de estimación semiparamétrica en muestreo de poblaciones finitas, se han hecho algunos avances. En el trabajo de (Breidt y Opsomer, 2000) se ha desarrollado el estimador del total por regresión localmente polinomial. Además en (Breidt *et al.*, 2005) se propone un estimador para encuestas complejas utilizando splines penalizados. El concepto de encuestas complejas se refiere a la complejidad del diseño de muestreo utilizado. Este tipo de diseños son estratificados, en dos o más etapas y en la última etapa, las unidades de muestreo pueden ser conglomerados de unidades poblacionales. Este tipo de diseños de muestreo son utilizados, por ejemplo, en las encuestas periódicas del INEGI.

Para el caso de regresión localmente polinomial, (Breidt y Opsomer, 2000) proponen el estimador del total poblacional y su respectiva varianza, para una muestra observada denotada por  $s$  y un ancho de banda  $h_N$ :

$$\tilde{t}_y^0 = \sum_{i \in s} \frac{y_i - \hat{m}_i^0}{\pi_i} + \sum_{i \in U_N} \hat{m}_i^0,$$

donde

$$\begin{aligned} \pi_i &= P(i \in s), \\ e_1^T &= (1, 0, \dots, 0), \\ \hat{m}_i^0 &= e_1^T (X_{si}^T W_{si} X_{si})^{-1} X_{si}^T W_{si} y_s, \\ X_{si} &= [1, (x_j - x_i), \dots, (x_j - x_i)^p]_{j \in s}, \\ W_{si} &= \text{diag} \left\{ \frac{1}{\pi_j h_N} K \left( \frac{x_j - x_i}{h_N} \right) \right\}_{j \in s}. \end{aligned}$$

Un estimador de la varianza de  $\tilde{t}_y^0$  es:

$$\widehat{\text{Var}}(\tilde{t}_y^0) = \sum_{i, j \in s} \frac{\pi_{ij} - \pi_i \pi_j}{\pi_{ij}} \frac{y_i - \hat{m}_i^0}{\pi_i} \frac{y_j - \hat{m}_j^0}{\pi_j},$$

donde  $\pi_{ij} = P(\{i, j\} \subset s)$ .

Para el caso de regresión por splines penalizados, (Breidt *et al.*, 2005) proponen el estimador del total poblacional y su respectiva varianza, para una muestra observada  $s$  y un parámetro de suavizamiento  $\lambda$ :

$$\hat{t}_{y, spl} = \sum_{i \in U_N} \hat{m}_i,$$

## 1. Introducción

---

donde

$$\begin{aligned}\widehat{m}_i &= X_i^T [X_s^T \Pi_s X_s + A_\lambda]^{-1} X_s^T \Pi_s y_s, \\ X_i^T &= [1, x_i, \dots, x_i^p, (x_i - K_1)_+^p, \dots, (x_i - K_J)_+^p]_{i \in U_N}, \\ \Pi_s &= \text{diag} \left\{ \frac{1}{\pi_i} \right\}_{i \in s}, \\ A_\lambda &= \text{diag} \{0, \dots, 0, \lambda, \dots, \lambda\}.\end{aligned}$$

Un estimador de la varianza del estimador es:

$$\widehat{Var}(\widehat{t}_{y, spl}) = \sum_{i, j \in s} \frac{\pi_{ij} - \pi_i \pi_j}{\pi_{ij}} \frac{y_i - \widehat{m}_i}{\pi_i} \frac{y_j - \widehat{m}_j}{\pi_j}.$$

Los métodos de estimación semiparamétrica en muestreo antes citados están orientados principalmente para aplicarse al caso de una sola variable auxiliar. Cuando se tienen dos o más variables, es necesario buscar otras alternativas. Una de estas alternativas puede ser la estimación por regresión con coeficientes variantes. Dicha metodología ha sido propuesta en los trabajos de (Cai *et al.*, 2000b) y (Cao *et al.*, 2010) en el contexto del análisis de series de tiempo. Además parece ser factible la adaptación de esta metodología de estimación semiparamétrica al caso del análisis de encuestas complejas.

Por otro lado, en los trabajos de (Breidt y Opsomer, 2000) y (Breidt *et al.*, 2005), el estimador de la varianza del estimador semiparamétrico del total está basado en el concepto de estimador por diferencia (ver Sección 6.3 de (Sarndal *et al.*, 1992)). Este tipo de estimadores tiende a subestimar la varianza, por lo tanto es necesario explorar otras formas de proponer estimadores de dicha varianza.

Como se puede observar de lo anterior, la estimación semiparamétrica en el muestreo de poblaciones finitas es un tema de investigación de interés vigente. Además, los resultados de la investigación sobre esta temática han motivado un gran interés tanto teórico como práctico.

# Capítulo 2

## Objetivos

### 2.1. Objetivos Generales

Desarrollar y adaptar un método de estimación semiparamétrica en el análisis de encuestas complejas.

### 2.2. Objetivos Particulares

Desarrollar el modelo de coeficientes variantes para solucionar problemas de estimación de parámetros poblacionales utilizando información multivariada de encuestas complejas.

Proponer un estimador del total poblacional y estudiar algunas de sus propiedades estadísticas.

Proponer una forma de seleccionar los parámetros de suavizamiento en el contexto de muestreo de poblaciones finitas.

Realizar un estudio de simulación para estudiar de manera empírica la metodología propuesta.

Aplicar la metodología propuesta a conjunto de datos reales.



# Capítulo 3

## Muestreo de poblaciones finitas

En ocasiones en *Estadística* queremos estudiar una población finita de interés. Cuando la población es demasiado grande, es imposible estudiar las características individuales de los elementos de la población debido a tiempo y aspectos económicos. Entonces se recurre a estudiar solamente una *muestra* de esa *población*; donde la muestra seleccionada debe reflejar las características de interés de la población. A partir de la muestra, se realizan inferencias para conocer a toda la población.

### 3.1. Población, muestra y selección de muestras

Consideremos una población constituida de  $N$  elementos etiquetados como  $k = 1, 2, \dots, N$ . Denotaremos a la población finita como  $U = \{1, \dots, k, \dots, N\}$ , asumiendo que  $N$  no es necesariamente conocido. La característica de interés para cada elemento de la población se representará por  $y_U = \{y_1, y_2, \dots, y_N\}$ . Para el objetivo del presente capítulo,  $y_j \in \mathbb{R}$ ,  $j = 1, 2, \dots, N$ . En ocasiones es conveniente modelar la característica de interés con el conjunto de variables aleatorias  $Y_U = \{Y_1, Y_2, \dots, Y_N\}$ , es decir, el valor de la característica  $y_U$  se conceptúa como un posible valor de la variable aleatoria  $Y_U$ .

El espacio de probabilidad sobre el cual está definido el conjunto de variables aleatorias  $Y_U$  o la función de distribución conjunta del mismo, se denomina *modelo* de las características de interés.

Una *muestra* es cualquier subconjunto  $s$  de  $U$ . Por ejemplo una muestra  $s$  de tamaño  $n$  es de la forma  $s = \{k_1, \dots, k_n\}$ , donde  $k_j$  representa al  $k_j$  -ésimo elemento de la población  $U$ .

Para seleccionar los elementos de la población en la muestra es necesario un *procedimiento*

## 3.2. Diseño de muestreo

---

*de selección.* Un procedimiento de selección es un conjunto de experimentos cuyo objetivo es seleccionar elementos de la población para integrar la muestra.

## 3.2. Diseño de muestreo

Si  $S$  es un conjunto de muestras de  $U$ , entonces un *diseño de muestreo* es la probabilidad  $P$  de obtener cada una de las muestras en  $S$ , siempre que  $\sum_{s \in S} P(\{s\}) = 1$ . Usualmente,  $P$  es la función de probabilidad resultante del *procedimiento de selección* de la muestra.

Un conjunto de utilidad práctica de muestras  $S$  es el conjunto  $\{s \in U : P(\{s\}) > 0\}$ . Si  $S$  tiene esta última propiedad, a este conjunto se le denomina el *conjunto de posibles muestras* de  $U$  bajo el diseño de muestreo  $P$ . Con un conjunto de muestras  $S$  de este tipo y con el diseño de muestreo  $P$  es posible formar el espacio de probabilidad  $(S, 2^S, P)$ . Más adelante se mostrará que este espacio de probabilidad es el usado para hacer inferencia, siempre que dicha inferencia utilice únicamente la aleatoriedad descrita por tal espacio.

Varios procedimientos de selección de la muestra pueden producir un mismo diseño de muestreo  $P$ . Dado un diseño de muestreo  $P$ , el conjunto de experimentos del procedimiento de selección debe ser tal que la probabilidad resultante de seleccionar la muestra coincida con el diseño de muestreo  $P$ .

En ocasiones un diseño de muestreo se conoce por su procedimiento de selección y no por su probabilidad de seleccionar una muestra, por ejemplo, en el “*muestreo aleatorio simple sin reemplazo*” (MSR).

Ahora veamos dos procedimientos de selección para el MSR.

### Procedimiento 1 de selección.

1. Seleccionar con la misma probabilidad,  $1/N$ , un primer elemento de los  $N$  elementos de la población y no regresarlo;
2. Seleccionar con la misma probabilidad,  $1/(N - 1)$ , un segundo elemento de los  $N - 1$  elementos restantes y no regresarlo;
- ⋮
- $n$ . Seleccionar con la misma probabilidad,  $1/(N - n + 1)$ , un  $n - \text{ésimo}$  elemento de los  $N - n + 1$  elementos que quedan después de las primeras  $n - 1$  selecciones.

### 3.3. Probabilidades de inclusión

---

Este procedimiento de selección produce muestras  $s$  de tamaño  $n$ . Además, la probabilidad de obtener cada muestra  $s$  es

$$P(\{s\}) = \frac{1}{\binom{N}{n}}.$$

El conjunto de posibles muestras es  $S = \{s \subset U : s \text{ tiene } n \text{ elementos distintos}\}$ . Otro procedimiento de selección en un MSR es el siguiente.

#### Procedimiento 2 de selección.

1. Para cada elemento de la población generar un valor de una distribución *uniforme*  $(0, 1)$ ;
2. Ordenar en forma ascendente los elementos de acuerdo a los números generados;
3. Seleccionar los primeros  $n$  elementos para integrar la muestra.

Observe que los dos procedimientos de selección descritos anteriormente producen el mismo diseño de muestreo, ver ([Sarndal et al., 1992](#)).

### 3.3. Probabilidades de inclusión

Por la estructura del espacio de probabilidad asociada a un diseño, cualquier función del conjunto de muestras  $S$  a los números reales  $\mathbb{R}$  es una variable aleatoria. En especial las funciones

$$\lambda_k(s) = \begin{cases} \alpha & \text{si } k \in s, \alpha \neq 0 \\ 0 & \text{si } k \notin s \end{cases},$$

para todo  $k \in U$  y toda  $s \in S$ , son variables aleatorias.

La importancia de éste conjunto de variables aleatorias en la teoría de muestreo se muestra con el siguiente ejemplo.

**Ejemplo 1** *Variables Indicadoras.* Sea  $I_k$  la variable aleatoria que nos indica si la unidad  $k$  está en la muestra  $s$ . Esta variable se define como

$$I_k(s) = \begin{cases} 1 & \text{si } k \in s \\ 0 & \text{si } k \notin s \end{cases}$$

para todo  $k \in U$  y toda  $s \in S$ .

### 3.3. Probabilidades de inclusión

---

Ahora tenemos las propiedades de las variables indicadoras. Notemos que cada variable indicadora  $I_k$  es una variable aleatoria Bernoulli con probabilidad de éxito

$$P(I_k = 1) = P(\{s \in S : s \ni k\}) = \sum_{s \ni k} P(\{s\}) := \pi_k.$$

A  $\pi_k$  se le conoce como la *probabilidad de inclusión de primer orden* del elemento  $k$ . A  $1/\pi_k$  se le denomina *factor de expansión* y se utiliza en la estimación de parámetros poblacionales.

Dado que  $I_k$  es una variable aleatoria Bernoulli, entonces

$$E(I_k) = \pi_k,$$

y

$$Var(I_k) = \pi_k(1 - \pi_k),$$

$k = 1, \dots, N$ .

Ahora veamos como calcular la probabilidad de que la muestra contenga dos elementos, es decir,

$$\begin{aligned} P(\{j, k\} \subset s) &= P(I_j I_k = 1) \\ &= P(\{s \in S : s \supset \{j, k\}\}) \\ &= \sum_{s \supset \{j, k\}} P(\{s\}) := \pi_{jk}. \end{aligned}$$

A  $\pi_{jk}$  se le conoce como la *probabilidad de inclusión de segundo orden* de los elementos  $j$  y  $k$ . También es posible mostrar que

$$\begin{aligned} E(I_j I_k = 1) &= \pi_{jk}, \\ Cov(I_j, I_k) &= \pi_{jk} - \pi_j \pi_k := \Delta_{jk}. \end{aligned}$$

Veamos un ejemplo en donde apliquemos los conceptos de probabilidades de inclusión dados anteriormente, cuando utilizamos MSR para la selección de las muestras.

**Ejemplo 2** El conjunto de posibles muestras para el MSR es de la forma

$$S_{MSR} = \{s \subset U : s \text{ tiene } n \text{ elementos distintos}\},$$

y la probabilidad de seleccionar cada muestra, es decir, el diseño de muestreo, es

$$P_{MSR}(\{s\}) = \frac{1}{\binom{N}{n}},$$

para toda  $s \in S_{MSR}$ .

### 3.3. Probabilidades de inclusión

---

De esta forma la probabilidad de inclusión de primer orden para cada elemento  $k$  es

$$\pi_k = \sum_{s \ni k} P(\{s\}) = \sum_{s \ni k} \frac{1}{\binom{N}{n}} = \frac{\binom{N-1}{n-1}}{\binom{N}{n}} = \frac{n}{N},$$

ya que existen  $\binom{N-1}{n-1}$  muestras de tamaño  $n$  que contienen al elemento  $k$ .

De forma similar se obtiene que la probabilidad de inclusión de segundo orden de los elementos  $j$  y  $k$  es

$$\pi_{jk} = \sum_{s \supset \{j,k\}} P(\{s\}) = \sum_{s \supset \{j,k\}} \frac{1}{\binom{N}{n}} = \frac{\binom{N-2}{n-2}}{\binom{N}{n}} = \frac{n(n-1)}{N(N-1)}.$$

Cuando las muestras son de tamaño variable es útil tener la siguiente variable aleatoria, que nos proporciona el tamaño de la muestra. Esta variable se expresa como

$$\mathbf{n} = \sum_{k \in U} I_k.$$

Entonces para una muestra  $s$  seleccionada, su tamaño de muestra es

$$\mathbf{n}(s) = \sum_{k \in U} I_k(s) = \sum_{k \in s} 1,$$

$\mathbf{n}(s)$  es el número de elementos diferentes en  $s$ . También se tienen algunas propiedades de  $\mathbf{n}$ ,

$$E(\mathbf{n}) = \sum_{k \in U} E(I_k) = \sum_{k \in U} \pi_k;$$

es decir, la suma de las probabilidades de inclusión de primer orden proporciona el tamaño de muestra esperado. La expresión de la varianza del tamaño de la muestra es

$$Var(\mathbf{n}) = \sum_{k \in U} Var(I_k) + \sum_{k \neq j \in U} Cov(I_j, I_k) = \sum_{k \in U} \pi_k(1 - \pi_k) + \sum_{k \neq j \in U} (\pi_{jk} - \pi_j \pi_k).$$

Si la muestra  $s$  es de tamaño fijo  $n$ , entonces  $\mathbf{n} \equiv n$  con probabilidad 1 y

$$n = \sum_{k \in U} \pi_k.$$

## 3.4. Parámetros poblacionales y sus estimadores

En la primera sección denotamos a la característica de interés de cada elemento de la población por  $y_U = \{y_1, y_2, \dots, y_N\}$ . Cualquier función de  $y_U$  se le denomina *parámetro poblacional*. Si la característica de interés se modela con las variables aleatorias  $Y_U = \{Y_1, Y_2, \dots, Y_N\}$  cuya función de distribución conjunta es  $G_\theta$ , entonces a  $\theta$  se le denomina *parámetro del modelo*.

Las siguientes cantidades son ejemplos de parámetros poblacionales:

Total poblacional:

$$t_y = \sum_{k \in U} y_k,$$

Media poblacional:

$$\mu_y = \frac{1}{N} \sum_{k \in U} y_k,$$

Varianza poblacional:

$$\sigma_y^2 = \frac{1}{N-1} \sum_{k \in U} (y_k - \mu_y)^2.$$

Si para cada elemento de la población se estudian dos características, digamos

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\},$$

un parámetro poblacional puede ser

$$\theta = \frac{\sum_{k \in U} y_k}{\sum_{k \in U} x_k}.$$

Sea  $t \in \mathbb{R}$ . Otro parámetro poblacional es la función de distribución poblacional evaluada en  $t$ . Esta función se define por:

$$G_U(t) = \frac{\#(y_k \leq t)}{N}.$$

El anterior parámetro poblacional se puede interpretar como la fracción de individuos en la población cuyo valor de la característica de interés es menor o igual a  $t$ .

En general un estimador del parámetro poblacional  $\theta$  es de la forma

$$\hat{\theta}(y_1, y_2, \dots, y_N, \lambda_1, \lambda_2, \dots, \lambda_N).$$

### 3.5. Distribución aproximada de un estimador

---

En particular es función de la forma  $\hat{\theta}(y_1, y_2, \dots, y_N, I_1, I_2, \dots, I_N)$ . Si el estimador del parámetro poblacional  $\theta$  es de la forma

$$\hat{\theta}(y_1, y_2, \dots, y_N, \lambda_1, \lambda_2, \dots, \lambda_N),$$

la inferencia se dice basada en diseños, es decir, las propiedades estadísticas de  $\hat{\theta}$  se calculan con base al diseño de muestreo  $P$ . Si el estimador fuera de la forma

$$\hat{\theta}(Y_1, Y_2, \dots, Y_N, \lambda_1, \lambda_2, \dots, \lambda_N)$$

y sus propiedades estadísticas se calculan con base al modelo  $G_\theta$  únicamente, entonces la inferencia se dice basada en modelos. Además si la característica de interés se modela con las variables aleatorias  $Y_U = \{Y_1, Y_2, \dots, Y_N\}$ , entonces el parámetro poblacional  $\theta(Y_1, Y_2, \dots, Y_N)$ , antes visto como  $\theta(y_1, y_2, \dots, y_N)$ , es formalmente un estadístico y

$$\hat{\theta}(Y_1, Y_2, \dots, Y_N, \lambda_1, \lambda_2, \dots, \lambda_N)$$

es formalmente un predictor de este estadístico.

**Ejemplo 3** Si utilizamos *MSR*, un posible estimador del total poblacional  $t_y$  es

$$\hat{t}_y = \frac{N}{n} \sum_{k \in U} y_k I_k.$$

Para una muestra  $s$  seleccionada, la estimación de  $t_y$  es

$$\hat{t}_y(s) = \frac{N}{n} \sum_{k \in s} y_k = N\bar{y}_s.$$

Un parámetro poblacional toma un sólo valor y éste puede ser desconocido. Por lo tanto, un estimador tendría que proporcionar este único elemento, el valor del parámetro. Pedir lo anterior sería equivalente a pedir que cualquier muestra y cualquier método de estimación nos proporcionaran el valor del parámetro de interés. Lo cual es poco factible. Aquí un estimador nos proporcionará una variedad de valores para el parámetro de interés aunque este conjunto de valores no quede bien determinado. Lo deseable es que el rango de valores que contenga al parámetro poblacional fuera corto con una probabilidad grande.

### 3.5. Distribución aproximada de un estimador

Dada la dificultad para obtener la función de distribución de un estimador  $\hat{\theta}$  del parámetro poblacional  $\theta$ , se puede optar por obtener una aproximación a dicha distribución.

### 3.6. Teoría asintótica en muestreo de poblaciones finitas

---

Para cierto tipo de estimadores y diseños de muestreo es posible obtener que

$$\frac{\hat{\theta} - \theta}{\sqrt{\widehat{Var}(\hat{\theta})}}$$

tiene una distribución *Normal estándar*, para más detalle ver el Capítulo 3 de (Thompson, 1997). Esto es, cuando el tamaño de la población  $N \rightarrow \infty$  y el tamaño de la muestra  $n \rightarrow \infty$ , entonces la distribución del cociente anterior tiende a una normal estándar.

Con base a lo anterior, podemos construir intervalos de confianza para  $\theta$  de la forma

$$\hat{\theta} \pm z_{1-\frac{\alpha}{2}} \sqrt{\widehat{Var}(\hat{\theta})},$$

donde  $z_{1-\frac{\alpha}{2}}$  es el cuantil de orden  $1 - \frac{\alpha}{2}$  de una normal estándar. En otras palabras, dado el comportamiento de la distribución de  $\hat{\theta}$ , se espera que aproximadamente  $100(1 - \alpha)\%$  de los intervalos así construidos contengan al valor del parámetro poblacional  $\theta$  (tengan una cobertura aproximada de  $100(1 - \alpha)\%$ ).

### 3.6. Teoría asintótica en muestreo de poblaciones finitas

En esta sección se presentan definiciones y resultados que son de utilidad en el Capítulo 6.

En un contexto general de probabilidad y estadística, la notación  $o_p(1)$  (o pequeña  $p$ -uno) es para una sucesión de vectores aleatorios que convergen a cero en probabilidad. La expresión  $O_p(1)$  (o grande  $p$ -uno) denota una sucesión que es acotada en probabilidad. En general, para una sucesión dada de variables aleatorias  $R_n$ ,

$$X_n = o_p(R_n) \Leftrightarrow X_n = Y_n R_n, Y_n \xrightarrow{p} 0.$$

$$X_n = O_p(R_n) \Leftrightarrow X_n = Y_n R_n, Y_n = O_p(1).$$

Para formalizar los términos de orden estocástico mencionados anteriormente, sean  $\{a_n\}_{n=1}^{\infty}$  y  $\{b_n\}_{n=1}^{\infty}$  sucesiones de números reales, sean  $\{f_n\}_{n=1}^{\infty}$  y  $\{g_n\}_{n=1}^{\infty}$  sucesiones de números reales positivos, y sean  $\{X_n\}_{n=1}^{\infty}$ ,  $\{Y_n\}_{n=1}^{\infty}$  sucesiones de variables aleatorias.

**Definición** Decimos que  $a_n$  es de orden más pequeño que  $g_n$  y se escribe  $a_n = o(g_n)$  si

$$\lim_{n \rightarrow \infty} \frac{a_n}{g_n} = 0.$$



### 3.6. Teoría asintótica en muestreo de poblaciones finitas

---

**Definición** Decimos que  $a_n$  es a lo más de orden  $g_n$  y se escribe  $a_n = O(g_n)$  si existe un número real  $M$  tal que

$$g_n^{-1} |a_n| \leq M \quad \forall n.$$

**Definición** La sucesión de variables aleatorias  $\{X_n\}$  converge en probabilidad a la v.a.  $X$ , denotado como  $\text{plim } X_n = X$ , si para cualquier  $\epsilon > 0$

$$\lim_{n \rightarrow \infty} P \{|X_n - X| > \epsilon\} = 0.$$

**Definición** Decimos que  $X_n$  es de orden más pequeño en probabilidad que  $g_n$ , denotado como  $X_n = o_p(g_n)$  si

$$\text{plim } g_n^{-1} X_n = 0.$$

**Definición** Decimos que  $X_n$  es a lo más de orden  $g_n$  en probabilidad (o acotado en probabilidad por  $g_n$ ), denotado como  $X_n = O_p(g_n)$  si para cualquier  $\epsilon > 0$  existe un número real positivo  $M_\epsilon$  tal que

$$P[|X_n| \geq M_\epsilon g_n] \leq \epsilon, \quad \forall n.$$

Usando las definiciones anteriores y las propiedades de límites, se puede probar:

1. Si  $X_n = o_p(f_n)$  y  $Y_n = o_p(g_n)$ ,

$$X_n Y_n = o_p(f_n g_n),$$

$$X_n + Y_n = o_p(\text{máx}\{f_n, g_n\}),$$

$$|X_n|^s = o_p(f_n^s), \quad s > 0.$$

2. Si  $X_n = O_p(f_n)$  y  $Y_n = O_p(g_n)$ ,

$$X_n Y_n = O_p(f_n g_n),$$

$$X_n + Y_n = O_p(\text{máx}\{f_n, g_n\}),$$

$$|X_n|^s = O_p(f_n^s), \quad s > 0.$$

3. Si  $X_n = o_p(f_n)$  y  $Y_n = O_p(g_n)$ ,

$$X_n Y_n = o_p(f_n g_n).$$

Para más detalle de la teoría asintótica en estadística, ver ([Vaart, 1998](#)) y ([Serfling,](#)

### 3.6. Teoría asintótica en muestreo de poblaciones finitas

---

1980). En (Fuller, 2009) hay una excelente revisión de teoría asintótica en el contexto de muestreo.

En la siguiente parte de esta Sección se explica y describe la consistencia e insesgidez asintótica en el contexto de muestreo de poblaciones finitas.

Supongamos que  $\hat{\theta}_n$  es un estimador de  $\theta$  basado en una muestra de tamaño  $n$  de una población finita dada de tamaño  $N$ , entonces no podemos decir que  $n \rightarrow \infty$ , ya que  $n \leq N$ , además  $N$  es fijo y finito. En este contexto, los resultados asintóticos requieren una maquinaria más compleja con una sucesión de poblaciones crecientes tal que  $n$  y  $N$  tiendan a infinito.

Empezamos con la idea de una sucesión infinita de elementos, etiquetado como  $k = 1, 2, \dots$ , y una sucesión infinita asociada de valores de  $y$ , denotada por  $y_1, y_2, \dots$ , donde  $y_k$  está ligada al  $k$ -ésimo elemento.

Consideremos una sucesión de poblaciones  $U_1, U_2, U_3, \dots$ , donde  $U_\nu$  consiste de los primeros  $N_\nu$  elementos de la sucesión infinita de elementos mencionados anteriormente, esto es,  $U_\nu = \{1, 2, \dots, N_\nu\}$ . Se asume que  $U_1 \subset U_2 \subset U_3 \subset \dots$  y por lo tanto  $N_1 < N_2 < N_3 < \dots$ . Sea  $\theta_\nu$  los valores de cierto parámetro en la población  $U_\nu$ , es decir,  $\theta_\nu$  es una función de los valores  $y_1, y_2, \dots, y_{N_\nu}$ .

Para cada población  $U_\nu$ , considere un diseño de muestreo  $p_\nu(\cdot)$  que asigna una cierta probabilidad  $p_\nu(s_\nu)$  a cada muestra posible  $s_\nu$  de elementos de  $U_\nu$ . Sean  $\pi_{\nu k}$  y  $\pi_{\nu kl}$  ( $k, l = 1, 2, \dots, N_\nu$ ) las probabilidades de inclusión determinadas por el diseño  $p_\nu(\cdot)$ . Por simplicidad, se asume que el tamaño de muestra es fijo y se denota por  $n_\nu$ . De igual forma, se asume que  $n_1 < n_2 < n_3 < \dots$ . Se puede observar,  $\nu \rightarrow \infty$  significa que  $n_\nu \rightarrow \infty$  y  $N_\nu \rightarrow \infty$ . Sea  $\hat{\theta}_\nu$  un estimador de  $\theta_\nu$ , basado en los valores observados  $y_k$ , tales que  $k \in s_\nu$ .

Por ejemplo, el parámetro  $\theta_\nu$  puede ser la media poblacional

$$\theta_\nu = \bar{y}_{U_\nu} = \sum_{k \in U_\nu} \frac{y_k}{N_\nu}$$

y  $\hat{\theta}_\nu$  puede ser el estimador de Horvitz-Thompson (se revisa en la Sección siguiente)

$$\hat{\theta}_\nu = \sum_{k \in s_\nu} \frac{y_k}{N_\nu \pi_{\nu k}}. \quad (3.1)$$

Con referencia a la sucesión de poblaciones y diseños de muestreo descritos anteriormente, ahora se tienen las siguientes definiciones:

### 3.7. Estimador de Horvitz-Thompson

---

**Definición** Un estimador  $\hat{\theta}_\nu$  es asintóticamente insesgado para  $\theta_\nu$ , si

$$\lim_{\nu \rightarrow \infty} \left[ E_{p_\nu} \left( \hat{\theta}_\nu \right) - \theta_\nu \right] = 0.$$

**Definición** Un estimador  $\hat{\theta}_\nu$  es consistente para  $\theta_\nu$ , si para cualquier  $\epsilon > 0$  fijo,

$$\lim_{\nu \rightarrow \infty} P \left[ \left| \hat{\theta}_\nu - \theta_\nu \right| > \epsilon \right] = 0.$$

En el trabajo de (Isaki y Fuller, 1982) se dan condiciones para la consistencia del estimador (3.1) y el estimador de regresión.

### 3.7. Estimador de Horvitz-Thompson

**Definición** Supóngase que el parámetro poblacional de interés es el total poblacional  $t_y$ . El *estimador de Horvitz-Thompson* (**HT**) (Horvitz y Thompson, 1952)  $\hat{t}_{y\pi}$  del total poblacional  $t_y$  es de la forma

$$\hat{t}_{y\pi} = \sum_{k \in U} y_k \frac{I_k}{\pi_k},$$

siempre que  $\pi_k > 0$  para todo  $k \in U$ . Para una muestra  $s$  seleccionada, la estimación del total poblacional  $t_y$  es

$$\hat{t}_{y\pi}(s) = \sum_{k \in s} \frac{y_k}{\pi_k}. \quad (3.2)$$

Tenemos las siguientes propiedades del estimador de Horvitz-Thompson descritas en la siguiente:

#### Proposición 1

a. El estimador  $\hat{t}_{y\pi}$  es insesgado, es decir,  $E(\hat{t}_{y\pi}) = t_y$ .

b. Su varianza (un parámetro poblacional) es

$$Var(\hat{t}_{y\pi}) = \sum_{k \in U} \frac{y_k^2}{\pi_k} + \sum_{j \in U} \sum_{k \neq j \in U} \frac{\pi_{jk}}{\pi_j \pi_k} y_j y_k - t_y^2. \quad (3.3)$$

### 3.8. Muestreo para unidades poblacionales

---

c. Un estimador insesgado de su varianza es

$$\widehat{Var}(t_{y\pi}) = \sum_{k \in U} \left( \frac{1}{\pi_k} - 1 \right) y_k^2 \frac{I_k}{\pi_k} + \sum_{j \in U} \sum_{k \neq j \in U} (\pi_{jk} - \pi_j \pi_k) \frac{y_j y_k}{\pi_j \pi_k} \frac{I_j I_k}{\pi_{jk}},$$

suponiendo que  $\pi_{jk} > 0$  para todo  $k \neq j \in U$ . Para una muestra  $s$  seleccionada, la estimación de  $Var(t_{y\pi})$  es

$$\widehat{Var}(t_{y\pi})(s) = \sum_{k \in s} \left( \frac{1}{\pi_k} - 1 \right) \frac{y_k^2}{\pi_k} + \sum_{j \in s} \sum_{k \neq j \in s} \left( \frac{1}{\pi_j \pi_k} - \frac{1}{\pi_{jk}} \right) y_j y_k.$$

d. Si el tamaño de muestra es fijo, entonces la varianza en la Parte b se puede expresar como

$$Var(\widehat{t}_{y\pi}) = \sum_{i \in U} \sum_{i > j \in U} (\pi_i \pi_j - \pi_{ij}) \left( \frac{y_i}{\pi_i} - \frac{y_j}{\pi_j} \right)^2. \quad (3.4)$$

Esta cantidad se conoce como *la expresión de Sen-Yates-Grundy* de la varianza del estimador de **HT**, para más detalle de esta varianza, ver (Sen, 1953) y (Yates y Grundy, 1953). Además, un estimador insesgado de la anterior cantidad es

$$\widehat{Var}(t_{y\pi}) = \sum_{i \in U} \sum_{i > j \in U} (\pi_i \pi_j - \pi_{ij}) \left( \frac{y_i}{\pi_i} - \frac{y_j}{\pi_j} \right)^2 \frac{I_i I_j}{\pi_{ij}}. \quad (3.5)$$

Para una muestra  $s$  seleccionada, la estimación correspondiente es

$$\widehat{Var}(t_{y\pi})(s) = \sum_{i \in s} \sum_{i > j \in s} (\pi_i \pi_j - \pi_{ij}) \left( \frac{y_i}{\pi_i} - \frac{y_j}{\pi_j} \right)^2 \frac{1}{\pi_{ij}}. \quad (3.6)$$

**Demostración:** Ver (Sarndal *et al.*, 1992).

## 3.8. Muestreo para unidades poblacionales

A continuación se muestran algunos procedimientos de selección de muestras. Estos procedimientos son comúnmente utilizados en la práctica.

### 3.8.1. Muestreo Bernoulli

El *muestreo Bernoulli*,  $B$ , es un diseño de muestreo que produce muestras de tamaño variable. Una forma de seleccionar una muestra por este diseño es realizar  $N$  experimentos

### 3.8. Muestreo para unidades poblacionales

---

**Bernoulli** independientes y con probabilidad de éxito  $\pi$ . Si el  $i$  –ésimo experimento es un éxito, entonces el individuo  $i$  de la población es seleccionado en la muestra.

Este diseño tiene un valor didáctico y sirve para ilustrar un diseño de muestreo que produce muestras de tamaño variable. También, este diseño se puede utilizar para modelar el número de individuos que sí responden a uno o varios objetos de un cuestionario.

Las muestras son desde tamaño cero hasta el tamaño de la población. Si  $\pi$  es la probabilidad de éxito de los experimentos Bernoulli con los cuales se seleccionan a los individuos en la muestra, entonces el diseño de muestreo es

$$P(\{s\}) = \pi^{\mathbf{n}(s)}(1 - \pi)^{N - \mathbf{n}(s)},$$

donde  $\mathbf{n}$  representa a la variable aleatoria tamaño de la muestra. Es posible mostrar que las variables indicadoras  $I_1, \dots, I_N$  son variables aleatorias independientes Bernoulli( $\pi$ ). Entonces la variable aleatoria  $\mathbf{n}$  se distribuye como una distribución *Binomial*( $N, \pi$ ).

De acuerdo a lo comentado arriba tenemos que las probabilidades de inclusión de primer y segundo orden son

$$\pi_j = E(I_j) = \pi;$$

para todo  $j \in U$ , y

$$\pi_{ij} = E(I_i I_j) = \pi^2;$$

para todos  $i \neq j \in U$ .

Por consiguiente, el estimador de **HT** del total poblacional  $t_y$  es:

$$\hat{t}_{y\pi B} = \frac{1}{\pi} \sum_{j \in U} y_j I_j.$$

Para una muestra  $s$  seleccionada la estimación resultante es:

$$\hat{t}_{y\pi B}(s) = \frac{1}{\pi} \sum_{j \in s} y_j.$$

Ahora, la varianza del estimador de **HT** del total poblacional  $t_y$  es

$$Var(\hat{t}_{y\pi B}) = \left(\frac{1}{\pi} - 1\right) \sum_{j \in U} y_j^2,$$

ya que de la ecuación (3.3), tenemos que

$$Var(\hat{t}_{y\pi B}) = \sum_{k \in U} \frac{y_k^2}{\pi} + \sum_{j \in U} \sum_{k \neq j \in U} \frac{\pi^2}{\pi\pi} y_j y_k - \left(\sum_{k \in U} y_k\right)^2 = \left(\frac{1}{\pi} - 1\right) \sum_{k \in U} y_k^2.$$

### 3.8. Muestreo para unidades poblacionales

---

Un estimador de la varianza es

$$Var(\widehat{t_{y\pi B}}) = \left(\frac{1}{\pi} - 1\right) \sum_{j \in U} y_j^2 \frac{I_j}{\pi}.$$

Para una muestra  $s$  seleccionada la estimación resultante es:

$$Var(\widehat{t_{y\pi B}})(s) = \left(\frac{1}{\pi} - 1\right) \frac{1}{\pi} \sum_{j \in s} y_j^2.$$

#### 3.8.2. Muestreo aleatorio simple sin reemplazo

El *muestreo aleatorio simple sin reemplazo*, *MSR*, es uno de los diseños de muestreo más conocidos. Este diseño produce muestras de tamaño  $n$  fijo. También este diseño fué mencionado en la Sección 3.2 y se recomienda utilizarlo cuando la población es homogénea con respecto a la característica que se desea estudiar o no conocemos nada acerca de la población con respecto a la característica de interés. Además este diseño presupone que se tiene una lista de los individuos de la población y la cual se utilizará en la etapa de selección de la muestra. En ocasiones *MSR* se utiliza por conveniencia, cuando no se tiene información que permita la elección de un diseño alternativo.

La probabilidad de seleccionar una muestra por *MSR* es

$$P(\{s\}) = \frac{1}{\binom{N}{n}},$$

para toda muestra  $s$  de tamaño fijo  $n$ . Ya vimos en el ejemplo 2 de la Sección 3.3 que las probabilidades de inclusión de primer y segundo orden son

$$\pi_j = \frac{n}{N},$$

para todo  $j \in U$  y

$$\pi_{jk} = \frac{n(n-1)}{N(N-1)},$$

para cualesquiera  $\{j, k\} \subset U$ .

Entonces el estimador de **HT** del total poblacional  $t_y$  queda de la siguiente manera

$$\widehat{t_{y\pi MSR}} = \frac{N}{n} \sum_{j \in U} y_j I_j.$$

### 3.8. Muestreo para unidades poblacionales

---

Para una muestra  $s$  seleccionada, la estimación resultante es:

$$\widehat{t}_{y\pi MSR}(s) = \frac{N}{n} \sum_{j \in s} y_j.$$

La varianza del estimador de **HT** es de la manera siguiente:

$$Var(\widehat{t}_{y\pi MSR}) = N^2 \left( \frac{1}{n} - \frac{1}{N} \right) \sigma_y^2, \quad (3.7)$$

antes de desarrollar la deducción de (3.7), observese que en este caso tenemos muestras de tamaño fijo  $n$ , por tanto la ecuación (3.3) se reduce a la ecuación (3.4). Por consiguiente

$$\begin{aligned} Var(\widehat{t}_{y\pi MSR}) &= \sum_{i \in U} \sum_{i > j \in U} (\pi_i \pi_j - \pi_{ij}) \left( \frac{y_i}{\pi_i} - \frac{y_j}{\pi_j} \right)^2 \\ &= \frac{N-n}{n(N-1)} \sum_{i \in U} \sum_{i > j \in U} (y_i - y_j)^2 \\ &= \frac{N-n}{2n(N-1)} \left\{ \sum_{i \in U} \sum_{j \in U} (y_i - y_j)^2 - \sum_{i \in U} (y_i - y_i)^2 \right\} \\ &= \frac{N-n}{2n(N-1)} \left\{ 2 \sum_{i \in U} \sum_{j \in U} y_i^2 - 2 \sum_{i \in U} \sum_{j \in U} y_i y_j \right\} \\ &= \frac{N-n}{n(N-1)} \left\{ \sum_{i \in U} N y_i^2 - N^2 \mu_y^2 \right\} \\ &= \frac{N(N-n)}{n(N-1)} \sum_{i \in U} (y_i - \mu_y)^2 \\ &= \frac{N(N-n)}{n} \sigma^2 \\ &= N^2 \left( \frac{1}{n} - \frac{1}{N} \right) \sigma_y^2. \end{aligned}$$

Un estimador de la varianza anterior es:

$$Var(\widehat{\widehat{t}_{y\pi MSR}}) = \left( \frac{1}{n} - \frac{1}{N} \right) \frac{N}{N-1} \sum_{j \in U} \sum_{j < k \in U} (y_j - y_k)^2 \frac{I_j I_k}{\pi_{jk}},$$

observese que la ecuación anterior es fácil de obtener de la ecuación (3.5). Para una muestra  $s$  seleccionada, la estimación resultante es:

$$Var(\widehat{\widehat{t}_{y\pi MSR}})(s) = N^2 \left( \frac{1}{n} - \frac{1}{N} \right) \sigma_s^2, \quad (3.8)$$

### 3.8. Muestreo para unidades poblacionales

---

donde

$$\sigma_s^2 = \frac{1}{n-1} \sum_{j \in s} (y_j - \mu_s)^2$$

y

$$\mu_s = \frac{1}{n} \sum_{j \in s} y_j.$$

La expresión (3.8) se obtiene de forma similar a como se obtuvo la ecuación (3.7).

#### 3.8.3. Muestreo con probabilidad proporcional al tamaño

En ocasiones se conoce el valor de una variable auxiliar para cada uno de los individuos de la población, denotemos estos valores por  $\{x_1, \dots, x_N\}$ . También en ocasiones, estos valores son tales que  $x_k \propto y_k$  (aproximadamente) para todo  $k \in U$ . Si este es el caso, se acostumbra a denotar a  $x_k$  como el tamaño de la  $k$ -ésima unidad poblacional.

Con lo anterior, el término *muestreo con probabilidad proporcional al tamaño*, *PPT*, sirve para denotar a cualquier diseño de muestreo donde las probabilidades de inclusión de primer orden son proporcionales al tamaño de la unidad poblacional, donde  $\pi_k \propto x_k$ .

Observese que si se utiliza un diseño que produce muestras de tamaño fijo y si se utiliza el estimador de HT para estimar el total, entonces la varianza de dicho estimador se puede expresar como (ver la ecuación 3.4):

$$Var(\hat{t}_{y\pi}) = \sum_{i \in U} \sum_{i > j \in U} (\pi_i \pi_j - \pi_{ij}) \left( \frac{y_i}{\pi_i} - \frac{y_j}{\pi_j} \right)^2.$$

De aquí, si  $\pi_k$  es aproximadamente proporcional a  $x_k$ , y por tanto a  $y_k$ , entonces el estimador tendrá una varianza pequeña. Es posible obtener esta propiedad sobre la precisión del estimador cuando se tienen muestras de tamaño variable, ver el Capítulo 13 de (Huskova *et al.*, 1998). De lo anterior se desprende, que la combinación de un diseño con probabilidad proporcional al tamaño y el estimador de HT del total, producen estimaciones con una alta precisión.

Por lo tanto, es conveniente utilizar el muestreo PPT cuando se tiene la información disponible sobre lo que se ha denominado tamaño de la unidad poblacional.

#### *Diseño de muestreo*

En lugar de proporcionar la probabilidad de selección de la muestra, se describirá un procedimiento de selección de la muestra (PPT sin reemplazo).



### 3.8. Muestreo para unidades poblacionales

---

Procedimiento de selección:

1. Ordenar a los individuos de la población en forma descendente con respecto a su valor del tamaño de la unidad;
2. Para el elemento  $k = 1$ , generar un número aleatorio  $\varepsilon_1$  de una distribución  $Uniforme(0, 1)$ . Si

$$\varepsilon_1 \leq \frac{nx_1}{t_x},$$

entonces seleccionar al elemento  $k = 1$  en la muestra, en caso contrario no. Aquí  $t_x = \sum_{j \in U} x_j$ ;

3. Para  $k = 2, 3, \dots$ , generar un número aleatorio  $\varepsilon_k$  de una  $Uniforme(0, 1)$ . Si

$$\varepsilon_k \leq \frac{(n - n_k)x_k}{t_{xk}},$$

entonces seleccionar al elemento  $k$  en la muestra, en caso contrario no. Aquí  $n_k$  es el número de elementos seleccionados en la muestra hasta el momento  $k - 1$  y  $t_{xk} = x_k + x_{k+1} + \dots + x_N$ .

4. El proceso descrito en los pasos 1 al 3 termina cuando  $n_k = n$  o  $k = k^*$ , lo que ocurra primero, donde  $k^* = \min\{k_0, N - n + 1\}$  y  $k_0$  es el mínimo  $k$  para el cual  $\frac{nx_k}{t_{xk}} > 1$ ;
5. Si  $n_{k^*} < n$ , el proceso de selección descrito en los pasos 1 al 3 no produjo una muestra del tamaño deseado. Los  $n - n_{k^*}$  elementos faltantes en la muestra son seleccionados de los restantes  $N - k^* + 1$  de la población por muestreo MSR.

Los detalles del anterior procedimiento de selección se puede consultar en ([Sunter, 1977](#)).

#### ***Probabilidades de Inclusión***

Es posible mostrar que las probabilidades de inclusión de primer orden son:

$$\pi_k = \begin{cases} \frac{nx_k}{t_x}, & k = 1, 2, \dots, k^* - 1 \\ \frac{n\bar{x}_{k^*}}{t_x}, & k = k^*, \dots, N \end{cases}$$

donde

$$\bar{x}_{k^*} = \frac{t_{xk^*}}{N - k^* + 1}.$$

El diseño de muestreo descrito aquí produce probabilidades de inclusión de primer orden proporcionales al tamaño de la unidad, excepto para las últimas unidades de tamaño más pequeño.

### 3.8. Muestreo para unidades poblacionales

Se puede mostrar también que las probabilidades de inclusión de segundo orden son:

$$\pi_{jk} = \begin{cases} \frac{n(n-1)}{t_x} g_j x_j x_k & \text{para } 1 \leq j < k < k^* \\ \frac{n(n-1)}{t_x} g_j x_j \bar{x}_{k^*} & \text{para } 1 \leq j < k^* \leq k \leq N \\ \frac{n(n-1)}{t_x} g_{k^*-1} \frac{t_{k^*} - x_{k^*-1}}{t_{k^*} - \bar{x}_{k^*}} (\bar{x}_{k^*})^2 & \text{para } k^* \leq j < k \leq N, \end{cases}$$

donde  $g_1 = \frac{1}{t_{x2}}$  y para  $k = 2, 3, \dots, k^* - 1$

$$\begin{aligned} g_k &= \left(1 - \frac{x_1}{t_{x,2}}\right) \left(1 - \frac{x_2}{t_{x,3}}\right) \dots \left(1 - \frac{x_{k-1}}{t_{x,k}}\right) / t_{x,k+1} \\ &= g_{k-1} \frac{t_k - x_{k-1}}{t_{k+1}}. \end{aligned}$$

Veamos un ejemplo ilustrando el muestreo PPT.

**Ejemplo 4** Supongamos que se desea seleccionar una muestra de tamaño  $n = 2$  de una población de tamaño  $N = 5$  por muestreo PPT sin reemplazo y con:

$k$	$x_k$	$t_{xk}$	$\frac{nx_k}{t_{xk}}$
1	40	100	$\frac{80}{100}$
2	25	60	$\frac{50}{60}$
3	20	35	$> 1$
4	10	15	$> 1$
5	5	5	$> 1$

Aquí  $k_0 = 3$ ,  $k^* = \min\{k_0, N - n + 1\} = 3$ ,

$$\pi_1 = \frac{80}{100}, \pi_2 = \frac{50}{100}, \pi_3 = \pi_4 = \pi_5 = \frac{7}{30}.$$

Además

$$g_1 = \frac{1}{60}; g_2 = g_1 \frac{t_{x2} - x_1}{t_{x3}} = \frac{1}{60} \frac{60 - 40}{35} = \frac{1}{105}; \bar{x}_3 = \frac{35}{3};$$

$$\pi_{1,2} = \frac{2}{100} \frac{1}{60} (40)(25) = \frac{1}{3};$$

$$\pi_{1,3} = \pi_{1,4} = \pi_{1,5} = \frac{2}{100} \frac{1}{60} (40) \frac{35}{3} = \frac{7}{45};$$

$$\pi_{2,3} = \pi_{2,4} = \pi_{2,5} = \frac{2}{100} \frac{1}{105} (25) \frac{35}{3} = \frac{1}{18};$$

$$\pi_{3,4} = \pi_{3,5} = \pi_{4,5} = \frac{2}{100} \frac{1}{105} \frac{35 - 25}{35 - 35/3} \left(\frac{35}{3}\right)^2 = \frac{1}{90}.$$

## 3.8. Muestreo para unidades poblacionales

---

### 3.8.4. Muestreo estratificado

En *muestreo estratificado*,  $ST$ , la población es dividida en subpoblaciones ajenas llamadas *estratos*. Un muestreo de probabilidad es seleccionado en cada estrato. Las selecciones en los estratos diferentes son independientes. El muestreo estratificado es un método poderoso y flexible que es ampliamente usado en la práctica.

Algunas de las razones por las cuales es conveniente utilizar muestreo estratificado son:

*i*). Supongamos que se desea obtener estimaciones por separado de algunas de las subpoblaciones. Cada subpoblación de estudio puede tratarse como un estrato si la pertenencia a cada una de éstas está especificada en el marco de muestreo;

*ii*). Por razones administrativas, diferentes regiones geográficas pueden tratarse como estratos.

*iii*). Disminuye la varianza de los estimadores.

Por variable de estratificación entendemos como la característica o las características usadas para subdividir la población en estratos. Por ejemplo, en estratificación de edad y sexo sería preferible a una estratificación por grupo ocupacional. Un diseño de muestreo y un tamaño de muestra puede ser especificado en cada estrato. Frecuentemente el mismo tipo de diseño de muestreo es aplicado en todos los estratos. Un estimador puede ser especificado para cada estrato. A menudo esta elección es también hecha uniformemente para todos los estratos.

Por una estratificación de una población finita  $U = \{1, \dots, k, \dots, N\}$  entendemos como una partición de  $U$  en  $H$  subpoblaciones. Un *estrato* es un subconjunto de la población tal que la pertenencia de un elemento al estrato es conocida. Los estratos de una población  $U$  se representarán por  $U_1, \dots, U_H$ . Estos subconjuntos son tales que  $U_j \cap U_k = \emptyset$  para toda  $j \neq k$  y  $U_1 \cup U_2 \cup \dots \cup U_H = U$ . Por *muestreo estratificado* entendemos que una muestra  $s_h$  es seleccionada de  $U_h$  acorde a un diseño  $P_h(\cdot)$  ( $h = 1, \dots, H$ ) y que la selección en un estrato es independiente de las selecciones en todos los otros estratos. En esta familia de diseños se seleccionan independientemente una muestra en cada estrato y posiblemente con diseños diferentes.

La muestra total que resulta, denotada como  $s$ , estará compuesta como

$$s = \bigcup_{h=1}^H s_h,$$

y por la característica de independencia,

$$P(\{s\}) = P_1(\{s_1\})P_2(\{s_2\}) \cdots P_H(\{s_H\}).$$

### 3.8. Muestreo para unidades poblacionales

Las probabilidades de inclusión de primer y segundo orden están definidos de la manera siguiente:

$$\pi_k = P(\{k \in s\}) = P_h(\{k \in s_h\}) = \sum_{s_h \ni k} P_h(\{s_h\}),$$

donde  $s_h \ni k$  denota que la suma es sobre aquellas muestras  $s_h$  que contienen al elemento  $k$ , para  $k \in U_h$ . Las probabilidades de inclusión de segundo orden están dados por

$$\pi_{kl} = P(\{\{k, l\} \subset s\}) = P_h(\{\{k, l\} \subset s_h\}) = \sum_{s_h \supset \{k, l\}} P_h(\{s_h\}),$$

si las unidades  $k$  y  $l$  están en el mismo estrato  $h$ . Ahora si la unidad  $k$  está en el estrato  $h$  y la unidad  $l$  está en el estrato  $h'$  (en diferentes estratos), entonces

$$\pi_{kl} = P(\{\{k, l\} \subset s\}) = P_h(\{k \in s_h\} \cap \{l \in s_{h'}\}) = P_h(\{k \in s_h\}) P_{h'}(\{l \in s_{h'}\}) := \pi_k \pi_l.$$

El número de elementos en el estrato  $h$ , es llamado el tamaño del estrato  $h$ , denotado por  $N_h$ , el cual suponemos conocido. Ya que los estratos forman una partición de  $U$ , tenemos que  $N = \sum_{h=1}^H N_h$ . Además, la población total puede estar descompuesta como

$$t_y = \sum_{k \in U} y_k = \sum_{h=1}^H t_h = \sum_{h=1}^H N_h \mu_{U_h},$$

donde  $t_h = \sum_{k \in U_h} y_k$  es el total del estrato  $h$  y  $\mu_{U_h}$  es la media del estrato  $h$ . Sea  $W_h = \frac{N_h}{N}$  el tamaño relativo del estrato  $U_h$ , entonces la media poblacional tiene la descomposición  $\mu_U = \sum_{h=1}^H W_h \mu_{U_h}$ .

En muestreo estratificado, el estimador de la población total  $t_y = \sum_{k \in U} y_k$  puede ser escrita como

$$\hat{t}_{y\pi ST} = \sum_{h=1}^H \hat{t}_{hy\pi},$$

donde  $\hat{t}_{hy\pi}$  es el estimador de  $t_h = \sum_{k \in U_h} y_k$ . La varianza puede ser escrita como

$$Var_{ST}(\hat{t}_{y\pi ST}) = \sum_{h=1}^H Var_{ST}(\hat{t}_{hy\pi}).$$

Un estimador insesgado de la varianza esta dado por

$$\widehat{Var}_{ST}(\hat{t}_{y\pi ST}) = \sum_{h=1}^H \widehat{Var}_{ST}(\hat{t}_{hy\pi}),$$

suponiendo que existe un estimador insesgado de la varianza  $\widehat{Var}_{ST}(\hat{t}_{hy\pi})$  para cualquier  $h$ .

### 3.8. Muestreo para unidades poblacionales

**Muestreo aleatorio simple sin reemplazo estratificado** En este diseño se selecciona una muestra por *MSR* en cada estrato. Este diseño lo denotaremos por *STSR*. Las siguientes propiedades de los estimadores para *STSR* se pueden demostrar usando las técnicas utilizadas en la Sección 3.8.2. Entonces el estimador de HT del total poblacional  $t_y$  es:

$$\hat{t}_{y\pi STSR} = \sum_{h=1}^H \left( \frac{N_h}{n_h} \sum_{j \in U_h} y_j I_j \right).$$

Para una muestra  $s = s_1 \cup s_2 \cup \dots \cup s_H$  seleccionada, la estimación resultante es:

$$\hat{t}_{y\pi STSR}(s) = \sum_{h=1}^H \left( \frac{N_h}{n_h} \sum_{j \in s_h} y_j \right).$$

Ahora, la varianza del estimador de HT del total poblacional  $t_y$  es:

$$Var(\hat{t}_{y\pi STSR}) = \sum_{h=1}^H \left[ N_h^2 \left( \frac{1}{n_h} - \frac{1}{N_h} \right) \sigma_h^2 \right],$$

donde  $\sigma_h^2 = \frac{\sum_{k \in U_h} (y_k - \mu_h)^2}{N_h - 1}$  y  $\mu_h$  es la media de la característica de interés en el estrato  $h$ . Un estimador de la varianza anterior es:

$$Var(\widehat{t}_{y\pi STSR}) = \sum_{h=1}^H \left[ \left( \frac{1}{n_h} - \frac{1}{N_h} \right) \frac{N_h}{N_h - 1} \sum_{j \in U_h} \sum_{k \in U_h, j < k} (y_j - y_k)^2 \frac{I_j I_k}{\pi_{jk}} \right].$$

Para una muestra  $s = s_1 \cup s_2 \cup \dots \cup s_H$  seleccionada, la estimación resultante es:

$$Var(\widehat{t}_{y\pi STSR})(s) = \sum_{h=1}^H \left[ N_h^2 \left( \frac{1}{n_h} - \frac{1}{N_h} \right) \sigma_{s_h}^2 \right],$$

donde

$$\sigma_{s_h}^2 = \frac{1}{n_h - 1} \sum_{j \in s_h} (y_j - \mu_{s_h})^2$$

y

$$\mu_{s_h} = \frac{1}{n_h} \sum_{j \in s_h} y_j.$$

#### 3.8.5. Muestreo de conglomerados en una sola etapa

En ocasiones no se tiene, en el marco de muestreo, la lista de los individuos de la población. Sin embargo, sí se tiene una lista de subconjuntos de la población. Como no se tiene la lista de los individuos de la población, no es posible saber que elementos pertenecen a

### 3.8. Muestreo para unidades poblacionales

---

que subconjunto y muy posiblemente tampoco se conozca cuantos elementos contiene cada subconjunto. A estos subconjuntos así descritos se les denomina *conglomerados*.

En estudios socioeconómicos de una población, en general no se dispone de la lista de individuos de estudio pero sí se tiene por ejemplo una lista de viviendas de estos individuos. En estudios electorales no se tiene la lista de los votantes pero si una lista de las secciones electorales. Estos son ejemplos para ilustrar posibles conglomerados (viviendas, secciones electorales, etc.).

El *muestreo de conglomerados*,  $C$ , es una familia de diseños; en los cuales los conglomerados son las unidades de muestreo. En diseños de muestreo como el B, el MSR, las unidades de muestreo son los individuos de la población.

Una razón para utilizar muestreo de conglomerados es por conveniencia si existe necesidad para utilizar este diseño. Si se tiene una lista de los individuos de la población, es mejor utilizar ésta en lugar de conglomerados. Se utiliza un diseño de muestreo que selecciona conglomerados porque no se tiene mayor información en el marco de muestreo.

En muestreo de conglomerados la población finita  $U = \{1, \dots, i, \dots, N\}$  es particionada en  $M$  subpoblaciones, llamados conglomerados, y denotados por  $U_1, \dots, U_i, \dots, U_M$ . El conjunto de conglomerados, simbólicamente es representado como  $U_I = \{1, \dots, i, \dots, M\}$ , esto representa una población de conglomerados del cual se selecciona una muestra de conglomerados.

El número de elementos de la población en el  $i$  -ésimo conglomerado  $U_i$  es denotado por  $N_i$ . La partición de  $U$  es expresada por las ecuaciones  $U = \bigcup_{i \in U_I} U_i$  y  $N = \sum_{i \in U_I} N_i$ .

El muestreo de conglomerado en una sola etapa (o simplemente *muestreo de conglomerado*) es ahora definido de la manera siguiente:

a. Una muestra  $s^c$  de  $n^c$  conglomerados es tomado de  $U_i$  acorde al diseño  $P_I(\cdot)$ . El tamaño de  $s^c$  es denotado por  $n$ , para un diseño de tamaño fijo, o por  $n_{s^c}$  para un diseño de tamaño variable.

b. Cualquier elemento de la población en los conglomerados seleccionados es observado.

Aquí,  $P_I(\cdot)$  puede ser cualquiera de los diseños convencionales. Continuamos usando  $s$  como un símbolo para el conjunto de elementos que son observados. Entonces,

$$s = \bigcup_{j \in s^c} U_j.$$

El tamaño de  $s$  es  $n_s = \sum_{i \in s^c} N_i$ . Notemos que si  $P_I(\cdot)$  es un diseño de tamaño fijo, el número de elementos observados  $n_s$  en general no será fijo, porque los tamaños de conglomerados  $N_i$  pueden variar.

### 3.8. Muestreo para unidades poblacionales

Las probabilidades de inclusión de primer y segundo orden inducidas por el diseño  $P_I(\cdot)$  son  $\pi_k = \pi_j^c$  para todo  $k \in U_j$  y donde  $\pi_j^c$  es la probabilidad de inclusión de primer orden para el  $j$ -ésimo conglomerado. Ahora, las probabilidades de inclusión de segundo orden son

$$\pi_{kl} = \begin{cases} \pi_j^c & \text{si } k, l \in U_j \\ \pi_{ij}^c & \text{si } k \in U_i \text{ y } l \in U_j, \end{cases}$$

donde  $\pi_{ij}^c$  es la probabilidad de inclusión de segundo orden para los conglomerados  $i$  e  $j$ .

Es conveniente introducir la notación simplificada

$$t_j = \sum_{k \in U_j} y_k$$

para el total del  $j$ -ésimo conglomerado. La población total  $t_y$  puede ser expresada como

$$t_y = \sum_{k \in U} y_k = \sum_{j \in U_I} t_j.$$

En el muestreo de conglomerados, el estimador de HT de la población total  $t_y$  puede ser escrito como

$$\widehat{t}_{y\pi C} = \sum_{j \in U_I} t_j \frac{I_j^c}{\pi_j^c},$$

donde  $I_j^c$  es la indicadora de que el  $j$ -ésimo conglomerado está en la muestra. Para una muestra  $s = \bigcup_{j \in s^c} U_j$  seleccionada, la estimación correspondiente es

$$\widehat{t}_{y\pi C}(s) = \sum_{j \in s^c} \frac{t_j}{\pi_j^c}.$$

La varianza es dada por

$$Var(\widehat{t}_{y\pi C}) = \sum_{j \in U_I} \frac{t_j^2}{\pi_j^c} + \sum_{i, j \in U_I, i \neq j} \frac{\pi_{ij}^c}{\pi_i^c \pi_j^c} t_i t_j - t_y^2. \quad (3.9)$$

Un estimador insesgado de la varianza anterior es

$$Var(\widehat{\widehat{t}_{y\pi C}}) = \sum_{j \in U_I} \left( \frac{1}{\pi_j^c} - 1 \right) t_j^2 \frac{I_j^c}{\pi_j^c} + \sum_{i, j \in U_I, i \neq j} (\pi_{ij}^c - \pi_i^c \pi_j^c) \frac{t_i t_j}{\pi_i^c \pi_j^c} \frac{I_i^c I_j^c}{\pi_{ij}^c}.$$

### 3.8. Muestreo para unidades poblacionales

Para una muestra  $s = \bigcup_{j \in s^c} U_j$  seleccionada, la estimación correspondiente es

$$\widehat{Var}(t_{y\pi C})(s) = \sum_{j \in s^c} \left( \frac{1}{\pi_j^c} - 1 \right) \frac{t_j^2}{\pi_j^c} + \sum_{i, j \in s^c, i \neq j} \left( \frac{1}{\pi_i^c \pi_j^c} - \frac{1}{\pi_{ij}^c} \right) t_i t_j.$$

Si el tamaño de muestra (de conglomerados) es fijo, entonces la varianza  $Var(\widehat{t}_{y\pi C})$  de la expresión (3.9) se reduce a

$$Var(\widehat{t}_{y\pi C}) = -\frac{1}{2} \sum_{i \in U_I} \sum_{j \in U_I} (\pi_{ij}^c - \pi_i^c \pi_j^c) \left( \frac{t_i}{\pi_i^c} - \frac{t_j}{\pi_j^c} \right)^2.$$

Un estimador insesgado de la varianza anterior para una muestra  $s = \bigcup_{j \in s^c} U_j$  seleccionada es

$$\widehat{Var}(\widehat{t}_{y\pi C})(s) = -\frac{1}{2} \sum_{i \in s^c} \sum_{j \in s^c} \frac{(\pi_{ij}^c - \pi_i^c \pi_j^c)}{\pi_{ij}^c} \left( \frac{t_i}{\pi_i^c} - \frac{t_j}{\pi_j^c} \right)^2.$$

El resultado anterior nos lleva a algunas conclusiones interesantes acerca de la eficiencia de muestreo de conglomerados. De la última ecuación, vemos que si todos los  $\frac{t_i}{\pi_i^c}$  son iguales, entonces  $Var(\widehat{t}_{y\pi C}) = 0$ . Así, si podemos elegir  $\pi_i^c \propto t_i$ , entonces el muestreo de conglomerados será altamente eficiente. Si los tamaños de conglomerados  $N_i$  son conocidos en la etapa de planeación, uno puede elegir un diseño con  $\pi_i^c \propto N_i$ . Ya que  $t_i = N_i \mu_{U_i} = \sum_{k \in U_i} y_k$ , esto es una buena elección si hay poca variación entre la media de los conglomerados  $\mu_{U_i}$ . Si todos los  $\mu_{U_i}$  son iguales, en efecto tendríamos  $Var(\widehat{t}_{y\pi C}) = 0$ .

**Muestreo de conglomerados bajo MSR** Esto es, una muestra  $s_I$  de tamaño fijo  $n_I$  es tomada por MSR de los  $N_I$  conglomerados en  $U_I$ , y todos los elementos en los conglomerados seleccionados son observados. Por el resultado presentado anteriormente, el estimador de la población total es dado por

$$\widehat{t}_{y\pi CMSR} = N_I \mu_{s_I},$$

donde,  $\mu_{s_I} = \frac{\sum_{i \in s_I} t_i}{n_I}$  es la media de los conglomerados totales  $t_i$  en  $s_I$ . La varianza puede ser escrita como

$$Var(\widehat{t}_{y\pi CMSR}) = N_I^2 \frac{1 - f_I}{n_I} \sigma_{tU_I}^2,$$

donde,  $f_I = \frac{n_I}{N_I}$  es la fracción de muestreo del conglomerado y

$$\sigma_{tU_I}^2 = \frac{1}{N_I - 1} \sum_{i \in U_I} (t_i - \mu_{U_I})^2,$$



### 3.8. Muestreo para unidades poblacionales

con  $\mu_{U_I} = \frac{\sum_{i \in U_I} t_i}{N_I}$ . El estimador de la varianza insesgado es

$$Var(\widehat{t_{y\pi CMSR}}) = N_I^2 \frac{1 - f_I}{n_I} \sigma_{ts_I}^2,$$

donde,

$$\sigma_{ts_I}^2 = \frac{1}{n_I - 1} \sum_{i \in s_I} (t_i - \mu_{s_I})^2.$$

**Muestreo estratificado con MSR de conglomerados en cada estrato** En este caso, la población esta dividida en  $H$  estratos;  $U = U_1 \cup U_2 \cup \dots \cup U_H$ . Ahora el  $h$  –ésimo estrato está compuesto de  $M_h$  conglomerados;  $U_h = U_{h1} \cup U_{h2} \cup \dots \cup U_{hM_h}$ . El total poblacional se puede expresar como

$$t_y = \sum_{h=1}^H t_h,$$

donde,  $t_h = \sum_{j=1}^{M_h} t_{hj}$  y  $t_{hj} = \sum_{k \in U_{hj}} y_k$ .

Ahora, el estimador de HT del total poblacional  $t_y$  es:

$$\widehat{t_{y\pi STMSRC}} = \sum_{h=1}^H \left( \frac{M_h}{n_h^c} \sum_{j=1}^{M_h} t_{hj} I_{hj}^c \right).$$

Para una muestra  $s = s_1 \cup s_2 \cup \dots \cup s_H$  seleccionada; donde  $s_h = \bigcup_{j \in s_h^c} U_{hj}$  y  $s_h^c$  representa la muestra de conglomerados seleccionados con el diseño  $P_h$ , la estimación resultante es:

$$\widehat{t_{y\pi STMSRC}}(s) = \sum_{h=1}^H \left( \frac{M_h}{n_h^c} \sum_{j \in s_h^c} t_{hj} \right).$$

La varianza del estimador de HT del total poblacional  $t_y$  es:

$$Var(\widehat{t_{y\pi STMSRC}}) = \sum_{h=1}^H \left[ M_h^2 \left( \frac{1}{n_h^c} - \frac{1}{M_h} \right) \sigma_h^2 \right],$$

donde

$$\sigma_h^2 = \frac{1}{M_h - 1} \sum_{j \in U_h} (t_{hj} - \mu_h)^2$$

y

$$\mu_h = \frac{1}{M_h} \sum_{j \in U_h} t_{hj}.$$

### 3.9. Muestreo de poblaciones finitas asistido por modelos

Un estimador de la varianza anterior es:

$$Var(\widehat{t_{y\pi STMSRC}}) = \sum_{h=1}^H \left[ M_h^2 \left( \frac{1}{n_h^c} - \frac{1}{M_h} \right) \frac{1}{n_h^c(n_h^c - 1)} \sum_{j,k \in U_h, j < k} (t_{hj} - t_{hk})^2 I_j^c I_k^c \right].$$

Para una muestra  $s = s_1 \cup s_2 \cup \dots \cup s_H$  seleccionada, la estimación resultante es:

$$Var(\widehat{t_{y\pi STMSRC}})(s) = \sum_{h=1}^H \left[ M_h^2 \left( \frac{1}{n_h^c} - \frac{1}{M_h} \right) \sigma_{s_h}^2 \right],$$

donde

$$\sigma_{s_h}^2 = \frac{1}{n_h^c - 1} \sum_{j \in s_h^c} (t_{hj} - \mu_{s_h})^2$$

y

$$\mu_{s_h} = \frac{1}{n_h^c} \sum_{j \in s_h^c} t_{hj}.$$

### 3.9. Muestreo de poblaciones finitas asistido por modelos

En esta sección se presenta y desarrolla la teoría básica del muestreo de poblaciones finitas asistido por modelos. En la inferencia de poblaciones finitas basada en el diseño de muestreo, las propiedades estadísticas en la estimación de los parámetros poblacionales debe estar basada en el diseño del muestreo usado.

Cuando se tiene conocimiento de información auxiliar de tipo continuo o categórico a nivel poblacional, decimos que para cada elemento de la población existe un vector de información auxiliar  $\mathbf{x}_k$  para la  $k$ -ésima unidad. Si este vector contiene  $r$  variables auxiliares entonces toma la siguiente forma:

$$\mathbf{x}_k = (x_{1k}, \dots, x_{rk})^T.$$

Cuando se pretende determinar la relación entre la variable de interés y la información auxiliar, es necesario acudir a un modelo probabilístico de superpoblación,  $\xi$ .

Supongamos que los valores  $y_1, \dots, y_N$  son una realización de un modelo de superpoblación  $\xi$  y sean  $\mathbf{x}_1, \dots, \mathbf{x}_N$  vectores de variables auxiliares. La relación entre  $y_k$  y  $\mathbf{x}_k$  está dada por un modelo de superpoblación  $\xi$  de la forma:

$$y_k = \mathbf{x}_k^T \beta + \epsilon_k, \quad (3.10)$$

donde  $\epsilon_k \sim (0, c_k \sigma^2)$  son variables aleatorias independientes e idénticamente distribuidas,

### 3.9. Muestreo de poblaciones finitas asistido por modelos

con  $k \in U$ . Al vector  $\beta$  se le conoce como vector de coeficientes de regresión en el modelo de superpoblación.

Es fácil observar que  $E_\xi(y_k) = \mathbf{x}_k^T \beta$  y  $Var_\xi(y_k) = c_k \sigma^2$ .

Bajo este modelo de superpoblación los valores  $y_1, \dots, y_N$  para la variable de interés se consideran realizaciones de variables aleatorias.

(Cassel *et al.*, 1976) afirma que a  $\xi$  se le conoce como modelo de superpoblación porque supone que la población finita  $U$  se toma como si hubiese sido seleccionada de un universo más grande al que pertenecen todo tipo de valores para  $y_k$  y para  $\mathbf{x}_k$ .

#### 3.9.1. Estimación en la población finita

Una forma de estimar el coeficiente de regresión  $\beta$  es utilizar el método de mínimos cuadrados, el cual tendrá como estimador al vector  $\mathbf{B}$ . El método de mínimos cuadrados asigna a  $\mathbf{B}$  el valor que minimiza

$$\sum_{k \in U} \left( \frac{y_k - \mathbf{x}_k^T \mathbf{B}}{c_k \sigma^2} \right)^2.$$

Usando el método de mínimos cuadrados, el estimador de  $\beta$  en la población finita  $U$  está dado por:

$$\mathbf{B} = [\mathbf{x} \Sigma^{-1} \mathbf{x}^T]^{-1} \mathbf{x} \Sigma^{-1} \mathbf{y} = \left[ \sum_{k \in U} \frac{\mathbf{x}_k \mathbf{x}_k^T}{c_k \sigma^2} \right]^{-1} \sum_{k \in U} \frac{\mathbf{x}_k y_k}{c_k \sigma^2} = \left[ \sum_{k \in U} \frac{\mathbf{x}_k \mathbf{x}_k^T}{c_k} \right]^{-1} \sum_{k \in U} \frac{\mathbf{x}_k y_k}{c_k},$$

donde

$$\mathbf{x} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \dots & \dots & \dots & \dots \\ x_{r1} & x_{r2} & \cdots & x_{rN} \end{pmatrix} = (\mathbf{x}_1, \dots, \mathbf{x}_N); \mathbf{y} = (y_1, \dots, y_N)^T; \Sigma = \begin{pmatrix} c_1 \sigma^2 & \cdots & 0 \\ 0 & \cdots & 0 \\ \dots & \dots & \dots \\ 0 & \cdots & c_N \sigma^2 \end{pmatrix}.$$

#### 3.9.2. Estimación en la muestra

En la práctica no se tiene acceso a todos los valores de la variable de interés, y en muchas ocasiones no se tiene acceso a todos los valores de la información auxiliar para cada elemento en la población finita. Para estimar el coeficiente de regresión, primero se

### 3.9. Muestreo de poblaciones finitas asistido por modelos

---

expresa  $\mathbf{B}$  como una función de totales, es decir,

$$\mathbf{B} = \mathbf{T}^{-1}\mathbf{t},$$

donde,  $\mathbf{T} = \sum_{k \in U} \frac{\mathbf{x}_k \mathbf{x}_k^T}{c_k}$  y  $\mathbf{t} = \sum_{k \in U} \frac{\mathbf{x}_k y_k}{c_k}$ .

**Resultado** Usando los principios de estimación de una función de totales, cuando el método de mínimos cuadrados es usado,  $\mathbf{B}$  es estimado por:

$$\widehat{\mathbf{B}} = \widehat{\mathbf{T}}^{-1}\widehat{\mathbf{t}},$$

donde  $\widehat{\mathbf{T}} = \sum_{k \in s} \frac{\mathbf{x}_k \mathbf{x}_k^T}{\pi_k c_k}$  y  $\widehat{\mathbf{t}} = \sum_{k \in s} \frac{\mathbf{x}_k y_k}{\pi_k c_k}$ .

Nótese que  $\widehat{\mathbf{T}}$  y  $\widehat{\mathbf{t}}$  son estimadores insesgados para  $\mathbf{T}$  y  $\mathbf{t}$ , respectivamente. Sin embargo,  $\widehat{\mathbf{B}}$  no es insesgado para  $\mathbf{B}$ .

#### 3.9.3. Estimador de regresión general

Al considerar el modelo (3.10) en general es posible construir un estimador del total poblacional que tome en cuenta esta relación. Se asume que existe una relación entre la variable de interés y la información auxiliar por el modelo de superpoblación  $\xi$ , es decir,

$$y_k = f(x_{1k}, \dots, x_{rk}) + \epsilon_k.$$

En particular, bajo  $\xi$  existe una relación de tipo lineal entre  $y_k$  y  $\mathbf{x}_k$ . Por tanto, en la población finita se tiene que:

$$y_k = \mathbf{x}_k^T \mathbf{B} + \epsilon_k = y_k^0 + \epsilon_k.$$

Entonces  $t_y = \sum_{k \in U} y_k$  se puede escribir como:

$$t_y = \sum_{k \in U} (y_k^0 + y_k - y_k^0) = \sum_{k \in U} \mathbf{x}_k^T \mathbf{B} + \sum_{k \in U} \epsilon_k.$$

Como el objetivo es estimar  $t_y$  con datos en la muestra. Entonces se tiene la definición del estimador de regresión general.

**Definición** El estimador de regresión general del total poblacional está definido como:

$$\widehat{t}_{y,greg} = \sum_{k \in U} \mathbf{x}_k^T \widehat{\mathbf{B}} + \sum_{k \in s} \frac{y_k - \mathbf{x}_k^T \widehat{\mathbf{B}}}{\pi_k}. \quad (3.11)$$

### 3.9. Muestreo de poblaciones finitas asistido por modelos

Desarrollando el término anterior, tenemos que:

$$\hat{t}_{y,greg} = \hat{t}_{y\pi} + \left( \mathbf{t}_x - \hat{\mathbf{t}}_{x\pi} \right)^T \hat{\mathbf{B}},$$

donde  $\mathbf{t}_x = \sum_{k \in U} \mathbf{x}_k$  es conocido y  $\hat{\mathbf{t}}_{x\pi} = \sum_{k \in s} \frac{\mathbf{x}_k}{\pi_k}$ .

Es posible mostrar que:

$$\hat{t}_{y,greg} = \sum_{k \in s} \left[ 1 + \left( \mathbf{t}_x - \hat{\mathbf{t}}_{x\pi} \right)^T \hat{\mathbf{T}}^{-1} \frac{\mathbf{x}_k}{c_k} \right] \frac{y_k}{\pi_k} = \sum_{k \in s} g_{ks} \frac{y_k}{\pi_k},$$

donde los valores  $g_{ks}$  se conocen como *g-weights*, los pesos  $g_{ks}$  tienen la propiedad de inducir estrategias representativas sobre cualquier variable del vector auxiliar.

**Propiedades del estimador de regresión general** Notemos que:

$$\hat{t}_{y,greg} = \sum_{k \in U} \mathbf{x}_k^T \hat{\mathbf{B}} + \sum_{k \in s} \frac{y_k - \mathbf{x}_k^T \hat{\mathbf{B}}}{\pi_k} = \sum_{k \in U} \hat{y}_k + \sum_{k \in s} \frac{e_k}{\pi_k}.$$

En algunas ocasiones el modelo  $\xi$  que relaciona la variable de interés y la información auxiliar es tal que:

$$\sum_{k \in s} \frac{e_k}{\pi_k} = 0.$$

Si la anterior condición se satisface, entonces  $\hat{t}_{y,greg}$  queda expresada como:

$$\hat{t}_{y,greg} = \sum_{k \in U} \hat{y}_k = \sum_{k \in U} \mathbf{x}_k^T \hat{\mathbf{B}} = \mathbf{t}_x^T \hat{\mathbf{B}}.$$

**Resultado** Una condición suficiente para que

$$\sum_{k \in s} \frac{e_k}{\pi_k} = 0,$$

es que exista un vector  $\mathbf{v}$  tal que  $\mathbf{v}^T \mathbf{x}_k = c_k$ .

(Sarndal *et al.*, 1992) menciona algunos ejemplos que satisfacen el resultado anterior:

1. Modelo de regresión lineal con intercepto  $x_{1k} = 1 \forall k \in U$  y  $c_k = 1$ .
2. Modelo de regresión lineal con estructura de varianza proporcional a alguna variable

### 3.9. Muestreo de poblaciones finitas asistido por modelos

del vector de información auxiliar. Es decir,

$$\sigma^2 c_k \propto x_{jk},$$

para algún  $j = 1, \dots, r$  y  $\forall k \in U$ .

3. Modelo de regresión lineal con estructura de varianza proporcional a una combinación lineal de las variables auxiliares. Es decir,

$$\sigma^2 c_k \propto \sum_{j=1}^r a_j x_{jk},$$

para algunas constantes  $a_1, \dots, a_r$  y  $\forall k \in U$ .

**Resultado** La varianza del estimador de regresión general para el total poblacional es:

$$Var(\hat{t}_{y,greg}) = \sum_{k,l \in U} \Delta_{kl} \frac{y_k - \mathbf{x}_k^T \mathbf{B}}{\pi_k} \frac{y_l - \mathbf{x}_l^T \mathbf{B}}{\pi_l},$$

Un estimador de la varianza anterior es:

$$\widehat{Var}(\hat{t}_{y,greg}) = \sum_{k,l \in s} \frac{\Delta_{kl}}{\pi_{kl}} \frac{y_k - \mathbf{x}_k^T \widehat{\mathbf{B}}}{\pi_k} \frac{y_l - \mathbf{x}_l^T \widehat{\mathbf{B}}}{\pi_l}.$$

(Sarndal *et al.*, 1992) proponen un estimador de la varianza que utiliza los *g-weights*  $g_{ks}$ . Notemos que:

$$\hat{t}_{y,greg} = \sum_{k \in U} \mathbf{x}_k^T \mathbf{B} + \sum_{k \in s} \frac{g_{ks} (y_k - \mathbf{x}_k^T \mathbf{B})}{\pi_k}.$$

Entonces,  $Var(\hat{t}_{y,greg}) = Var\left(\sum_{k \in s} \frac{g_{ks} E_k}{\pi_k}\right)$  con  $E_k = y_k - \mathbf{x}_k^T \mathbf{B}$ . Por lo tanto, un estimador de la varianza anterior es:

$$\widehat{Var}(\hat{t}_{y,greg}) = \sum_{k,l \in s} \frac{\Delta_{kl}}{\pi_{kl}} \frac{g_{ks} e_k}{\pi_k} \frac{g_{ls} e_l}{\pi_l},$$

donde  $e_k = y_k - \mathbf{x}_k^T \widehat{\mathbf{B}}$ .

(Sarndal *et al.*, 1992) mencionan que el papel que juega el modelo  $\xi$  se limita a la descripción, más no a la explicación, de la nube de puntos en la población finita. Argumentan que se espera que el modelo propuesto ajuste razonablemente bien y que haga pensar que pudo haber generado el comportamiento particular de la variable de interés. Nótese que el supuesto es flexible y no exige la certeza de que el modelo en verdad haya generado los valores de  $y$ . Por tanto, aunque el modelo induce aleatoriedad per se, las conclusiones de las estimaciones son independientes del mismo.

### 3.9. Muestreo de poblaciones finitas asistido por modelos

---

En este capítulo solamente se mencionaron algunos métodos de muestreo, existe una amplia literatura que describen otras técnicas de muestreo basadas en diseño, por ejemplo, ver (Cochran, 1977), (Sarndal *et al.*, 1992), (Lohr, 2000), (Brewer, 2002), (Tillé, 2006) y (Gutiérrez, 2009).

# Capítulo 4

## Regresión semiparamétrica en muestreo de poblaciones finitas

En este capítulo se describe y desarrolla el estimador del total poblacional con información auxiliar usando métodos semiparamétricos, basado en los trabajos de (Breidt y Opsomer, 2000) y (Breidt *et al.*, 2005). En la primera parte de este capítulo se desarrollan la teoría básica de los métodos de regresión local polinomial y el método de spline penalizados en el contexto de muestreo. Se replica el estudio de simulación realizado en (Breidt y Opsomer, 2000) y (Breidt *et al.*, 2005). Se pretende comparar los estimadores paramétricos con los semiparamétricos. Usaremos tres estimadores paramétricos, Horvitz-Thompson, Regresión Polinomial y Lineal, y dos estimadores semiparamétricos, Regresión Local Polinomial y Splines Penalizados. La comparación se hará por medio del cociente de razón de Errores Cuadráticos Medios con respecto al de Splines Penalizados.

### 4.1. Introducción

Es conveniente contar con métodos de estimación que hagan un uso eficiente de la información generada por una muestra y la información auxiliar. Si existe alguna relación conocida entre la información de la muestra y la información auxiliar, por ejemplo una relación lineal, dicha relación podría utilizarse para obtener estimadores eficientes de los parámetros poblacionales de interés, el cual es el enfoque de estimación paramétrica.

En el caso de no existir evidencia sobre alguna relación conocida, podemos utilizar métodos de estimación semiparamétricos, los cuales son independientes de la forma de la relación entre las variables de interés.

Los métodos semiparamétricos y no paramétricos son una herramienta estadística que



## 4.2. Regresión local polinomial

---

han ganado aceptación en muchas áreas de estadística. Estos hacen posible analizar datos, estimar tendencias y hacer inferencias sin tener completamente especificado un modelo paramétrico para los datos.

Los métodos no paramétricos son aquellos que no asumen una forma paramétrica en la modelación de los datos. En contraste, los métodos semiparamétricos usan una combinación de especificación paramétrica y no paramétrica en la modelación de los datos.

Para el caso de estimación semiparamétrica en muestreo de poblaciones finitas, se han hecho algunos avances. En el trabajo de (Breidt y Opsomer, 2000) se ha desarrollado el estimador del total por regresión localmente polinomial. Además en (Breidt *et al.*, 2005) proponen un estimador para encuestas complejas utilizando splines penalizados.

Los métodos de estimación semiparamétrica en muestreo antes citados están orientados principalmente para aplicarse al caso de una sola variable auxiliar. Cuando se tienen dos o más variables es necesario buscar otras alternativas. Una de estas alternativas puede ser la estimación por regresión con coeficientes variantes, el cual se desarrolla en el Capítulo 6.

## 4.2. Regresión local polinomial

Supongamos que se tienen  $N$  pares de observaciones  $(x_1, y_1), \dots, (x_N, y_N)$ . Consideremos que existe una relación entre las variables, de acuerdo al modelo de superpoblación:

$$y_i = m(x_i) + e_i,$$

donde  $m(\cdot)$  es una función suave pero se desconoce y  $e_i$  son los residuales inherentes al modelo. Una de las formas de estimar  $m(\cdot)$  es usando regresión local polinomial.

Antes de estimar  $m(\cdot)$ , aproximemos la función analítica  $m(\cdot)$  como una serie de Taylor de orden  $p$ , donde  $p$  se fija normalmente con  $p < 4$ , y  $x_j$  está en una vecindad de  $x_i$ :

$$m(x_j) = \sum_{k=0}^p \beta_k (x_j - x_i)^k, \quad (4.1)$$

donde  $\beta_k = \frac{m^{(k)}(x_i)}{k!}$ ,  $k = 0, \dots, p$ . Note que  $m(x_i)$  está representado por  $\beta_0$ . Para un valor dado  $x_i$ , el estimador  $\hat{m}(x_i)$  es definido como  $\hat{\beta}_0$ , donde los  $\hat{\beta} = (\hat{\beta}_0, \dots, \hat{\beta}_p)^T$  son encontrados resolviendo el siguiente problema de mínimos cuadrados ponderados:

$$\min_{\beta} \sum_{j=1}^N K \left( \frac{x_j - x_i}{h} \right) \left( y_j - \sum_{k=0}^p \beta_k (x_j - x_i)^k \right)^2, \quad (4.2)$$

## 4.2. Regresión local polinomial

donde  $K\left(\frac{x_j - x_i}{h}\right) = \frac{1}{h}g\left(\frac{x_j - x_i}{h}\right)$  denota la función Kernel,  $h > 0$  es el ancho de banda y  $g$  es una función de densidad simétrica alrededor de cero tales como Uniforme, Triangular, Epanechnikov, Tricúbica y Gaussiano. La función  $g$  determina qué peso se asigna a cada observación dependiendo de su distancia a  $x_i$ . Para más detalle sobre el método de regresión local polinomial, ver (Stone, 1977), (Cleveland, 1979), (Wand y Jones, 1995), (Fan y Gijbels, 1996), (Loader, 1999) y (Ruppert *et al.*, 2003).

La idea de un ajuste local polinomial es simple: en una vecindad de  $x_i$  se ajusta un polinomio de grado  $p$  a los datos y los  $p+1$  coeficientes del ajuste conducen a estimadores de  $\beta_k$ , o equivalentemente a,

$$\widehat{m}_k = k! \widehat{\beta}_k, \quad k = 0, 1, \dots, p,$$

que es un estimador de  $m^{(k)}$ , derivada  $k$ -ésima de la función de regresión evaluada en  $x_i$ .

La ecuación (4.2) se puede tener de manera matricial como:

$$\min_{\beta} (Y - X_{U_i} \beta)^T W_{U_i} (Y - X_{U_i} \beta), \quad (4.3)$$

donde,  $Y = (y_1, \dots, y_N)^T$ ,  $W_{U_i} = \text{diag} \left\{ K\left(\frac{x_1 - x_i}{h}\right), \dots, K\left(\frac{x_N - x_i}{h}\right) \right\}$  y

$$X_{U_i} = \begin{pmatrix} 1 & (x_1 - x_i) & \cdots & (x_1 - x_i)^p \\ \vdots & \vdots & \ddots & \vdots \\ 1 & (x_N - x_i) & \cdots & (x_N - x_i)^p \end{pmatrix} = [1, x_j - x_i, \dots, (x_j - x_i)^p]_{j \in U}.$$

Entonces la solución al problema de minimización (4.2) es:

$$\widehat{\beta} = (X_{U_i}^T W_{U_i} X_{U_i})^{-1} X_{U_i}^T W_{U_i} Y. \quad (4.4)$$

Por lo tanto, la estimación de la función  $m(x_i)$  está dada por:

$$m_i = e_1^T (X_{U_i}^T W_{U_i} X_{U_i})^{-1} X_{U_i}^T W_{U_i} Y, \quad (4.5)$$

donde  $e_1 = (1, 0, \dots, 0)^T$  y  $m_i$  está bien definida, siempre y cuando  $X_{U_i}^T W_{U_i} X_{U_i}$  es invertible.

Si todas las  $m_i$  son conocidas, entonces un estimador insesgado del total poblacional  $t_y = \sum_{i \in U} y_i$  sería el estimador de diferencia generalizada (ver Capítulo 3, (Sarndal *et al.*, 1992)):

$$t_y^* = \sum_{i \in U} m_i + \sum_{i \in s} \frac{y_i - m_i}{\pi_i}. \quad (4.6)$$

## 4.2. Regresión local polinomial

---

La varianza diseño del estimador (4.6) es

$$\text{Var}(t_y^*) = \sum_{i,j \in U} (\pi_{ij} - \pi_i \pi_j) \frac{y_i - m_i}{\pi_i} \frac{y_j - m_j}{\pi_j}.$$

En el contexto de muestreo de poblaciones finitas,  $m_i$  no puede ser calculada, ya que solamente se tiene una muestra  $s$  de  $U$  para la variable de interés  $y$ . Por lo tanto, se reemplazan los  $m_i$  por un estimador consistente basado en la muestra  $s$ . Entonces, para un escalar  $x_i$  y una muestra observada  $s$  bajo algún diseño de muestreo, sea  $y_s = [y_i]_{i \in s}$  y se define la matriz de diseño local como:

$$X_{si} = [1, (x_j - x_i), \dots, (x_j - x_i)^p]_{j \in s},$$

y la matriz diagonal ponderada  $W_{si} = \text{diag} \left\{ \frac{1}{\pi_j h} K \left( \frac{x_j - x_i}{h} \right) \right\}_{j \in s}$ . Entonces el estimador muestral de  $m(x_i)$  de la regresión localmente polinomial es dado por:

$$\hat{m}_i = e_1^T (X_{si}^T W_{si} X_{si})^{-1} X_{si}^T W_{si} y_s. \quad (4.7)$$

siempre que  $X_{si}^T W_{si} X_{si}$  sea invertible. Si se sustituyen los  $\hat{m}_i$  en (4.6) se obtiene el estimador de regresión local polinomial para el total poblacional

$$\hat{t}_y = \sum_{i \in s} \frac{y_i - \hat{m}_i}{\pi_i} + \sum_{i \in U} \hat{m}_i. \quad (4.8)$$

(Breidt y Opsomer, 2000) afirma que el estimador (4.7) difiere significativamente del estimador de regresión local polinomial tradicional. La presencia de las probabilidades de inclusión en  $W_{si}$  hacen que el estimador  $\hat{m}_i$  basado en la muestra, sea un estimador consistente bajo el diseño, con un ancho de banda  $h$  fijo y no necesariamente óptimo. (Opsomer y Miller, 2005) presentan una manera de seleccionar el ancho de banda  $h$  y estudian algunas propiedades asintóticas.

En principio, el estimador (4.7) puede no estar definido para cierto  $i \in U$ : si para alguna muestra  $s$  hay menos de  $p + 1$  observaciones en el dominio de definición del kernel para un  $x_i$  entonces la matriz  $X_{si}^T W_{si} X_{si}$  sería singular. Esto no es un problema en la práctica, ya que se puede seleccionar un ancho de banda lo suficientemente grande para que  $X_{si}^T W_{si} X_{si}$  sea invertible para todos los  $x_i$ . Sin embargo, esta situación no puede ser excluida teóricamente mientras el ancho de banda se considere fijo para una determinada población. A continuación se considera un estimador muestral que existe para cualquier muestra  $s$  de  $U$ .

### 4.3. Regresión por Splines Penalizados

El ajuste del estimador muestral para  $m_i$  está dado por

$$\tilde{m}_i = e_1^T \left( X_{si}^T W_{si} X_{si} + \left\{ \frac{\delta}{N^2} \right\}_{j=1}^{p+1} \right)^{-1} X_{si}^T W_{si} y_s, \quad (4.9)$$

para  $\delta > 0$ . El término  $\delta/N^2$  garantiza que el estimador este bien definido para todo  $s \subset U$ . Este ajuste también fue usado por (Fan, 1993). Otro posible ajuste consistiría en reemplazar la elección habitual del kernel (como puede ser un kernel Gaussiano). En la práctica esta elección sólo aumenta la complejidad computacional del estimador lineal. Finalmente el estimador de regresión polinómico local para (4.9) está dado por

$$\tilde{t}_y = \sum_{i \in s} \frac{y_i - \tilde{m}_i}{\pi_i} + \sum_{i \in U} \tilde{m}_i. \quad (4.10)$$

En (Breidt y Opsomer, 2000) se estudian otras propiedades estadísticas del estimador del total poblacional (4.10).

### 4.3. Regresión por Splines Penalizados

A continuación se describe el método de estimación semiparamétrica de splines penalizados en el contexto de muestreo de poblaciones finitas. Para más detalle de este método en un contexto general, ver (O'Sullivan, 1986), (Eilers y Marx, 1996), (Hastie, 1996), (Härdle, 1990), (Eubank, 1994), (Eubank, 1999), (Green y Silverman, 1994) y (Ruppert *et al.*, 2003).

Supongamos que se tienen  $N$  pares de observaciones  $(x_1, y_1), \dots, (x_N, y_N)$  y asumamos que siguen el modelo de superpoblación:

$$y_i = m(x_i) + e_i.$$

Ahora adicionemos el supuesto de que  $m(\cdot)$  es bien aproximada por una función spline de orden  $p$ . Las funciones splines polinomiales son definidas como:

$$m(x; \beta) = \sum_{k=0}^p \beta_k x^k + \sum_{j=1}^K \beta_{p+j} (x - \gamma_j)_+^p, \quad (4.11)$$

donde  $(x - \gamma_j)_+^p = (x - \gamma_j)^p$  si  $x > \gamma_j$  y 0 en otro caso. Los valores preestablecidos  $\gamma_1, \dots, \gamma_K$  son llamados nodos. La flexibilidad del modelo se obtiene por los polinomios truncados de grado  $p$ . El número  $K$  y la localización de los nodos se asumen fijos. (Ruppert *et al.*, 2003) menciona que una regla empírica para determinar  $K$  es  $\min \left\{ \frac{N}{4}, 35 \right\}$  y los nodos  $\gamma_j$  se pueden distribuir de manera uniforme como los cuantiles de los  $x_i$ ; esto

### 4.3. Regresión por Splines Penalizados

es,  $\gamma_j$  es igual al  $\frac{j}{K+1}$ -ésimo cuantil poblacional de los valores únicos  $x_i$ .

Sea  $\beta = (\beta_0, \dots, \beta_p, \beta_{p+1}, \dots, \beta_{p+K})^T$  el vector de parámetros. Entonces, el estimador de  $m(\cdot)$  es  $m(\cdot; \mathbf{B}_U)$ , donde  $\mathbf{B}_U$  es el vector que minimiza

$$\sum_{i=1}^N \left( y_i - \sum_{k=0}^p \beta_k x_i^k - \sum_{j=1}^K \beta_{p+j} (x_i - \gamma_j)_+^p \right)^2 + \lambda \sum_{j=1}^K \beta_{p+j}^2, \quad (4.12)$$

donde  $\lambda > 0$  es conocido como parámetro de suavizamiento. La suavidad del ajuste resultante depende del valor usado para  $\lambda$ . En el Capítulo 6 se propone una forma de seleccionar múltiples parámetros de suavizamiento basado en la idea de (Opsomer y Miller, 2005). La ecuación (4.12) de manera matricial se expresa como:

$$(Y - X\beta)^T (Y - X\beta) + \beta^T A_\lambda \beta, \quad (4.13)$$

donde  $Y = (y_1, \dots, y_N)^T$ , la matriz diagonal  $A_\lambda = \text{diag}\{0, \dots, 0, \lambda, \dots, \lambda\}$  con  $p+1$  ceros seguidos de  $K$  elementos iguales a  $\lambda$ .

Sea  $X$  la matriz de diseño para la regresión spline tal que la  $i$ -ésima fila de  $X$  es  $X_i^T = [1, x_i, \dots, x_i^p, (x_i - \gamma_1)_+^p, \dots, (x_i - \gamma_K)_+^p]$  para  $i \in U$ . Es posible demostrar que la solución al problema de minimización en (4.13) es

$$\mathbf{B}_U = (X^T X + A_\lambda)^{-1} X^T Y. \quad (4.14)$$

Se definen los grados de libertad del modelo de spline de orden  $p$  como:

$$df(\lambda) = \text{Traza} \left[ (X^T X + A_\lambda)^{-1} X^T Y \right]. \quad (4.15)$$

Sea  $m_i = m(x_i; \mathbf{B}_U) \equiv X_i^T \mathbf{B}_U$ , para  $i \in U$ . Si estos valores son conocidos, entonces pueden ser usado en la estimación del total poblacional usando el estimador de diferencia (Sarndal *et al.*, 1992):

$$\hat{t}_{y,diff} = \sum_{i \in U} m_i + \sum_{i \in s} \frac{y_i - m_i}{\pi_i}. \quad (4.16)$$

La varianza diseño del anterior estimador es:

$$\text{Var}(\hat{t}_{y,diff}) = \sum_{i,j \in U} (\pi_{ij} - \pi_i \pi_j) \frac{y_i - m_i}{\pi_i} \frac{y_j - m_j}{\pi_j}. \quad (4.17)$$

De la expresión (4.17), la eficiencia de  $\hat{t}_{y,diff}$  depende de si el modelo de superpoblación es una representación apropiada para la población. Entonces  $\hat{t}_{y,diff}$  será más eficiente que el estimador de Horvitz-Thompson  $\hat{t}_{y\pi}$ .

El estimador (4.16) es inviable, ya que los valores  $m_i$  no pueden ser calculados. Sin

#### 4.4. Estudio de Simulación

embargo, dado una muestra  $s$  de  $U$ . Definamos  $X_s = [X_i^T]_{i \in s}$  y  $W_s = \text{diag}[1/\pi_i]_{i \in s}$ . El estimador de  $m(x_i)$  por splines a partir de una muestra  $s$  y basado en el diseño de muestreo es

$$\hat{m}_i \equiv X_i^T \hat{\mathbf{B}}_U = X_i^T (X_s^T W_s X_s + A_\lambda)^{-1} X_s^T W_s Y_s = X_i^T \mathbf{G}_\lambda Y_s. \quad (4.18)$$

Entonces el estimador de spline penalizado asistido por el modelo, del total poblacional  $t_y$  para una muestra observada  $s$  es

$$\hat{t}_{y,spl} = \sum_{i \in U} \hat{m}_i + \sum_{i \in s} \frac{y_i - \hat{m}_i}{\pi_i}. \quad (4.19)$$

Si se definen la función indicadora  $I_i = 1$  si  $i \in s$  y  $I_i = 0$  en otro caso, y el vector indicador  $e_i$  es un vector de ceros y un uno en la posición  $i$ , entonces (4.19) se puede reescribir como:

$$\hat{t}_{y,spl} = \sum_{i \in s} \frac{y_i}{\pi_i} + \sum_{i \in U} \left(1 - \frac{I_i}{\pi_i}\right) X_i^T \mathbf{G}_\lambda Y_s = \sum_{i \in s} \left\{ \frac{1}{\pi_i} + \sum_{j \in U} \left(1 - \frac{I_j}{\pi_j}\right) X_j^T \mathbf{G}_\lambda e_i \right\} y_i. \quad (4.20)$$

De igual manera se puede demostrar que (Breidt *et al.*, 2005):

$$\hat{t}_{y,spl} = t_X^T \mathbf{B}_U = \sum_{i \in U} \hat{m}_i. \quad (4.21)$$

La ecuación (4.21) se obtiene observando que  $\hat{t}_{y,spl} = \hat{t}_{y\pi} + (t_X - \hat{t}_{X\pi})^T \mathbf{B}_U$ , donde  $t_X = \sum_{i \in U} X_i$  y  $\hat{t}_{X\pi} = \sum_{i \in s} \frac{X_i}{\pi_i}$ . Definamos  $e_1 = (1, 0, \dots, 0)^T$ , entonces se tiene que  $e_1^T X_i = 1$  y  $e_1^T A_\lambda = (0, 0, \dots, 0)$ . Entonces:

$$\hat{t}_{X\pi}^T \mathbf{B}_U = \left( \sum_{i \in s} \frac{e_1^T X_i X_i^T}{\pi_i} + e_1^T A_\lambda \right) \left( \sum_{j \in s} \frac{X_j X_j^T}{\pi_j} + A_\lambda \right)^{-1} \sum_{j \in s} \frac{X_j y_j}{\pi_j} = \hat{t}_{y\pi}. \quad (4.22)$$

En (Breidt *et al.*, 2005) estudian otras propiedades estadísticas del estimador del total poblacional (4.21).

#### 4.4. Estudio de Simulación

En esta sección se hace el estudio de simulación comparando los estimadores paramétricos y semiparamétricos para la estimación del total poblacional  $t_y$  usando una variable auxiliar. Se usan como estimadores paramétricos el de Regresión polinomial, lineal y el de Horvitz-Thompson; mientras que en el caso semiparamétrico se usan el de regresión local polinomial y el de splines penalizados.

#### 4.4. Estudio de Simulación

Todos los estimadores serán elegidos teniendo el mismo grado de libertad y el parámetro de suavizamiento es elegido de tal forma que la traza de la matriz de suavizamiento es el grado de libertad. Para el método de splines usaremos  $K = 25$  nodos.

En la regresión local polinomial usamos el kernel de Epanechnikov definido como  $K(t) = 0.75(1 - t^2)I_{|t| \leq 1}$ .

Consideramos las siguientes funciones de regresión:

$$\begin{aligned}
 \text{Lineal: } & m_1(x) = 1 + 2(x - 0.5), \\
 \text{Cuadrático: } & m_2(x) = 1 + 2(x - 0.5)^2, \\
 \text{Bump: } & m_3(x) = 1 + 2(x - 0.5) + \exp(-200(x - 0.5)^2), \\
 \text{Salto: } & m_4(x) = 1 + 2(x - 0.5)I_{x \leq 0.65} + 0.65I_{x > 0.65}, \\
 \text{F. D. A.: } & m_5(x) = \Phi\left(\frac{1.5-2x}{\sigma}\right), \text{ donde } \Phi \text{ es una f.d.a. } N(0,1), \\
 \text{Exponencial: } & m_6(x) = \exp(-8x), \\
 \text{Ciclo 1: } & m_7(x) = 2 + \sin(2\pi x), \\
 \text{Ciclo 4: } & m_8(x) = 2 + \sin(8\pi x).
 \end{aligned}$$

donde  $x \in [0, 1]$ . La población  $x_k$  son generadas como una muestra aleatoria de la distribución uniforme  $(0, 1)$ . Los valores poblacionales  $y_{ik}$  ( $i = 1, \dots, 8$ ) son generadas de las funciones de regresión añadiendo errores  $N(0, \sigma^2)$  en todos los casos, excepto en la F.D.A. La población simulada será de tamaño  $N = 1000$ .

Las muestras son generadas bajo dos tipos de diseños (*MSR* y *STSR*) con tamaños de muestra  $n = 50$ . En el diseño *STSR* usaremos 7 estratos de tamaños iguales. Los estratos son generados a partir de los cuantiles de los  $x_k$  generados. Para cada combinación de funciones de regresión, desviación estándar, grados de libertad y diseño de muestreo, 1000 muestras replicadas son seleccionadas y los estimadores del total son calculados como sigue: para los estimadores de regresión se usa la ecuación (3.11) y para el estimador de HT se utiliza la ecuación (3.2). En el caso semiparamétrico se utilizan las ecuaciones (4.10) y (4.21).

Como la población es fijada en las  $N = 1000$  réplicas, podemos evaluar la eficiencia, para lo cual calculamos los errores cuadráticos medios. Estos errores cuadráticos medios son comparados como un cociente con respecto al error cuadrático medio del modelo de regresión por splines penalizados.

La Tabla 4.1 muestra las razones de ECM obtenidos en el proceso de simulación, donde el  $ECM(\hat{t}_y)$  se define como:

$$ECM(\hat{t}_y) = \frac{1}{N} \sum_{i=1}^N (\hat{t}_y^{(i)} - t_y)^2. \quad (4.23)$$

Para el caso de los estimadores de Regresión Local Lineal el kernel usado es el de Epanech-

## 4.5. Conclusiones

---

nikov y ancho de banda  $h = 0.2943$  cuando  $df = 4$  y  $h = 0.135$  cuando  $df = 7$ .

## 4.5. Conclusiones

Los experimentos de simulación muestran evidencia empírica que los estimadores semi-paramétricos son más eficientes que los estimadores de regresión paramétricos cuando el modelo paramétrico se especifica incorrectamente y son aproximadamente eficientes cuando la especificación paramétrica es la correcta. En muchos casos el método de splines penalizados es competitivo o mejor que los estimadores paramétricos.



## 4.5. Conclusiones

**Tabla 4.1:** Razón de ECM relativo al estimador de spline penalizado, basado en 1000 réplicas de tamaños de muestra  $n = 50$  de una población fija de tamaño  $N = 1000$ .

Modelo	$\sigma$	g.l.	Reg. Pol. MSR STSR	Reg. Lin. MSR STSR	H-T MSR STSR	Reg. Lin. Local MSR STSR
LINEAL	0.1	4	1.02 1.00	0.98 1.00	28.15 1.72	1.01 1.00
	0.1	7	1.20 1.11	0.88 0.99	32.92 1.93	1.27 1.05
	0.4	4	1.02 1.00	0.97 1.00	2.85 1.04	1.01 1.00
	0.4	7	1.23 1.09	0.95 0.97	2.77 1.03	1.20 1.02
CUADRÁTICO	0.1	4	1.00 1.00	3.10 1.22	2.95 1.17	1.01 0.99
	0.1	7	1.21 1.09	3.07 1.19	2.94 1.18	1.32 1.03
	0.4	4	1.03 1.01	1.14 1.02	1.11 1.02	1.03 1.00
	0.4	7	1.32 1.10	0.99 0.99	0.97 0.98	1.29 1.04
BUMP	0.1	4	1.40 1.16	1.67 1.07	8.02 1.61	1.06 1.05
	0.1	7	2.28 1.79	3.13 1.20	14.18 1.64	1.02 1.02
	0.4	4	1.10 1.03	1.26 0.99	2.68 1.03	1.03 1.00
	0.4	7	1.27 1.11	1.12 0.99	2.69 1.02	1.05 1.03
SALTO	0.1	4	1.34 1.10	3.07 1.16	3.87 1.16	1.03 1.03
	0.1	7	1.79 1.45	4.41 1.46	5.29 1.45	1.02 1.00
	0.4	4	1.05 1.02	1.32 1.05	1.44 1.04	1.02 1.01
	0.4	7	1.27 1.14	1.32 1.02	1.46 1.02	1.12 1.03
F. D. A.	0.1	4	1.17 1.10	1.97 1.09	3.76 1.09	0.98 0.97
	0.1	7	1.69 1.28	2.26 1.18	4.81 1.17	1.02 1.01
	0.4	4	1.02 1.00	1.03 0.99	1.27 1.00	1.00 1.00
	0.4	7	1.33 1.07	0.96 1.00	1.17 1.00	1.13 1.02
EXPONENCIAL	0.1	4	0.88 0.93	2.16 1.29	4.04 1.38	0.92 0.95
	0.1	7	1.26 1.03	2.50 1.54	4.45 1.67	0.99 1.00
	0.4	4	1.02 1.00	1.08 1.01	1.32 1.02	1.02 1.00
	0.4	7	1.26 1.08	1.03 1.02	1.11 1.03	1.00 1.03
CICLO 1	0.1	4	0.27 0.69	3.50 2.75	8.38 2.39	0.84 0.78
	0.1	7	1.06 1.01	13.93 4.43	31.39 3.84	0.99 1.02
	0.4	4	0.77 1.00	1.76 1.17	3.07 1.17	0.96 0.99
	0.4	7	1.29 1.06	1.93 1.28	3.47 1.21	0.99 1.02
CICLO 4	0.1	4	1.03 1.04	1.01 1.00	1.04 1.01	1.07 1.10
	0.1	7	2.25 2.23	1.37 1.31	1.38 1.31	0.99 0.84
	0.4	4	1.01 1.03	1.05 1.00	1.05 1.01	1.03 1.06
	0.4	7	1.88 1.95	1.19 1.14	1.21 1.13	0.99 0.87

# Capítulo 5

## El efecto sobre la estimación del total poblacional de la estimación basada en la muestra del parámetro de suavizamiento de regresión por spline penalizado en muestreo de poblaciones finitas

En este capítulo se muestra que no hay una pérdida significativa en la precisión del estimador del total poblacional si calculamos el parámetro de suavizamiento con información de la muestra en lugar de utilizar toda la información de la población. El parámetro de suavizamiento es obtenido fijando los grados de libertad y resolviendo una ecuación basado únicamente en datos de la muestra. Se presentan los resultados de un estudio de simulación de la determinación del parámetro de suavizamiento. Además, se evaluó el efecto de la propuesta de determinación del parámetro de suavizamiento en la estimación del total poblacional usando una población real. Para esta población, los experimentos de simulación muestran que la determinación del parámetro de suavizamiento, ya sea con datos de la muestra o con los datos de toda la población, produce errores relativos similares en la estimación del total poblacional.

### 5.1. Introducción

En los últimos años los métodos de estimación semiparamétrica han ganado una considerable atención debido a su flexibilidad. Uno de estos métodos es la estimación por splines

## 5.1. Introducción

---

penalizados. Para utilizar este método se deben determinar el número y localización de los nodos, el grado del polinomio y los grados de libertad. En la literatura a los grados de libertad se les conoce también como el número equivalente de parámetros. En la literatura existen reglas prácticas para determinar el grado del polinomio y el número y localización de los nodos (ver (Ruppert *et al.*, 2003)). Los grados de libertad se establecen de acuerdo a la experiencia del usuario, por ejemplo ver la aplicación en (Breidt *et al.*, 2005).

Existe una relación funcional entre los grados de libertad y el parámetro de suavizamiento, véase la ecuación (5.5). En este trabajo se muestra el efecto del parámetro de suavizamiento basado en una muestra en la precisión del total poblacional.

La determinación del parámetro de suavizamiento en un contexto general ha sido ampliamente explorado en la literatura. (Wand y Scott, 1994) usan validación cruzada, (Eilers y Marx, 1996) usan validación cruzada generalizada en sus ejemplos, (Schwarz, 1978) usa el criterio de información bayesiana, y (Hurvich *et al.*, 1998) utiliza una versión mejorada del Criterio de Información de Akaike que se construye para elegir el parámetro de suavizamiento para suavizadores semiparamétricos lineales. Los anteriores métodos utilizan toda la información poblacional de las variables involucradas. El método propuesto en este trabajo para la determinación del parámetro de suavizamiento utiliza únicamente la información de la muestra seleccionada de la población y este parámetro de suavizamiento se utiliza para calcular el total poblacional.

El método utilizado para determinar el parámetro de suavizamiento es el siguiente. En primer lugar, se determinan el grado del polinomio, el número y localización de los nodos. Entonces, con los datos obtenidos a partir de una muestra, se estima la matriz de proyección del correspondiente modelo de splines. Por último, se fijan los grados de libertad y se resuelve para el parámetro de suavizamiento la ecuación que relaciona los grados de libertad con dicho parámetro (véase la ecuación (5.10)). El objetivo de este trabajo es mostrar que no hay una pérdida significativa en la precisión del estimador del total poblacional, si se calcula el parámetro de suavizamiento con información de la muestra.

La regresión por splines penalizados ha sido adaptada en el contexto de muestreo de poblaciones finitas. Esta metodología fue propuesta en el trabajo de (Breidt *et al.*, 2005). Este trabajo propuso un estimador del total poblacional y sus propiedades estadísticas basados en un modelo de splines penalizados. En el mismo artículo se aplicó la metodología a una muestra de una población real y el parámetro de suavizamiento correspondiente se determinó de acuerdo a la experiencia de los autores. La metodología de estimación por splines y otros métodos semiparamétricos en muestreo de poblaciones finitas se han tratado recientemente en los trabajos, por ejemplo, en (Rueda *et al.*, 2010), (Montanari y Ranalli, 2005), (Opsomer *et al.*, 2008), (Wang, 2009), (Wang y Wang, 2011), (Rueda y Sánchez-Borrego, 2009), (Breidt y Opsomer, 2000) y (Opsomer y Miller, 2005).

En la siguiente sección, se revisan algunos conceptos de regresión por splines penalizados en el contexto de muestreo y se describe como se determina el parámetro de suavizamiento.

to.

## 5.2. Selección del parámetro de suavizamiento basada en la muestra

En la primera parte de esta sección se revisa la metodología de estimación por en splines penalizados. Gran parte de esta revisión está basada en el trabajo de (Breidt *et al.*, 2005). En la segunda parte se describe el método propuesto para la determinación del parámetro de suavizamiento basado en la información de la muestra.

Denotaremos por  $U = \{1, \dots, N\}$  a la población bajo estudio. Sean  $y_1, \dots, y_N$  los valores desconocidos de la característica de interés y  $x_1, \dots, x_N$  los valores conocidos de la variable auxiliar. Adicionalmente, se supone que existe una relación entre la variable interés y la variable auxiliar. Esta relación se expresa por:

$$y_i = f(x_i) + \epsilon_i, \quad i = 1, \dots, N, \quad (5.1)$$

donde los  $\epsilon_i$  son la parte residual del modelo. Esto es, las variables  $\epsilon_i$  absorben el comportamiento de las  $y_i$  que no es explicado por el modelo  $f$ . En el contexto de estimación semiparamétrica, la función  $f$  es desconocida.

En este trabajo se supone que la función  $f$  se puede aproximar adecuadamente por un modelo de spline de orden  $p$ , es decir:

$$f(x) \approx m(x; \beta) = \sum_{k=0}^p \beta_k x^k + \sum_{j=1}^J \beta_{p+j} (x - \gamma_j)_+^p, \quad (5.2)$$

donde  $(x - \gamma_j)_+^p = (x - \gamma_j)^p$  si  $x > \gamma_j$  y 0 en otro caso. Adicionalmente, las constantes  $\gamma_j$ ,  $j = 1, \dots, J$ , representan los nodos del modelo. El vector  $\beta = (\beta_0, \dots, \beta_{p+J})^T$  representa al vector de coeficientes del modelo de splines.

Un método de estimación del vector de parámetros  $\beta$  utilizado comúnmente en la literatura es el de mínimos cuadrados penalizados. Esto es, los valores estimados de estos parámetros son los que minimizan la siguiente función:

$$\sum_{i \in U} \{y_i - m(x_i; \beta)\}^2 + \lambda \sum_{j=1}^J \beta_{p+j}^2. \quad (5.3)$$

La constante  $\lambda \geq 0$  se le conoce como el parámetro de penalización, porque este parámetro penaliza a los coeficientes de los polinomios truncados. Sin embargo, a  $\lambda$  se le conoce también como parámetro de suavizamiento porque esta cantidad determina el grado de

## 5.2. Selección del parámetro de suavizamiento basada en la muestra

suavidad de la curva estimada. El parámetro de suavizamiento se determina mediante el uso de la información de la muestra y se muestra que el efecto sobre la estimación del total no es significativo, en el contexto de muestreo de poblaciones finitas.

Ahora, sea  $X$  la matriz diseño con filas

$$X_i = \{1, x_i, \dots, x_i^p, (x_i - \gamma_1)_+^p, \dots, (x_i - \gamma_J)_+^p\}$$

para  $i \in U$ , y sea  $Y = (y_1, \dots, y_N)^T$ . Adicionalmente, sea  $A_\lambda = \text{diag}\{0, \dots, 0, \lambda, \dots, \lambda\}$  la matriz diagonal con  $p+1$  ceros sobre la diagonal seguido por  $J$  elementos iguales a  $\lambda$ .

Si la población  $U$  es totalmente observada, el vector de coeficientes  $\beta$  que minimiza la expresión (5.3) es:

$$\beta_U = (X^T X + A_\lambda)^{-1} X^T Y. \quad (5.4)$$

Para el modelo (5.2) y el estimador (5.4), los grados de libertad queda dado por:

$$\ell_U = \text{Traza} \left[ (X^T X + A_\lambda)^{-1} X^T X \right]. \quad (5.5)$$

Por otra parte, en el muestreo de poblaciones finitas, uno de los parámetros a estimar es el total de los valores de la variable de interés:  $t_y = \sum_{k \in U} y_k$ . Si la aproximación (5.2) es adecuada para la función  $f$ , entonces el anterior total se puede escribir como:

$$t_y = \sum_{k \in U} m(x_k) + \sum_{k \in U} \epsilon_k. \quad (5.6)$$

La expresión (5.6) del total sugiere un estimador para este parámetro dado por:

$$\hat{t}_y = \sum_{k \in U} \widehat{m(x_k)} + \sum_{k \in U} \widehat{\epsilon_k}. \quad (5.7)$$

Ahora bajo el contexto de muestreo, sea  $s = \{k_1, \dots, k_n\}$  una muestra de la población  $U$ . Además, un estimador para el parámetro  $\beta_U$  en (5.4) está dado por:

$$\hat{\beta}_U = (X_s^T W_s X_s + A_\lambda)^{-1} X_s^T W_s Y_s, \quad (5.8)$$

donde  $W_s = \text{diag}_{j=1, \dots, n} \{1/\pi_{k_j}\}$ ,  $X_s$  es la matriz de tamaño  $N \times (p+1+J)$  cuyos renglones son  $X_{k_j}$ ,  $Y_s = [y_{k_j}]$  y  $k_j \in s$ .

En (Breidt *et al.*, 2005) se demostró que el estimador (5.7), utilizando el estimador (5.8) de  $\beta_U$ , se reduce a

$$\hat{t}_y = t_x \hat{\beta}_U, \quad (5.9)$$

### 5.3. Estudio de simulación

---

donde  $t_x = \sum_{i \in U} X_i$ .

El estimador (5.9) depende del parámetro de suavizamiento, entonces en este trabajo el parámetro de suavizamiento es calculado a partir de una muestra seleccionada y se muestra más adelante el efecto que esta tiene en la estimación del total poblacional. Para esto, si se determinan el grado del polinomio del spline, los grados de libertad, el número y localización de los nodos y si se tiene la información completa de la variable auxiliar, entonces se podría estimar el parámetro de suavizamiento resolviendo la igualdad (5.5) para  $\lambda$ . Nuestra experiencia nos ha mostrado que en ocasiones resolver la ecuación (5.5) produce problemas de inestabilidad numérica para ciertos grados de libertad. Esta inestabilidad numérica se presenta cuando se pretende calcular la inversa  $(X^T X + A_\lambda)$ . En estos casos de inestabilidad numérica, proponemos resolver para el parámetro de suavizamiento,  $\lambda_s$ , la siguiente igualdad:

$$\ell = \text{Traza} \left[ (X_s^T W_s X_s + A_{\lambda_s})^{-1} X_s^T W_s X_s \right]. \quad (5.10)$$

Obsérvese que la forma de la anterior igualdad es similar a (5.5) salvo que el parámetro de suavizamiento se determina únicamente con la información de la variable auxiliar de la muestra seleccionada. Esta estrategia de determinación del parámetro de suavizamiento pretende evitar el problema de inestabilidad numérica antes mencionado. Note también que se puede utilizar el mismo parámetro de suavizamiento obtenido en (5.10) para estimar el total de varias variables de interés, relacionadas con la misma variable auxiliar.

En la siguiente sección se describe el algoritmo propuesto para la determinación del parámetro de suavizamiento. De igual manera se realizan experimentos de simulación para estudiar el comportamiento de la propuesta.

### 5.3. Estudio de simulación

En esta sección se describe el algoritmo de simulación para estudiar el comportamiento de la determinación del parámetro de suavizamiento. Adicionalmente, se presentan y analizan los resultados obtenidos del estudio de simulación.

#### 5.3.1. Algoritmo

El algoritmo antes mencionado se describe a continuación.

1. Fijar el tamaño de la población,  $N$ .
2. Generar los valores de la variable auxiliar,  $x_1, \dots, x_N$ .

### 5.3. Estudio de simulación

---

3. Elegir el tamaño de la muestra,  $n$ .
4. Fijar el grado del polinomio del modelo de splines,  $p$ .
5. Determinar el número de nodos por

$$J = \text{Min} \left\{ \frac{n}{4}, 35 \right\},$$

ver (Ruppert *et al.*, 2003).

6. Calcular la localización de los nodos por

$$\gamma_j = \frac{j+1}{J+2} \text{cuantil de los } x_i,$$

para  $j = 1, \dots, J$ , ver (Ruppert *et al.*, 2003).

7. Definir la matriz de diseño poblacional  $X$  de tamaño  $N \times (p+1+J)$  como:

$$X = \begin{pmatrix} 1 & x_1 & \cdots & x_1^p & (x_1 - \gamma_1)_+^p & \cdots & (x_1 - \gamma_J)_+^p \\ 1 & x_2 & \cdots & x_2^p & (x_2 - \gamma_1)_+^p & \cdots & (x_2 - \gamma_J)_+^p \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & \cdots & x_N^p & (x_N - \gamma_1)_+^p & \cdots & (x_N - \gamma_J)_+^p \end{pmatrix}.$$

8. Fijar los grados de libertad,  $\ell_U$ , y resolver la siguiente igualdad para  $\lambda_U$ :

$$\ell_U = \text{Traza} \left[ (X^T X + A_{\lambda_U})^{-1} X^T X \right].$$

La matriz diagonal  $A_{\lambda_U}$  resultante de tamaño  $(p+1+J) \times (p+1+J)$ , con  $\lambda_U > 0$ , es:

$$A_{\lambda_U} = \text{diag} \{0, \dots, 0, \lambda_U, \dots, \lambda_U\}.$$

Los primeros  $p+1$  elementos de  $A_{\lambda_U}$  son ceros y los restantes  $J$  elementos son iguales a  $\lambda_U$ .

9. Fijar el diseño de muestreo.
10. Fijar el número de simulaciones,  $M$ .
11. Repetir lo siguiente  $M$  veces:
  - 11.1 Seleccionar una muestra  $s$  de tamaño  $n$ .
  - 11.2 Determinar los grados de libertad,  $\ell_s$ , para el parámetro de suavizamiento  $\lambda_U$ :

$$\ell_s = \text{Traza} \left[ (X_s^T W_s X_s + A_{\lambda_U})^{-1} X_s^T W_s X_s \right],$$

### 5.3. Estudio de simulación

---

donde  $X_s$  y  $W_s$  se describen en la ecuación (5.8).

- 11.3 Determinar el parámetro de suavizamiento,  $\lambda_s$ , para la muestra  $s$  y grados de libertad  $\ell_U$  resolviendo la siguiente igualdad:

$$\ell_U = \text{Traza} \left[ (X_s^T W_s X_s + A_{\lambda_s})^{-1} X_s^T W_s X_s \right],$$

donde  $A_{\lambda_s}$  es similar a  $A_{\lambda_U}$ .

12. Calcular la raíz cuadrada del Error Cuadrático Medio Relativo (RMSE) para las estimaciones obtenidas en los pasos 11.2,  $\theta = \ell_U$ , y 11.3,  $\theta = \lambda_U$ , de la manera siguiente:

$$\sqrt{\frac{1}{M} \sum_{i=1}^M \left( \frac{\hat{\theta}_i - \theta}{\theta} \right)^2},$$

donde  $\hat{\theta}_i$  es el valor obtenido para  $\theta$  en la  $i$ -ésima simulación.

Observése que en el paso 11.3 se calcula el parámetro de suavizamiento resolviendo la ecuación (5.10). En el paso 11.2 se calculan los grados de libertad para la muestra correspondiente y el parámetro de suavizamiento obtenido en el paso 8. Esto último se realiza para estudiar la variación muestra a muestra de estos grados de libertad, de acuerdo a lo observado en (Breidt *et al.*, 2005).

En la siguiente parte de esta sección se describen los resultados obtenidos del ejercicio de simulación.

#### 5.3.2. Resultados empíricos

A continuación se describen los experimentos de simulación usando el algoritmo de la Sección 5.3.1. Luego, se muestran los resultados obtenidos de estos experimentos.

En los experimentos de simulación se utiliza un tamaño poblacional  $N = 3000$ . Se usan dos poblaciones, una generada con una distribución uniforme en  $[0, 1]$  y otra con una distribución exponencial con media 0.5 para la variable auxiliar. Se utilizan dos tamaños de muestra  $n_1 = 150$  y  $n_2 = 300$ . Para la selección de la muestra se usan dos diseños de muestreo: Muestreo Aleatorio Simple sin Reemplazo (SRS) y Muestreo de Máxima Entropía para probabilidades desiguales (MES). El diseño SRS se describió en la Sección 3.8.2 y el diseño MES se describe en (Matei y Tillé, 2005) y (Tillé, 2006). También se usan dos grados de libertad,  $\ell_{U,1} = 7$  y  $\ell_{U,2} = 11$ . Para el diseño MES, las probabilidades de inclusión de primer orden se definen como proporcionales al tamaño de cada unidad poblacional. Los valores de estos tamaños se generan de tal forma que la correlación entre



### 5.3. Estudio de simulación

la variable auxiliar y la variable tamaño es de 0.75. Con las dos poblaciones mencionadas, los dos tamaños de muestra, los dos diseños de muestreo y los dos grados de libertad se definen 16 posibles experimentos de simulación.

Adicionalmente, se fija el grado del polinomio del modelo de splines como  $p = 3$ . El número de nodos a considerar es  $J = \text{Min} \left\{ \frac{150}{4}, 35 \right\} = 35$ . La localización de los nodos se determinan como los cuantiles mencionados en el paso 6 del algoritmo de la Sección 5.3.1. Por último, el estudio de simulación se replica  $M = 5000$  veces. Este número de simulaciones se establece porque a partir de este número de repeticiones se estabiliza la estimación de  $\sqrt{ECMR(\lambda_s)}$  para la población exponencial, el diseño de máxima entropía y un tamaño de muestra igual a 150.

Ahora, los resultados de estos experimentos se presentan a continuación. En la Tabla 5.1, se muestran los valores de  $\sqrt{ECMR(\lambda_s)}$  para los 16 experimentos. En esta tabla se puede observar que la precisión de la estimación del parámetro  $\lambda_U$  decrece cuando se pasa de una población uniforme a una exponencial. Las diferencias en estas precisiones decrecen cuando se aumenta el tamaño de la muestra y los grados de libertad. El aumento en la precisión del estimador del parámetro es de esperar cuando se aumenta el tamaño de la muestra. Por otro lado, la disminución de la precisión de este estimador cuando se toman valores bajos de los grados de libertad se debe al comportamiento plano de la función en la expresión (5.10) en tales valores. Adicionalmente, para la población uniforme se puede observar que el diseño SRS produce estimaciones más precisas. Sin embargo, este comportamiento se invierte para la población exponencial. Explicar este comportamiento está más allá de los objetivos de este trabajo y se deja como una posterior investigación. Cabe mencionar que algunos valores de  $\sqrt{ECMR(\lambda_s)}$  son altos. Sin embargo, ya que en este trabajo indirectamente se estudia el efecto de la determinación del parámetro de suavizamiento en la estimación del total de la variable de interés, no se discutirá el rendimiento del estimador de  $\lambda_U$ . Esto se realiza en la siguiente sección para una población real.

**Tabla 5.1:**  $\sqrt{RMSE(\lambda_s)}$  para los 16 experimentos.

Diseño de muestreo y población	$n_1 = 150$		$n_2 = 300$	
	$\lambda_{U,1} = 0.00129$	$\lambda_{U,2} = 1.17 \times 10^{-5}$	$\lambda_{U,1} = 0.00129$	$\lambda_{U,2} = 1.17 \times 10^{-5}$
SRS y población uniforme	0.092085	0.115094	0.052364	0.057734
MES y población uniforme	0.144522	0.174384	0.085150	0.094315
Diseño de muestreo y población	$\lambda_{U,3} = 0.4525$ $\lambda_{U,4} = 0.00092$		$\lambda_{U,3} = 0.00129$ $\lambda_{U,4} = 0.00092$	
SRS y población exponencial	0.682349	0.369523	0.520342	0.177466
MES y población exponencial	0.343628	0.152860	0.171150	0.077195

Adicionalmente, en la Tabla 5.2 se muestran ahora los valores de  $\sqrt{RMSE(\ell_s)}$  para los experimentos. En esta tabla se puede observar que el comportamiento de la precisión del

## 5.4. Aplicación a datos reales

estimador de  $\ell_U$  es similar a lo descrito anteriormente. Sin embargo, los valores de la raíz cuadrada de RMSE son más pequeños. Notesé que hay más precisión en la estimación de  $\ell_U$ . Por otro lado, para estimar  $\lambda_U$  es necesario invertir Ecuación (5.10), lo cual adiciona un error numérico producido por el método que se utiliza para invertir la función.

**Tabla 5.2:**  $\sqrt{\widehat{RMSE}(\ell_s)}$  para los 16 experimentos.

Diseño de muestreo y población	$n_1 = 150$		$n_2 = 300$	
	$\ell_{U,1} = 7$	$\ell_{U,2} = 11$	$\ell_{U,1} = 7$	$\ell_{U,2} = 11$
SRS y población uniforme	0.008799	0.012658	0.004843	0.006132
MES y población uniforme	0.014467	0.020053	0.008028	0.010276
SRS y población exponencial	0.110197	0.051083	0.072501	0.019149
MES y población exponencial	0.040543	0.013238	0.016115	0.006183

En resumen, los resultados de los experimentos por simulación muestran que se tiene mayor precisión en la estimación de los grados de libertad que en la estimación del parámetro de suavizamiento. Esta precisión aumenta con el tamaño de la muestra para los dos parámetros mencionados. Sin embargo, se pierde precisión en la estimación de los dos parámetros cuando se pasa de una población con comportamiento uniforme a una con comportamiento exponencial.

En la siguiente sección se muestra el efecto que tiene el método de estimación del parámetro de suavizamiento, propuesto en este trabajo, sobre la estimación del total poblacional.

## 5.4. Aplicación a datos reales

Para estudiar la utilidad de la propuesta de la determinación del parámetro de suavizamiento se emplean datos reales tomados del Instituto Nacional de Estadística, Geografía e Informática de México (INEGI). La variable de interés es la población de los municipios de México en el año 2005 y la variable auxiliar es la población de los mismos municipios en el año 2000. Estos datos se pueden obtener de la página web de la institución citada. Para estos datos, se utilizan los mismos diseños de muestreo de la sección anterior. Además, en esta sección es de interés evaluar la precisión del estimador de la población total de México en el año 2005. Para el caso del MES, se definen las probabilidades de inclusión de primer orden como proporcionales al ingreso mensual promedio de los municipios de México en el año 2000. Buscar otra variable que aumente esta precisión está más allá de los objetivos de este trabajo.

El número de unidades de la población es de 2442, el número de municipios existentes en el año 2000 en México. Se utiliza un tamaño de muestra  $n = 150$ . Con este tamaño de muestra se obtiene un error relativo de estimación de 5% aproximadamente para el total

## 5.4. Aplicación a datos reales

poblacional utilizando el diseño SRS. Además se utilizan los mismos grados de libertad que en la sección anterior,  $\ell_{U,1} = 7$  y  $\ell_{U,2} = 11$ . De acuerdo a lo mencionado, se definen 4 posibles experimentos de simulación.

Adicionalmente, se fija el grado del polinomio del modelo de splines como  $p = 1$ . Para grados mayores del polinomio, la inversa de la matriz  $(X^T X + A_{\lambda_U})$  de la expresión (5.5) no se puede calcular su inversa. Además, se utilizan 35 nodos. La localización de los nodos se determina de acuerdo al paso 6 del algoritmo en la Sección 5.3.1. Por último, el estudio de simulación se replica 5000 veces. Los resultados de estos experimentos se presentan a continuación.

En la Tabla 5.3 se muestran los valores de  $\sqrt{RMSE}$  de los estimadores  $\hat{\ell}_s$ ,  $\hat{\lambda}_s$ ,  $\hat{t}_{y,\hat{\lambda}_s}$  y  $\hat{t}_{y,\lambda_U}$ . El estimador  $\hat{t}_{y,\lambda_U}$  es dado en la ecuación (5.9) con  $\lambda = \lambda_U$  y el estimador  $\hat{t}_{y,\hat{\lambda}_s}$  utiliza  $\lambda = \hat{\lambda}_s$ . Nuevamente se observa que los valores de  $\sqrt{RMSE}$  son mayores para las estimaciones de  $\lambda_U$  que para las de  $\ell_U$ . Además, estos valores decrecen cuando se utiliza el diseño MES. Por otro lado, el aumento en los valores del  $\sqrt{RMSE}$  del estimador (5.9) utilizando  $\hat{\lambda}_s$  no es significativo con respecto a los valores que se obtienen utilizando  $\lambda_U$ . Por lo tanto, estos experimentos de simulación muestran, para la población aquí utilizada, que los valores altos del  $\sqrt{RMSE}$  del estimador  $\hat{\lambda}_s$  no afectan significativamente en la precisión del estimador del total poblacional.

**Tabla 5.3:**  $\sqrt{RMSE}$  de los estimadores.

Estimadores	SRS( $n = 150$ )		MES( $n = 150$ )	
	$\ell_{U,1} = 7$	$\ell_{U,2} = 11$	$\ell_{U,1} = 7$	$\ell_{U,2} = 11$
	( $\lambda_{U,1} = 21, 866, 888, 904$ )	( $\lambda_{U,2} = 1, 415, 549, 050$ )	( $\lambda_{U,1} = 21, 866, 888, 904$ )	( $\lambda_{U,2} = 1, 415, 549, 050$ )
$\hat{\ell}_s$	0.036098	0.024691	0.017615	0.021898
$\hat{\lambda}_s$	0.163896	0.126032	0.100820	0.124445
$\hat{t}_{y,\hat{\lambda}_s}$	0.031584	0.052294	0.014194	0.014483
$\hat{t}_{y,\lambda_U}$	0.029331	0.049478	0.014182	0.014476

En la Tabla 5.4, se muestran los errores relativos de estimación del total poblacional. Estos errores se calculan como sigue. Primero, para cada muestra se calcula la siguiente cantidad:

$$\left| \frac{\hat{t}_{y,\lambda} - t_y}{t_y} \right|.$$

uno para  $\hat{t}_{y,\hat{\lambda}_s}$  y otra para  $\hat{t}_{y,\lambda_U}$ . Después, de los 5000 valores así obtenidos en el experimento de simulación, se calcula su respectivo percentil del 95 % como el error relativo de estimación,  $\hat{\epsilon}_\lambda$ . De esta forma las cantidades que se reportan en la Tabla 5.4 se interpretan como el máximo error relativo de estimación con al menos 95 % de probabilidad. Esto es, dicho error satisface aproximadamente la siguiente relación:

$$P\left(\left| \frac{\hat{t}_{y,\lambda} - t_y}{t_y} \right| < \hat{\epsilon}_\lambda\right) \geq 0.95.$$

## 5.5. Conclusiones

Observése primero que los errores relativos de estimación son menores para el diseño MES. Además, estos errores son similares si se utiliza el parámetro de suavizamiento  $\lambda_U$  o el estimador propuesto,  $\hat{\lambda}_s$ . Por lo tanto estos experimentos de simulación muestran también que el error relativo de estimación del total poblacional no se afecta significativamente cuando se utiliza el parámetro de suavizamiento aquí propuesto.

**Tabla 5.4:** Errores Relativos de Estimación(ERE).

ERE	SRS( $n = 150$ )		MES( $n = 150$ )	
	$\ell_{U,1} = 7$	$\ell_{U,2} = 11$	$\ell_{U,1} = 7$	$\ell_{U,2} = 11$
	$(\lambda_{U,1} = 21, 866, 888, 904)$	$(\lambda_{U,2} = 1, 415, 549, 050)$	$(\lambda_{U,1} = 21, 866, 888, 904)$	$(\lambda_{U,2} = 1, 415, 549, 050)$
$\hat{\epsilon}_{\hat{\lambda}_s}$	0.054792	0.059532	0.027925	0.028544
$\hat{\epsilon}_{\lambda_U}$	0.053230	0.058350	0.027964	0.028593

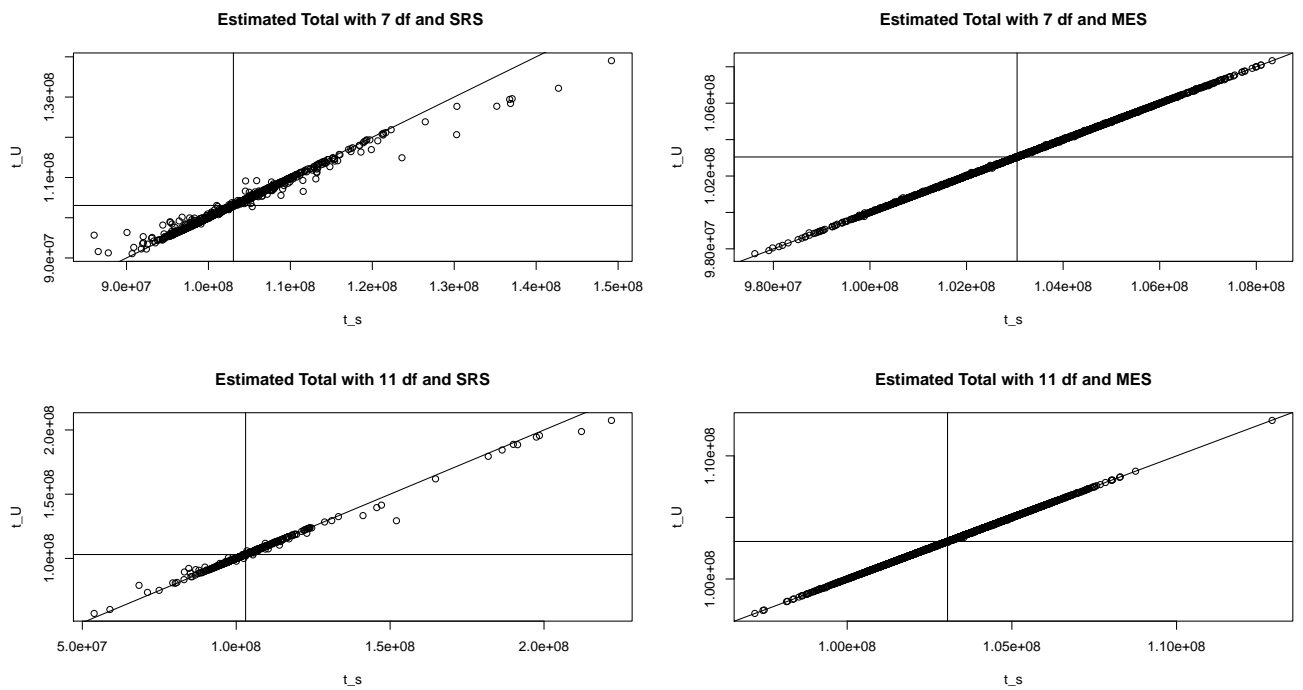
Ahora, en la Figura 5.1 se presentan los diagramas de dispersión de las estimaciones del total usando  $\lambda_U$  y  $\hat{\lambda}_s$ . Observése que la mayoría de los puntos están alrededor de la recta de referencia de  $45^\circ$ . Ya se ha observado que el error relativo de estimación del total poblacional es similar si utilizamos  $\lambda_U$  o  $\hat{\lambda}_s$ . Adicionalmente, el comportamiento en los anteriores diagramas de dispersión muestran que la mayoría de las estimaciones utilizando  $\lambda_U$  o  $\hat{\lambda}_s$  son similares.

## 5.5. Conclusiones

Se propuso un método para estimar el parámetro de suavizamiento mediante el uso de información de una muestra para el modelo de regresión por splines penalizado, la forma en que el parámetro de suavizamiento se calcula no afecta a la estimación del total poblacional, a pesar de que se calcula el parámetro de suavizamiento como se hace comúnmente. El error relativo de la estimación del total poblacional no se ve afectada significativamente cuando se usa el parámetro de suavizamiento que aquí se propone. Esta propuesta es una alternativa para los casos de inestabilidad numérica en la estimación de parámetros de suavizamiento, como se describe en la Sección 5.2.

El estudio de simulación realizado con datos reales proporcionó evidencia empírica de que la estimación propuesta del parámetro de suavizamiento produce errores relativos de estimación del total poblacional similares cuando se utiliza  $\lambda_U$  o  $\hat{\lambda}_s$ .

## 5.5. Conclusiones



**Figura 5.1:** Diagramas de dispersión de las estimaciones del total. Las estimaciones del total usando  $\hat{\lambda}_s$  están sobre el eje horizontal y las estimaciones usando  $\lambda_U$  están sobre el eje vertical. Las líneas horizontales y verticales indican el valor verdadero del total poblacional.

# Capítulo 6

## Modelo de coeficientes variantes en muestreo de poblaciones finitas

En este capítulo se presentan y desarrollan los modelos de coeficientes variantes en el contexto de muestreo de poblaciones finitas. Uno de los aportes de este trabajo es una forma de seleccionar los parámetros de suavizamiento y se estudian las propiedades estadísticas del estimador del total poblacional. Además se presenta un estudio de simulación para estudiar de manera empírica el estimador del total poblacional. Finalmente, se aplica la metodología propuesta a un conjunto de datos reales.

Los modelos de coeficientes variantes ([Hastie y Tibshirani, 1993](#)) y muchas de sus variaciones (por ejemplo, ([Hoover \*et al.\*, 1998](#))) han ganado mucha atención en la literatura. Las aplicaciones se encuentran en varias áreas científicas, tales como economía, negocios, ciencias médicas, ecología, econometría y epidemiología. Los modelos de coeficientes variantes amplían los modelos lineales clásicos al permitir que los coeficientes de regresión sean funciones semiparamétricas de una covariable, o variable auxiliar, como se llama en muestreo de poblaciones finitas. Estos modelos son una herramienta estadística útil para explorar los patrones dinámicos en una relación de regresión, en el que se toman las características de variación de los coeficientes de regresión como prueba principal para reflejar la relación dinámica entre las variables respuesta y explicativa. Para una revisión detallada de algunas aplicaciones de los modelos de coeficientes variantes, ver ([Fan y Zhang, 2008](#)).

En ([Fan y Zhang, 1999](#)), ([Cai \*et al.\*, 2000a](#)), ([Lu \*et al.\*, 2008](#)) y ([Zhang y Mei, 2012](#)) se presentan métodos de estimación e inferencia para los modelos de coeficientes variantes. En este trabajo solamente se está interesado en adaptar el modelo de coeficientes variantes a muestreo de poblaciones finitas y proponer un estimador del total poblacional.

Comúnmente en estudios de muestreo de poblaciones finitas la información auxiliar multivariada está disponible a nivel poblacional. Por ejemplo, en muchos países, los registros

## 6.1. Modelo de coeficientes variantes en muestreo de poblaciones finitas

---

administrativos proveen fuentes extensas de información auxiliar. Los registros completos pueden dar acceso a variables auxiliares tales como edad, sexo, ingreso y país de nacimiento. Otro ejemplo son los datos de imágenes satelitales o GPS que son usados en muestreo espacial. Generalmente, estos datos son coleccionados a nivel poblacional, los cuales son a menudo disponibles a un costo bajo, especialmente comparado con el costo de tener los datos de muestreo. Si no hay otra información más que las probabilidades de inclusión de primer orden, entonces el total poblacional puede ser estimado por el estimador insesgado de Horvitz-Thompson (Horvitz y Thompson, 1952). (Breidt y Opsomer, 2000) y (Breidt *et al.*, 2005) afirman que el uso de la información auxiliar y el uso del modelo de superpoblación pueden mejorar la eficiencia de la estimación del total poblacional.

En un contexto general los métodos de regresión semiparamétricos multivariados, tales como los modelos aditivos, modelos single-index y modelos de coeficientes variantes han sido utilizados ampliamente. En el contexto de muestreo, (Wang, 2009) usó los modelos de regresión single-index y (Wang y Wang, 2011) desarrolló y adaptó el modelo aditivo. En este trabajo, se describe el modelo de coeficientes variantes en muestreo de poblaciones finitas, se propone una forma de seleccionar los parámetros de suavizamiento y se estudian algunas propiedades estadísticas del estimador del total poblacional.

## 6.1. Modelo de coeficientes variantes en muestreo de poblaciones finitas

Sea  $U_N = \{1, \dots, N\}$  la población finita de interés compuesta de  $N$  elementos. Sea  $\mathbf{x}_k = \{u_k, x_{k1}, \dots, x_{kr}\}$  el vector de variables auxiliares  $(r + 1)$ -dimensional, conocidas para todo  $k \in U_N$ . Se está interesado en estimar el total poblacional  $t_y = \sum_{k \in U_N} y_k$ , donde  $y_k$  es el valor de la variable de interés para el elemento  $k$ -ésimo. Para estudiar la relación entre las variables  $y_k$  y el vector de variables auxiliares  $\mathbf{x}_k$  se considera el modelo de regresión de superpoblación (Breidt *et al.*, 2005):

$$y_k = m(\mathbf{x}_k) + \eta_k, \quad k \in U_N, \quad (6.1)$$

donde los valores  $\eta_k$  son variables aleatorias independientes con media cero y varianza  $\nu(\mathbf{x}_k)$ . Sea  $\{(\mathbf{x}_k, y_k)\}_{k \in U_N}$  una realización de este modelo de superpoblación.

En el contexto de regresión semiparamétrica, la función  $(r + 1)$ -variada  $m$  se desconoce y a menudo se asume que sea suave. Entonces la función  $m$  puede ser aproximada de acuerdo a un modelo de coeficientes variantes (Hastie y Tibshirani, 1993):

$$m(\mathbf{x}_k) \approx a_0(u_k) + \sum_{j=1}^r a_j(u_k)x_{kj}. \quad (6.2)$$

## 6.1. Modelo de coeficientes variantes en muestreo de poblaciones finitas

De igual manera, las funciones  $a_j$  son desconocidas y pueden ser aproximadas por un modelo de splines de orden  $p$  y nodos  $\gamma_i$ , (Ruppert *et al.*, 2003), ver Capítulo 4:

$$a_j(u_k) \approx \sum_{h=0}^p \beta_{j,h} u_k^h + \sum_{i=1}^K b_{j,i} (u_k - \gamma_i)_+^p, \quad (6.3)$$

para  $j = 0, \dots, r$  y  $k \in U_N$ . Se usa el mismo número de nodos  $K$  para cada función  $a_j$  ya que las funciones  $a_j$  están en función de la misma variable auxiliar  $u_k$ . (Hastie y Tibshirani, 1993) mencionan que la variable  $u_k$  puede ser multivariada, pero en este trabajo se asume univariada y puede ser una variable del vector  $\{x_{k1}, \dots, x_{kr}\}$ . (Cao *et al.*, 2010) afirman que si no hay información a priori de la variable  $u_k$ , esta variable puede ser elegida como se hace en la selección de variables en los modelos de regresión. También se puede seleccionar  $u_k$  por *stepwise addition* seguido por *stepwise deletion* usando, por ejemplo, el criterio de validación cruzada generalizada o el criterio de información de Akaike.

Entonces el modelo (6.1) puede ser escrito como un modelo lineal mixto:

$$Y = X\beta + Zb + \varepsilon, \quad (6.4)$$

donde

$$\begin{aligned} X &= [1, u_k, \dots, u_k^p, I(x_{k1}, u_k), \dots, I(x_{kr}, u_k)]_{k \in U_N}, \\ Z &= [(u_k - \gamma_1)_+^p, \dots, (u_k - \gamma_K)_+^p, T(x_{k1}, u_k), \dots, T(x_{kr}, u_k)]_{k \in U_N}, \\ I(x_{kh}, u_k) &= [x_{kh}, u_k x_{kh}, \dots, u_k^p x_{kh}], \\ T(x_{kh}, u_k) &= [x_{kh} (u_k - \gamma_1)_+^p, \dots, x_{kh} (u_k - \gamma_K)_+^p]. \end{aligned}$$

Notése que la matriz  $X$  contiene los términos del polinomio de spline de orden  $p$  con sus respectivas interacciones y la matriz  $Z$  contiene los términos del polinomio truncado del modelo de splines con interacciones. Uno de los aportes del modelo de coeficientes variantes es que es capaz de captar relaciones no lineales entre las variables.

Si definimos  $C = [X, Z]$  y  $B = [\beta, b]^T$ , entonces el modelo (6.4) puede ser escrito como:

$$Y = CB + \varepsilon.$$

Una forma de encontrar el estimador de  $B$ , es por medio de mínimos cuadrados penalizados, es decir, el vector  $B$  que minimiza:

$$(Y - CB)^T (Y - CB) + B^T D_\lambda B,$$

donde  $D_\lambda = \text{diag} \{ \mathbf{0}_{1 \times [(p+1) \times (r+1)]}, \lambda_1 \mathbf{1}_{1 \times K}, \dots, \lambda_{r+1} \mathbf{1}_{1 \times K} \}$ . Entonces, el estimador de



## 6.1. Modelo de coeficientes variantes en muestreo de poblaciones finitas

mínimos cuadrados penalizado del coeficiente de parámetros  $B$  es:

$$B_{VC} = (C^T C + D_\lambda)^{-1} C^T Y. \quad (6.5)$$

Notése que la matriz diagonal  $D_\lambda$  depende de los parámetros de suavizamiento  $\lambda$ , en donde una forma de obtener  $\lambda$  se discute en la sección siguiente.

Si la población  $U_N$  es observada, entonces

$$m_{VC}(\mathbf{x}_k) = c_k^T B_{VC} = c_k^T (C^T C + D_\lambda)^{-1} C^T Y, \quad k \in U_N, \quad (6.6)$$

donde  $c_k^T = [x_k^T, z_k^T]$  y  $Y = (y_1, \dots, y_N)^T$ . Sea  $m_k \equiv c_k^T B_{VC}$  para  $k \in U$  el cual denota el estimador a nivel poblacional. Si estos valores son conocidos entonces se pueden utilizar para estimar el total poblacional  $t_y$  como lo menciona (Sarndal *et al.*, 1992):

$$\hat{t}_{y,diff} = \sum_{k \in U_N} m_k + \sum_{k \in s} \frac{y_k - m_k}{\pi_k}. \quad (6.7)$$

El estimador (6.7) es insesgado y su varianza diseño es:

$$Var(\hat{t}_{y,diff}) = \sum_{k,l \in U_N} \Delta_{kl} \frac{y_k - m_k}{\pi_k} \frac{y_l - m_l}{\pi_l}, \quad (6.8)$$

donde  $\Delta_{kl} = \pi_{kl} - \pi_k \pi_l$ .

De igual forma, como se mencionó en el Capítulo 4, en muestreo de poblaciones finitas, los valores (6.6) solamente son conocidos para una muestra  $s$  de  $U_N$ .

En el contexto de muestreo de poblaciones finitas, sea  $s$  una muestra de  $U_N$  bajo algún diseño de muestreo  $p(s)$ , entonces un estimador de  $m_{VC}(\mathbf{x}_k)$  es:

$$\hat{m}_{VC}(\mathbf{x}_k) = c_k^T \hat{B}_{VC} = c_k^T (C_s^T W_s C_s + D_\lambda)^{-1} C_s^T W_s Y_s = c_k^T G_\lambda Y_s, \quad (6.9)$$

donde  $W_s = \text{diag} \left\{ \frac{1}{\pi_j} \right\}_{j \in s}$  y  $\pi_j = P(j \in s)$ .

El estimador del total poblacional  $t_y$  es (Sarndal *et al.*, 1992):

$$\hat{t}_{y,VC} = \sum_{k \in U_N} \hat{m}_{VC}(\mathbf{x}_k) + \sum_{k \in s} \frac{y_k - \hat{m}_{VC}(\mathbf{x}_k)}{\pi_k}. \quad (6.10)$$

Nótese que el estimador (6.10) es similar al obtenido en (Breidt *et al.*, 2005). Un estimador

## 6.2. Selección de los parámetros de suavizamiento

---

para la varianza  $Var(\hat{t}_{y,VC})$  es:

$$\widehat{Var}(\hat{t}_{y,VC}) = \sum_{i,j \in s} \frac{\Delta_{ij}}{\pi_{ij}} \frac{y_i - \hat{m}_{VC}(\mathbf{x}_i)}{\pi_i} \frac{y_j - \hat{m}_{VC}(\mathbf{x}_j)}{\pi_j}. \quad (6.11)$$

Note que las Ecuaciones (6.10) y (6.11) dependen de los parámetros de suavizamiento  $\lambda$ . En un contexto general, el método comúnmente usado para determinar estos parámetros es validación cruzada. En este trabajo se propone una forma de obtenerlos en un contexto de muestreo de poblaciones finitas, como se describe en la siguiente sección.

## 6.2. Selección de los parámetros de suavizamiento

La selección del parámetro de suavizamiento  $\lambda$  es crucial para obtener un buen ajuste a la curva. En este capítulo tenemos  $r + 1$  parámetros de suavizamiento, es decir, se tienen tantos parámetros de suavizamiento como coeficientes variantes tengamos, lo cual se hace más complicada la selección de  $\lambda$ .

En la literatura de splines, es muy popular el criterio  $GCV$  (Generalized Cross-Validation score). El criterio  $GCV$  selecciona como parámetro de suavizamiento aquél valor  $\lambda$  que minimiza

$$GCV(\lambda) = \frac{N^{-1} [Y - CB_{VC}(\lambda)]^T [Y - CB_{VC}(\lambda)]}{[1 - N^{-1}df(\lambda)]^2}, \quad (6.12)$$

donde  $df(\lambda)$  son los grados de libertad del modelo y se definen como:

$$df(\lambda) = \text{Traza} \left[ (C^T C + D_\lambda)^{-1} C^T C \right]. \quad (6.13)$$

Observe que se trata de un problema de optimización numérica en el espacio  $r + 1$  dimensional. Una solución aproximada a este tipo de problemas se encuentra en (Ruppert y Carroll, 2000).

El método  $EBBS$  (Empirical Bias Bandwidth) desarrollado por (Ruppert, 1997) para elegir el ancho de banda para regresión local puede ser extendido a otros parámetros de suavizamiento. (Cao *et al.*, 2010) modificó el método  $EBBS$  para usarlo en los modelos de splines usando el modelo de coeficientes variantes en el contexto de series de tiempo.

De lo expuesto anteriormente, la selección de los parámetros de suavizamiento es de suma importancia debido a que las estimaciones de los parámetros se ven afectada por  $\lambda$ .

En el contexto de muestreo de poblaciones finitas el problema de determinar el parámetro de suavizamiento sigue abierto. Una primera aproximación en la selección de estos parámetros se encuentra en el trabajo de (Opsomer y Miller, 2005). En el caso univariado, (Op-

## 6.2. Selección de los parámetros de suavizamiento

(somer y Miller, 2005) proponen un criterio de selección óptima del ancho de banda  $h$  dado por el error cuadrático medio ( $MSE$ ), basado en

$$\widehat{V}_{cv}(h) = \sum_{i,j \in s} \frac{\Delta_{ij}}{\pi_{ij}} \frac{y_i - \widehat{m}_i^{(-)}}{\pi_i} \frac{y_j - \widehat{m}_j^{(-)}}{\pi_j}, \quad (6.14)$$

donde  $\widehat{V}_{cv}(h)$  es el llamado criterio de selección del ancho de banda por validación cruzada basado en el diseño.

(Opsomer y Miller, 2005) mencionan que su idea puede extenderse a cualquier parámetro de suavizamiento, inclusive multivariado.

En este trabajo se propone una forma de seleccionar los parámetros de suavizamiento  $\lambda = (\lambda_1, \dots, \lambda_{r+1})$  como se describe a continuación.

Los grados de libertad del modelo se definen como en (6.13). (Ruppert *et al.*, 2003) afirma que es posible calcular los grados de libertad para cada componente en el modelo aditivo, donde el modelo aditivo se define como:

$$y_k = \beta_0 + \sum_{j=1}^r m_j(x_{kj}) + \epsilon_k. \quad (6.15)$$

A continuación se calculan los grados de libertad en el modelo de coeficientes variantes. Sea  $P$  el número de columnas de  $C = [X, Z]$ , y sea  $\{I_0, I_1, \dots, I_{r+1}\}$  una partición de los índices columnas  $\{1, 2, \dots, P\}$  tal que  $I_0$  corresponde al intercepto e  $I_j$  a cada función de las variables auxiliares para  $j = 1, \dots, r + 1$ .

Una aproximación simple a los grados de libertad individuales (para cada variable auxiliar) es :

$$df_j(\lambda_j) = \text{Traza} \left[ (C_j^T C_j + D_{\lambda_j})^{-1} C_j^T C_j \right] - 1, \quad (6.16)$$

donde  $C_j \equiv C_{I_0 \cup I_j}$ ,  $D_{\lambda_j} = \text{diag} \{ \mathbf{0}, \lambda_j \mathbf{1}_{1 \times K} \}$  para  $j = 1, \dots, r + 1$ . Para más detalle de esta aproximación, ver (Aerts *et al.*, 2002) y (Ruppert *et al.*, 2003).

La aproximación (6.16) es conveniente debido a su interpretación, es decir, dado un valor  $df_j(\lambda_j)$  se puede encontrar  $\lambda_j$  resolviendo la ecuación (6.16). La propuesta de este trabajo es construir una malla de dimensión  $r + 1$  asociados a los valores  $\lambda_j$  que corresponden a los grados de libertad  $df_j(\lambda_j)$ . Entonces, a partir de una muestra observada, el estimador de la varianza del tipo (6.14) es evaluada en cada nodo de la malla y el  $\widehat{\lambda}_{opt}$  es aquel valor que minimize el estimador de la varianza.

A continuación se presenta el algoritmo para obtener  $\widehat{\lambda}_{opt}$ .

1. Se construye una malla de dimensión  $(r + 1)$ , la cual se denota por  $\Delta$ . La malla se

## 6.2. Selección de los parámetros de suavizamiento

obtiene observando que  $n_j \leq df_j(\lambda_j) \leq m_j$  para  $j = 1, \dots, r + 1$ . Entonces la malla se define como el siguiente producto cartesiano:

$$\Delta = \{\lambda_{1,l_1}\}_{l_1=1}^{m_1-n_1+1} \times \{\lambda_{2,l_2}\}_{l_2=1}^{m_2-n_2+1} \times \dots \times \{\lambda_{r+1,l_{r+1}}\}_{l_{r+1}=1}^{m_{r+1}-n_{r+1}+1}, \quad (6.17)$$

donde  $\lambda_{j,l_j}$ ,  $j = 1, \dots, r + 1$  se obtiene resolviendo la ecuación (6.16) para un valor dado,  $df_j(\lambda_{j,l_j})$ . Se tiene una malla regular cuando  $m_1 = \dots = m_{r+1} = m$  y  $n_1 = \dots = n_{r+1} = n$ .

2. Se toma una muestra  $s$  de tamaño  $n$  de la población  $U_N$  bajo algún diseño de muestreo.

3. Se minimiza

$$\widehat{Var}_{cv}(\widehat{t}_{y,VC}) = \sum_{i,j \in s} \frac{\Delta_{ij}}{\pi_{ij}} \frac{y_i - \widehat{m}_{VC}^{(-)}(\mathbf{x}_i)}{\pi_i} \frac{y_j - \widehat{m}_{VC}^{(-)}(\mathbf{x}_j)}{\pi_j}, \quad (6.18)$$

sobre  $\Delta$  para obtener  $\widehat{\lambda}_{opt}$ , donde

$$\widehat{m}_{VC}^{(-)}(\mathbf{x}_i) = \mathbf{c}_i^T \left( \sum_{j \in s, j \neq i} \frac{\mathbf{c}_j \mathbf{c}_j^T}{\pi_j} + D_\lambda \right)^{-1} \sum_{j \in s, j \neq i} \frac{\mathbf{c}_j y_j}{\pi_j}.$$

Entonces  $\widehat{\lambda}_{opt} = \arg \min_{\lambda \in \Delta} \widehat{Var}_{cv}(\widehat{t}_{y,VC})$ .

Por lo tanto, la estimación de  $t_y$  se calcula usando la Ecuación (6.10) y  $\widehat{\lambda}_{opt}$  de la siguiente forma:

$$\widehat{t}_{y,VC}(\widehat{\lambda}_{opt}) = \sum_{k \in U_N} \widehat{m}_{VC}(\mathbf{x}_k; \widehat{\lambda}_{opt}) + \sum_{k \in s} \frac{y_k - \widehat{m}_{VC}(\mathbf{x}_k; \widehat{\lambda}_{opt})}{\pi_k}. \quad (6.19)$$

El criterio para seleccionar el valor óptimo de  $\lambda$ , en el contexto de muestreo, mide las desviaciones entre un estimador basado en la muestra y una cantidad poblacional finita, no desviaciones entre un estimador y un modelo. Otra diferencia entre este criterio y el tradicional CV (Cross-Validation), es que el propuesto depende del estimador semi-paramétrico (también de  $\lambda$ ) en una forma complicada y no convencional, porque ésta necesita incorporar el diseño de muestreo. (Opsomer y Miller, 2005) mencionan que los diseños *ST*, *PPT*, y *C*, descritos en el Capítulo 3, tienen un impacto sobre el procedimiento de estimación y necesitan ser tomados en cuenta para la selección de los parámetros de suavizamiento.

De acuerdo a la naturaleza de los datos de muestreo que incorporan el diseño de muestreo, los métodos existentes de selección de parámetros de suavizamiento, mencionados anteriormente y en el Capítulo 5, no pueden ser aplicados en este caso y necesitan ser reemplazados por métodos específicamente para estimadores asistidos por modelos. La principal ventaja de CV como una corrección para sobreajuste en la selección de modelos son conocidas ampliamente en situaciones de modelación paramétrica y semiparamétrica.

## 6.3. Propiedades del estimador del total poblacional

Para poder estudiar las propiedades asintóticas del estimador del total poblacional, se usa el marco asintótico descrito en la Sección 3.6 en el cual la población  $U_N$  y el diseño de muestreo  $p(\cdot)$  son incrustados en una sucesión de poblaciones y diseños indexados por  $N$ ,  $\{U_N, p_N(\cdot)\}$ , con  $N \rightarrow \infty$ . Por simplicidad, supongamos que el tamaño de muestra  $n_N$  es fijo para cada  $N$  y también se asume que  $n_N \rightarrow \infty$ . De igual manera,  $p$ ,  $K$ , los  $\{\gamma_i\}$  son fijos, y los parámetros de suavizamiento  $\lambda$  son fijos. Para demostrar los resultados, se hacen los siguientes supuestos técnicos.

*Supuesto 1.*  $B = \lim_{N \rightarrow \infty} B_{VC}$  existe, y  $\widehat{B}_{VC} - B_{VC} = o_p(1)$ .

*Supuesto 2.* La matriz de covarianza diseño límite de los estimadores de H-T normalizados,

$$\Sigma = \begin{pmatrix} \Sigma_{yy} & \Sigma_{yc} \\ \Sigma_{cy}^T & \Sigma_{cc} \end{pmatrix} = \lim_{N \rightarrow \infty} n_N \begin{bmatrix} \sum_{i,j \in U_N} \frac{\Delta_{ij}}{N^2} \frac{y_i y_j}{\pi_i \pi_j} & \sum_{i,j \in U_N} \frac{\Delta_{ij}}{N^2} \frac{y_i c_j^T}{\pi_i \pi_j} \\ \sum_{i,j \in U_N} \frac{\Delta_{ij}}{N^2} \frac{c_i y_j}{\pi_i \pi_j} & \sum_{i,j \in U_N} \frac{\Delta_{ij}}{N^2} \frac{c_i c_j^T}{\pi_i \pi_j} \end{bmatrix},$$

es definida positiva.

*Supuesto 3.* Los estimadores de H-T normalizados satisfacen el teorema central del límite :

$$\frac{\sqrt{n_N}}{N} \begin{bmatrix} \sum_{i \in U_N} y_i \left( \frac{I_i}{\pi_i} - 1 \right) \\ \sum_{i \in U_N} c_i^T \left( \frac{I_i}{\pi_i} - 1 \right) \end{bmatrix} \xrightarrow{d} N(\mathbf{0}, \Sigma).$$

*Supuesto 4.* La matriz de covarianza estimada para los estimadores de H-T es consistente en diseño en el siguiente sentido:

$$n_N \begin{bmatrix} \sum_{i,j \in s} \frac{\Delta_{ij}}{N^2 \pi_{ij}} \frac{y_i y_j}{\pi_i \pi_j} & \sum_{i,j \in s} \frac{\Delta_{ij}}{N^2 \pi_{ij}} \frac{y_i c_j^T}{\pi_i \pi_j} \\ \sum_{i,j \in s} \frac{\Delta_{ij}}{N^2 \pi_{ij}} \frac{c_i y_j}{\pi_i \pi_j} & \sum_{i,j \in s} \frac{\Delta_{ij}}{N^2 \pi_{ij}} \frac{c_i c_j^T}{\pi_i \pi_j} \end{bmatrix} - \Sigma = o_p(1),$$

cuando  $N \rightarrow \infty$ .

El *supuesto 1* asegura que el estimador muestral  $\widehat{B}_{VC}$  y el estimador a nivel poblacional  $B_{VC}$  comparten un límite común. Los *supuestos 2* y *3* comúnmente se satisfacen en diseños de muestreo con tamaño de muestra fijo. Estos supuestos también son ciertos para diseños de muestreo con tamaño de muestra aleatoria, pero en el caso en que  $n_N$  es reemplazado por  $E_p(n_N)$ . El *supuesto 4* se satisface para algunos diseños de muestreo.

Los siguientes resultados establecen consistencia en diseño y normalidad asintótica de  $\widehat{t}_{y,VC}$ , junto con un estimador de la varianza consistente. Las demostraciones se presentan en los **Anexos D, E y F**.

## 6.4. Estudio de simulación

---

**Teorema 1** Bajo los supuestos 1 – 3, el estimador  $\hat{t}_{y,VC}$  es  $\sqrt{n_N}$ -consistente en diseño para  $t_y$  en el sentido que  $N^{-1} (\hat{t}_{y,VC} - t_y) = O_p(n_N^{-\frac{1}{2}})$ . Además,

$$\frac{\hat{t}_{y,VC} - t_y}{\sqrt{Var(\hat{t}_{y,diff})}} \xrightarrow{d} N(0, 1).$$

**Teorema 2** Bajo los supuestos 1, 2 y 4,

$$\widehat{Var}(\hat{t}_{y,VC}) = \sum_{i,j \in s} \frac{\Delta_{ij} y_i - \hat{m}_i y_j - \hat{m}_j}{\pi_{ij} \pi_i \pi_j} = Var(\hat{t}_{y,diff}) + o_p\left(\frac{N^2}{n_N}\right).$$

**Corolario 1** Bajo los supuestos 1 – 4, el estimador asistido por el modelo de coeficientes variantes,  $\hat{t}_{y,VC}$ , satisface

$$\frac{\hat{t}_{y,VC} - t_y}{\sqrt{\widehat{Var}(\hat{t}_{y,VC})}} \xrightarrow{d} N(0, 1).$$

Estos resultados teóricos muestran que es posible utilizar una especificación de modelo flexible como en (6.2) para el modelo de superpoblación, mientras se mantengan las mismas propiedades básicas de diseño.

## 6.4. Estudio de simulación

En esta sección, se hace un estudio de simulación para estudiar de manera empírica el desempeño de  $\hat{t}_{y,VC}$ . Para la comparación, se usa el modelo aditivo y el modelo de regresión clásico.

Se genera una población aleatoria de  $N = 1000$  valores de  $x_{i1}$  y  $x_{i2}$  de la distribución uniforme sobre  $[0, 1]$ , se toman 1000 valores para los errores  $\epsilon$  de una distribución  $N(0, 1)$ . Se usa este error en todas las simulaciones y es multiplicado por  $\sigma$ . Se generan cuatro poblaciones como sigue:

$$y_{ih} = m_h(x_{i1}, x_{i2}) + \epsilon_i,$$

para  $i = 1, \dots, 1000$  y  $h = 1, 2, 3, 4$ .

Las funciones  $m_h(\cdot)$  se definen como:

$$m_1(x_{i1}, x_{i2}) = 5 + \sin[2\pi x_{i1}] + \cos[2\pi x_{i2}], \quad (6.20)$$

$$m_2(x_{i1}, x_{i2}) = 5 + \cos[2\pi |x_{i1} - x_{i2}|] - \sin[2\pi |x_{i1} + x_{i2}|], \quad (6.21)$$

## 6.4. Estudio de simulación

$$m_3(x_{i1}, x_{i2}) = 5 + x_{i2} \sin [2\pi x_{i1}] + \cos [2\pi x_{i1}], \quad (6.22)$$

$$m_4(x_{i1}, x_{i2}) = 5 + x_{i1} + x_{i2}. \quad (6.23)$$

Notése que la ecuación (6.20) representa a una población aditiva lineal, el modelo (6.21) representa a una población no lineal, la población (6.22) representa un modelo de coeficientes variantes (*VC*) y el modelo (6.23) representa a una población lineal.

Las cantidades poblacionales de interés son los totales  $t_{yh} = \sum_{i=1}^{1000} y_{ih}$ , para  $h = 1, 2, 3, 4$ .

Las muestras son tomadas usando el diseño de muestreo MSR. Son tomadas  $M = 1000$  muestras de  $\{(x_{i1}, x_{i2}, y_i)\}$ . Para cada muestra, se calcula el estimador del total poblacional como en la ecuación (6.19).

Se usan dos tamaños de muestra,  $n_1 = 100$  y  $n_2 = 200$ , se usan dos varianzas del error  $\sigma_1^2 = (0.4)^2$  y  $\sigma_2^2 = 1$ . En este estudio de simulación, se usa como grado del modelo de splines,  $p = 1$  y  $K = 25$  nodos para cada penalización.

Se calculan las estimaciones de las varianzas como:  $\widehat{V}_{\lambda_{opt,1}}$  como en la Ecuación (6.11),  $\widehat{V}_{\lambda_{opt,2}}$  como la Ecuación (6.11) y g-weights,  $\widehat{V}_{\lambda_{opt,3}}$  como la Ecuación (6.18) y  $\widehat{V}_{\lambda_{opt,4}}$  como la Ecuación (6.18) y g-weights.

Adicionalmente, en este estudio de simulación se usan, el Modelo Lineal (LM) (Ecuación (3.11)) y el Modelo Aditivo (AM) (Ecuación (6.15)). Para poder comparar los modelos, se usan los mismos grados libertad para todos los modelos, es decir, si para los modelos aditivos y de coeficientes variantes es de 5, se usa un modelo lineal de grado 4.

Una medida de eficiencia de un estimador es el Sesgo Relativo (*%RB*) de un estimador  $\widehat{\theta}$  del parámetro poblacional  $\theta$ , el cual se define como:

$$\%RB(\widehat{\theta}) = \frac{E_p(\widehat{\theta}) - \theta}{\theta} \times 100. \quad (6.24)$$

Es posible calcular  $\%RB(\widehat{\theta})$ . Supongamos que  $\theta = V_i$ . En un experimento Monte Carlo de  $M$  réplicas, se tiene que:

$$\%RB(\widehat{V}_i) \approx \frac{\frac{1}{M} \sum_{m=1}^M \widehat{V}_i^{(m)} - \frac{1}{M-1} \sum_{m=1}^M \left( \hat{t}_i^{(m)} - \bar{\hat{t}}_i \right)^2}{\frac{1}{M-1} \sum_{m=1}^M \left( \hat{t}_i^{(m)} - \bar{\hat{t}}_i \right)^2} \times 100. \quad (6.25)$$

Otra medida de eficiencia de un estimador, es la raíz cuadrada del Error Cuadrático Medio Relativo (*RMSE*), el cual fue definido en la Sección 5.3.1.

A continuación se presentan y describen los resultados obtenidos del estudio de simu-

## 6.4. Estudio de simulación

lación.

En la Tabla 6.1 se muestran los valores de  $RMSE$  y  $\%RB$ . Para el modelo  $VC$ , en la mayoría de los casos, los valores de  $RMSE$  son menores en comparación con el modelo  $LM$ . En algunos casos, las estimaciones del total poblacional son eficientes usando los modelos  $VC$  y  $AM$ . Para la población de coeficientes variantes ( $IV$ ), es más eficiente el modelo de coeficientes variantes que los modelos  $AM$  y  $LM$ .

Adicionalmente se observó que el estimador del total usando el modelo de coeficientes variantes es más eficiente que el estimador de HT, el cual no usa información auxiliar.

**Tabla 6.1:**  $\sqrt{RMSE(\hat{t}_{y,VC})}$  y  $\%RB(\hat{t}_{y,VC})$ .

Pob.	$\sigma$	$n$	$VC$	$AM$	$LM$	$VC$	$AM$	$LM$
I	0.4	100	0.0174	0.0080	0.0089	-0.4865	0.0355	0.1093
	0.4	200	0.0108	0.0050	0.0056	-0.3286	-0.0036	0.0247
	1.0	100	0.0242	0.0191	0.0197	-0.6082	-0.0388	0.0200
	1.0	200	0.0154	0.0127	0.0129	-0.2350	0.0089	0.0249
II	0.4	100	0.0194	0.0232	0.0195	-0.1078	0.0120	-0.1699
	0.4	200	0.0121	0.0143	0.0121	-0.0657	-0.0029	-0.0779
	1.0	100	0.0267	0.0285	0.0265	0.0046	0.0270	-0.1040
	1.0	200	0.0174	0.0185	0.0168	-0.1009	-0.0615	-0.1197
III	0.4	100	0.0080	0.0091	0.0097	-0.0305	-0.0150	0.0484
	0.4	200	0.0052	0.0059	0.0062	-0.0217	-0.0017	0.0259
	1.0	100	0.0202	0.0205	0.0213	0.0774	0.0551	0.1104
	1.0	200	0.0136	0.0139	0.0140	0.0200	0.0158	0.0481
IV	0.4	100	0.0060	0.0061	0.0062	0.0236	0.0214	0.0154
	0.4	200	0.0043	0.0043	0.0043	0.0006	0.0003	0.0021
	1.0	100	0.0161	0.0163	0.0167	-0.0225	-0.0218	-0.0068
	1.0	200	0.0107	0.0107	0.0107	-0.0042	-0.0059	-0.0148

En la Tabla 6.2 se muestran los valores  $\%RB$  usando los 4 de tipos de varianza mencionado anteriormente, usando el modelo  $VC$ . Se observa que los primeros tres estimadores de la varianza subestiman el valor de la varianza. La primera forma de estimar la varianza es la que se propusó en este trabajo. También se presentan otras formas de calcular la varianza del total poblacional.

En la Tabla 6.3 se muestran las probabilidades de cobertura, usando el modelo  $VC$  y los 4 tipos de varianza. Se muestra que para el caso de los estimadores  $\hat{V}_{\lambda_{opt,3}}$  y  $\hat{V}_{\lambda_{opt,4}}$  y las tres poblaciones, en la mayoría de los casos, las probabilidades de cobertura están más cerca de la nominal de 0.95. Note que conforme se pasa del estimador  $\hat{V}_{\lambda_{opt,1}}$  a  $\hat{V}_{\lambda_{opt,4}}$  las probabilidades de cobertura se van acercando al valor nominal. También, cuando el tamaño de muestra aumenta, las probabilidades aumentan y se acercan a la nominal.



## 6.4. Estudio de simulación

**Tabla 6.2:** Modelos de Coeficientes Variantes:  $\%RB(\widehat{V}_{\lambda_{opt},i})$

Pob.	$\sigma$	$n$	$\%RB(\widehat{V}_{\lambda_{opt},1})$	$\%RB(\widehat{V}_{\lambda_{opt},2})$	$\%RB(\widehat{V}_{\lambda_{opt},3})$	$\%RB(\widehat{V}_{\lambda_{opt},4})$
I	0.4	100	-29.71	-22.58	-13.48	-1.50
	0.4	200	-16.16	-12.10	-6.72	-1.35
	1.0	100	-16.08	-9.99	2.67	13.87
	1.0	200	-6.20	-2.85	4.19	8.74
II	0.4	100	-29.80	-22.48	-10.06	4.24
	0.4	200	-15.26	-10.33	-3.03	3.98
	1.0	100	-21.06	-16.08	-3.93	5.95
	1.0	200	-12.36	-9.39	-2.75	1.28
III	0.4	100	-26.31	-21.66	-3.59	7.48
	0.4	200	-16.69	-13.78	-3.36	1.44
	1.0	100	-15.80	-12.02	0.98	7.79
	1.0	200	-13.79	-11.55	-4.98	-1.93
IV	0.4	100	-2.88	1.20	12.96	19.65
	0.4	200	-10.66	-9.03	-3.87	-1.81
	1.0	100	-13.33	-10.08	1.87	7.52
	1.0	200	-8.71	-6.67	-1.04	1.63

**Tabla 6.3:** Modelos de Coeficientes Variantes. Probabilidad de Cobertura.

Pob.	$\sigma$	$n$	$CP(\widehat{V}_{\lambda_{opt},1})$	$CP(\widehat{V}_{\lambda_{opt},2})$	$CP(\widehat{V}_{\lambda_{opt},3})$	$CP(\widehat{V}_{\lambda_{opt},4})$
I	0.4	100	0.885	0.899	0.920	0.930
	0.4	200	0.920	0.926	0.941	0.948
	1.0	100	0.919	0.928	0.950	0.960
	1.0	200	0.939	0.941	0.947	0.951
II	0.4	100	0.890	0.906	0.933	0.945
	0.4	200	0.917	0.931	0.943	0.953
	1.0	100	0.903	0.912	0.940	0.946
	1.0	200	0.934	0.942	0.945	0.948
III	0.4	100	0.902	0.913	0.938	0.951
	0.4	200	0.920	0.927	0.944	0.951
	1.0	100	0.916	0.922	0.939	0.949
	1.0	200	0.935	0.938	0.947	0.949

## 6.5. Aplicación a datos reales

---

Los métodos semiparamétricos (*VC* y *AM*) son eficientes, y en algunos casos uno es mejor que el otro, en términos de eficiencia. De nueva cuenta se confirma lo observado en el Capítulo 4, de manera empírica, los estimadores semiparamétricos son más eficientes que los estimadores de regresión paramétricos cuando el modelo paramétrico se especifica inapropiadamente.

## 6.5. Aplicación a datos reales

Los datos consisten en una muestra aleatoria estratificada de las redes de arrastre en el sur del Golfo de San Lorenzo, Canadá. Los peces son atrapados en una gran red, llamada red de arrastre. La profundidad y la latitud son para las ubicaciones de la red de arrastre de la muestra y para la población. El objetivo es emplear la metodología descrita en este trabajo y estimar el total de la *Biomasa*, usando la relación entre la variable de interés  $y = \text{Biomasa}$ , y dos variables auxiliares  $x_1 = \log(\text{depth})$  y  $x_2 = \text{Lat}$ . Por lo tanto, se usa el método descrito en este trabajo para ajustar el modelo de la forma:

$$\text{Biomasa}_k = \left[ \beta_0 + \beta_1 x_{1k} + \sum_{j=1}^K b_j (x_{1k} - \gamma_j)_+ \right] + \left[ \beta_2 + \beta_3 x_{1k} + \sum_{j=1}^K b_j^* (x_{1k} - \gamma_j)_+ \right] * x_{2k} + \epsilon_k.$$

La correspondiente estimación del total de la *Biomasa* y su error estándar están dados por, ( $\hat{\lambda}_{opt} = (\mathbf{11.44}, \mathbf{11,269,080})$ ,  $df = df_1 + df_2 = 7 + 2 = 9$ .)

$$\hat{t}_{\text{Biomasa}}(\hat{\lambda}_{opt}) = 20,940,282, \quad (6.26)$$

$$EE_1 \left[ \hat{t}_{\text{Biomasa}}(\hat{\lambda}_{opt}) \right] = 1,306,298 \text{ (usando } \hat{V}_{\hat{\lambda}_{opt},3}), \quad (6.27)$$

$$EE_2 \left[ \hat{t}_{\text{Biomasa}}(\hat{\lambda}_{opt}) \right] = 1,291,010 \text{ (usando } \hat{V}_{\hat{\lambda}_{opt},1}). \quad (6.28)$$

Hay que notar que los errores estándar son parecidos. Uno esperaría que el error estándar que usa CV fuese mucho mayor, pero es muy similar al otro valor. De igual manera se obtienen los grados de libertad individuales para cada variable auxiliar. Para la variable el logaritmo de la profundidad le corresponden 7 grados de libertad y para la variable Latitud le corresponden 2 grados de libertad. También se determinaron los grados de libertad basado en la muestra y se obtuvo un valor a aproximado a 9, esto afirma lo propuesto en el Capítulo 5.

## 6.6. Conclusiones

En este trabajo, se propuso una forma de obtener los parámetros de suavizamiento usando validación cruzada. De igual manera, se propuso un estimador del total poblacional similar al obtenido en (Breidt *et al.*, 2005). Se estudiaron algunas propiedades estadísticas del estimador del total poblacional. Los resultados del experimento de simulación muestran que la propuesta de este trabajo mejora la precisión de los estimadores usando información auxiliar multivariada. Los modelos de coeficientes variantes contribuyen a los métodos de regresión semiparamétrica en muestreo de poblaciones finitas. Los modelos de coeficientes variantes propocionan una forma de identificar las relaciones existentes entre las variables. La estimación de la varianza usando validación cruzada y g-weights trabajaron muy bien en los estudios de estudios de simulación y aplicación. El uso de validación cruzada evita el problema del sobreajuste.

# Capítulo 7

## Conclusiones y trabajos futuros

De acuerdo a la revisión y réplica del estudio de simulación de los trabajos de (Breidt y Opsomer, 2000) y (Breidt *et al.*, 2005), los experimentos de simulación muestran evidencia empírica que los estimadores semiparamétricos son más eficientes que los estimadores de regresión paramétricos cuando el modelo paramétrico se especifica incorrectamente y son aproximadamente eficientes cuando la especificación paramétrica es la correcta.

En este trabajo se propuso una forma de seleccionar los parámetros de suavizamiento en el contexto de muestreo. Se observó de manera empírica que no hay pérdida significativa en el estimador del total poblacional cuando se usa el parámetro de suavizamiento basado en la muestra. Lo anterior se corroboró con un estudio de simulación y una aplicación a datos reales obtenidos por el INEGI.

Los aportes principales de este trabajo son: se propuso una forma de obtener los parámetros de suavizamiento usando validación cruzada asistida por modelos, se propuso un estimador del total poblacional, se demostró que el estimador del total poblacional propuesto, bajo ciertos supuestos, es consistente en diseño y asintóticamente normal. También se propuso otra forma de estimar la varianza, la cual usa CV y g-weights. De igual manera, en este trabajo los experimentos de simulación muestran que la propuesta de este trabajo mejora la precisión de los estimadores usando información auxiliar multivariada en comparación con los modelos paramétricos.

Los modelos de coeficientes variantes contribuyen a los métodos de regresión semiparamétrica en muestreo de poblaciones finitas.

Como futuro trabajo de investigación es adaptar otros métodos de regresión semiparamétrica en el contexto de muestreo de poblaciones finitas. En este trabajo, los coeficientes variantes fueron aproximados por splines que involucraba trabajar con los parámetros de suavizamiento. De igual manera, los coeficientes de regresión pueden ser aproximados por regresión local polinomial o algún otro método semiparamétrico univariado. En este

## 7. Conclusiones y trabajos futuros

---

mismo sentido, el uso de regresión local polinomial implica trabajar con más de un ancho de banda, entonces siguiendo la misma idea este trabajo es posible seleccionar de manera óptima los anchos de banda. También es de interés estudiar las propiedades estadísticas del estimador de la varianza que usa CV y g-weights, el cuál es otra forma de estimar una varianza del total poblacional.

En otra dirección, los modelos de coeficientes variantes no han sido estudiados en el enfoque basado en modelos. Una posible línea de investigación es estudiar estos modelos ya sea con enfoque Clásico o Bayesiano. En este caso, los valores de la variable de interés son tomados como una realización de variables aleatorias.

# Referencias

- Aerts, M., Claeskens, G. y Wand, M. P. (2002). Some theory for penalized spline generalized additive models. *Journal of Statistical Planning and Inference*, 103, 455–470.
- Breidt, F. J., Claeskens, G. y Opsomer, J. D. (2005). Model-Assisted estimation for complex surveys using penalized splines. *Biometrika*, 92, 831–846.
- Breidt, F. J. y Opsomer, J. D. (2000). Local Polynomial Regression estimators in survey sampling. *Ann. Statist.*, 28, 1026–1053.
- Brewer, K. (2002). *Combined sampling inference, weighting Basu's elephants*. London:Arnold.
- Cai, Z., Fan, J. y Li, R. (2000a). Efficient estimation and inferences for varying-coefficient models. *Journal of the American Statistical Association*, 95, 888–902.
- Cai, Z., Fan, J. y Yao, Q. (2000b). Functional-Coefficient Regression Models for Nonlinear Time Series. *Journal of the American Statistical Association*, 95, 941–956.
- Cao, Y., Lin, H., Wu, T. Z. y Yu, Y. (2010). Penalized spline estimation for functional coefficient regression models. *Computational Statistics and Data Analysis*, 54, 891–905.
- Cassel, C., Sarndal, C. y Wretman, J. (1976). *Foundations of Inference in Survey Sampling*. Wiley.
- Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74, 829:836.
- Cochran, W. (1977). *Sampling Techniques*. Wiley.
- Eilers, P. H. C. y Marx, B. D. (1996). Flexible smoothing with B-splines and penalties (with discussion). *Statistical Science*, 11, 89–121.
- Eubank, R. L. (1994). A simple smoothing spline. *American Statistician*, 48, 103–106.
- Eubank, R. L. (1999). *Nonparametric regression and spline smoothing*. New York: Marcel Dekker.
- Fan, J. (1993). Local linear regression smoothers and their minimax efficiencies. *Annals of St.*, 21, 196–216.
- Fan, J. y Gijbels, I. (1996). *Local Polynomial Modelling and Its Applications*. London: Chapman & Hall.

## Referencias

---

- Fan, J. y Zhang, W. (1999). Statistical estimation in varying coefficient models. *The Annals of Statistics*, 27, 1491–1518.
- Fan, J. y Zhang, W. (2008). Statistical methods with varying coefficient models. *Statistics and its Interface*, 1, 179–195.
- Fuller, W. A. (2009). *Sampling Statistics*. Wiley.
- Green, P. J. y Silverman, B. W. (1994). *Nonparametric regression and generalized linear models*. London: Chapman & Hall.
- Gutiérrez, H. A. (2009). *Estrategias de muestreo. Diseño de encuestas y estimación de parámetros*. Universidad Santo Tomás.
- Hardle, W. (1990). *Applied Non-parametric Regression*. Cambridge University Press.
- Hastie, T. y Tibshirani, R. (1993). Varying-Coefficient Models. *Journal of the Royal Statistical Society. Series B(Methodological)*, 55, 757–796.
- Hastie, T. J. (1996). Pseudosplines. *Journal of the Royal Statistical Society. Series B(Methodological)*, 58, 379–396.
- Hoover, D. R., Rice, J. A., Wu, C. O. y Yang, L. P. (1998). Nonparametric smoothing estimates of time-varying coefficient models with longitudinal data. *Biometrika*, 85, 809–822.
- Horvitz, D. G. y Thompson, D. J. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American Sta*, 47, 663–685.
- Hurvich, C. M., Simonoff, J. S. y Tsai, C. L. (1998). Smoothing parameter selection in nonparametric regression using an improved Akaike information criterion. *Journal of Royal Statistical Society, Serie B*, 60, 271–293.
- Huskova, M., Beran, R. y Dupac, V. (1998). *Collected Works of Jaroslav Hájek: With commentary*. Wiley.
- Isaki, C. T. y Fuller, W. A. (1982). Survey design under the regression superpopulation model. *Journal of the American Statistical Association*, 77, 89–96.
- Loader, C. (1999). *Local regression and Likelihood*. Springer-Verlag.
- Lohr, S. (2000). *Sampling: Design and Analysis*. Thompson.
- Lu, Y., Zhang, R. y Zhu, L. (2008). Penalized spline estimation for varying-coefficient models. *Communications in Statistics - Theory and Methods*, 37, 2249:2261.
- Matei, A. y Tillé, Y. (2005). Evaluation of variance approximations and estimators in maximum entropy sampling with unequal probability and fixed sample size. *Journal of Official Statistics*, 21, 543–570.
- Montanari, G. E. y Ranalli, M. G. (2005). Nonparametric model calibration estimation in survey sampling. *Journal of the American Statistical Association*, 100, 1429–1442.

## Referencias

---

- Opsomer, J. D., Claeskens, G., Ranalli, M. G., Kaurmann, G. y Breidt, F. J. (2008). Non-parametric small area estimation using penalized spline regression. *Journal of the Royal Statistical Society. Series B(Methodological)*, 70, 265–286.
- Opsomer, J. D. y Miller, C. (2005). Selecting the amount of smoothing in nonparametric regression estimation for complex surveys. *Nonparametric Statistics*, 17, 593–611.
- O’Sullivan, F. (1986). A statistical perspective on ill-posed inverse problems (with discussion). *Statistical Science*, 1, 505–527.
- Rueda, M. y Sánchez-Borrego, I. (2009). A predictive estimator of finite population mean using nonparametric regression. *Comput Stat*, 24, 1–14.
- Rueda, M., Sánchez-Borrego, I., Arcos, A. y Martínez, S. (2010). Model-calibration estimation of the distribution function using nonparametric regression. *Metrika*, 71, 33–44.
- Ruppert, D. (1997). Empirical-bias bandwidths for local polynomial nonparametric regression and density estimation. *Journal of the American Statistical Association*, 92, 1049–1062.
- Ruppert, D. y Carroll, R. J. (2000). Spatially-adaptive penalties for spline fitting. *Australian and New Zealand Journal of Statistics*, 42, 205–223.
- Ruppert, D., Wand, M. y Carroll, R. J. (2003). *Semiparametric Regression*. Cambridge University Press, Cambridge UK, primera edición.
- Sarndal, C. E., Swensson, B. y Wretman, J. (1992). *Model Assisted Survey Sampling*. Springer Verlag, New York, primera edición.
- Schwarz, G. E. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6, 461–464.
- Sen, A. R. (1953). On the estimate of the variance in sampling with varying probabilities. *Journal of the Indian Society in Agricultural Statistics*, 5, 119–127.
- Serfling, R. J. (1980). *Approximation Theorems of Mathematical Statistics*. Wiley.
- Stone, C. J. (1977). Consistent nonparametric regression. *Annals of S*, 5, 595–620.
- Sunter, A. B. (1977). List sequential sampling with equal or unequal probabilities without replacement. *Applied Statistics*, 26, 261–268.
- Thompson, M. E. (1997). *Theory of Sample Surveys*. Chapman & Hall.
- Tillé, Y. (2006). *Sampling Algorithms*. Springer.
- Vaart, A. W. V. D. (1998). *Asymptotic Statistics*. Cambridge University Press.
- Wand, F. T. y Scott, D. W. (1994). The  $L_1$  method for robust nonparametric regression. *Journal of the American Statistical Association*, 89, 65–76.
- Wand, M. P. y Jones, M. C. (1995). *Kernel Smoothing*. London: Chapman & Hall.
- Wang, L. (2009). Single-index model-assisted estimation in survey sampling. *J. Nonparametr. Stat.*, 21, 487–504.



## Referencias

---

- Wang, L. y Wang, S. (2011). Nonparametric Additive model-assisted estimation for survey data. *Journal of Multivariate Analysis*, 102, 1126–1140.
- Yates, F. y Grundy, P. (1953). Selection without replacement from within strata with probability proportional to size. *Journal of the Royal Statistical Society. Series B(Methodological)*, 15, 235–261.
- Zhang, H. G. y Mei, C. L. (2012). SiZer Inference for varying coefficient models. *Communications in Statistics - Simulation and Computation*, 41, 1944–1959.

# Anexos

## Anexo A: Códigos de Programas en R usados en el Capítulo 4

```
#### Dr. Luis Fernando Contreras Cruz, lufecon@gmail.com ####
```

```
N <- 1000
M <- 1000
n <- 50
s1 <- 0.1
s2 <- 0.4
h1 <- 0.2943
h2 <- 0.135
```

```
Xk <- matrix(0,N,1)
Yi1 <- matrix(0,N,1)
TOTALYi1k.Est <- matrix(0,M,1)
TOTALYi1k.EstSpl <- matrix(0,M,1)
TOTALYi1k.EstHT <- matrix(0,M,1)
TOTALYi1k.EstLIN <- matrix(0,M,1)
TOTALYi1k.EstRLL <- matrix(0,M,1)
MatrizX <- matrix(0,M,n)
XXU <- matrix(0,M,M)
WUi <- matrix(0,M,M)
mmi <- matrix(0,M,M)
```

```
Xk <- runif(N)
m1 <- 1+2*(Xk-0.5)
m2 <- 1+2*((Xk-0.5)^2)
m3 <- 1+2*(Xk-0.5)+exp(-200*((Xk-0.5)^2))
m4 <- ifelse(Xk<=0.65,1+2*(Xk-0.5),0.65)
m5 <- ifelse(m1+rnorm(N,mean = 0, sd=s1)<=1.5,1,0)
m6 <- exp(-8*Xk)
m7 <- 2+sin(2*pi*Xk)
m8 <- 2+sin(8*pi*Xk)
```

```
est1<- which(Xk <=1/7)
est2<- which(Xk>1/7 & Xk <=2/7)
est3<- which(Xk>2/7 & Xk <=3/7)
est4<- which(Xk>3/7 & Xk <=4/7)
est5<- which(Xk>4/7 & Xk <=5/7)
est6<- which(Xk>5/7 & Xk <=6/7)
est7<- which(Xk>6/7 & Xk <=7/7)
```

```
nh <- matrix(c(round((length(est1)/N)*n,0),round((length(est2)/N)*n,0),round((length(est3)/N)*n,0),
round((length(est4)/N)*n,0),round((length(est5)/N)*n,0),round((length(est6)/N)*n,0),
```

## Anexos

```
round((length(est7)/N)*n,0),7,1)

Nh <- matrix(c(length(est1),length(est2),length(est3),length(est4),length(est5),length(est6),
length(est7)),7,1)

MatrizX1 <- matrix(0,M,nh[1],0)
MatrizX2 <- matrix(0,M,nh[2],0)
MatrizX3 <- matrix(0,M,nh[3],0)
MatrizX4 <- matrix(0,M,nh[4],0)
MatrizX5 <- matrix(0,M,nh[5],0)
MatrizX6 <- matrix(0,M,nh[6],0)
MatrizX7 <- matrix(0,M,nh[7],0)
MatrizXXX <- matrix(0,M,n)

Yi1 <- m1 + rnorm(N, mean = 0, sd = s1)
TOTAL.REAL <- sum(Yi1)
XX <- matrix(c(rep(1,N),Xk,Xk^2,Xk^3,Xk^4,Xk^5,Xk^6),1000,7)
XXLIN <- XX[,1:2]

NODOS <- quantile(Xk,probs = seq(0/25, 24/25, 1/25))
MatrizNODOS <- matrix(rep(NODOS,1000),1000,25,byrow=T)
MatrizXX <- matrix(rep(Xk,25),1000,25,byrow=F)
ALFA <- 4.699
ALFA2 <- 0.299
MAT1 <- matrix(rep(1,1000),,1)
X1 <- matrix(cbind(MAT1,Xk,MatrizXX-MatrizNODOS),1000,)
XX1 <- ifelse(X1<0,X1==0,X1)
WS <- diag(c(rep((Nh/nh)[1],nh[1]),rep((Nh/nh)[2],nh[2]),rep((Nh/nh)[3],nh[3]),
rep((Nh/nh)[4],nh[4]),rep((Nh/nh)[5],nh[5]),
rep((Nh/nh)[6],nh[6]),rep((Nh/nh)[7],nh[7]))))

Aalfa1 <- diag(c(0,0,rep(1,25)),27,27)
S_ALFA <- XX1%*%solve(t(XX1)%*%XX1+(ALFA2)*Aalfa1,t(XX1))
GrLib <- sum(diag(S_ALFA))
GrLib

for (m in 1:M)
{
MatrizX1[m,] <- sample(est1, nh[1], replace = FALSE, prob = NULL)
MatrizX2[m,] <- sample(est2, nh[2], replace = FALSE, prob = NULL)
MatrizX3[m,] <- sample(est3, nh[3], replace = FALSE, prob = NULL)
MatrizX4[m,] <- sample(est4, nh[4], replace = FALSE, prob = NULL)
MatrizX5[m,] <- sample(est5, nh[5], replace = FALSE, prob = NULL)
MatrizX6[m,] <- sample(est6, nh[6], replace = FALSE, prob = NULL)
MatrizX7[m,] <- sample(est7, nh[7], replace = FALSE, prob = NULL)
MatrizXXX[m,] <- c(MatrizX1[m,],MatrizX2[m,],MatrizX3[m,],MatrizX4[m,],
MatrizX5[m,],MatrizX6[m,],MatrizX7[m,])

Tx <- matrix(colSums(XX),1,7)
TOTALYi1k.Est[m] <- Tx[,1:7]%*%solve(t(XX[MatrizXXX[m,],1:7])%*%XX[MatrizXXX[m,],1:7],
t(XX[MatrizXXX[m,],1:7])%*%Yi1[MatrizXXX[m,]])

TOTALYi1k.EstSpl[m] <- t(colSums(XX1))%*%(solve(t(XX1[MatrizXXX[m,],])%*%WS%*%XX1[MatrizXXX[m,],]+
(ALFA2)*Aalfa1,t(XX1[MatrizXXX[m,],])%*%WS%*%Yi1[MatrizXXX[m,]]))

TOTALYi1k.EstLIN[m] <- Tx[,1:2]%*%solve(t(XXLIN[MatrizXXX[m,],])%*%XXLIN[MatrizXXX[m,],],
t(XXLIN[MatrizXXX[m,],])%*%Yi1[MatrizXXX[m,]])

TOTALYi1k.EstHT[m] <- Nh[1]*mean(Yi1[MatrizX1[m,]])+ Nh[2]*mean(Yi1[MatrizX2[m,]])+ Nh[3]*mean(Yi1[MatrizX3[m,]])+
Nh[4]*mean(Yi1[MatrizX4[m,]])+ Nh[5]*mean(Yi1[MatrizX5[m,]])+ Nh[6]*mean(Yi1[MatrizX6[m,]])+
Nh[7]*mean(Yi1[MatrizX7[m,]])

}
}
```

## Anexos

---

```
ECM      <- (1/M)*sum((TOTALYi1k.Est-TOTAL.REAL)^2)
ECM
ECMSpl   <- (1/M)*sum((TOTALYi1k.EstSpl-TOTAL.REAL)^2)
ECMSpl
ECMLIN   <- (1/M)*sum((TOTALYi1k.EstLIN-TOTAL.REAL)^2)
ECMLIN
ECMHT    <- (1/M)*sum((TOTALYi1k.EstHT-TOTAL.REAL)^2)
ECMHT

razonECM1 <- ECM/ECMSpl
round(razonECM1,2)
razonECM2 <- ECMLIN/ECMSpl
round(razonECM2,2)
razonECM3 <- ECMHT/ECMSpl
round(razonECM3,2)

EREL     <- (1/M)*sum(abs(TOTALYi1k.Est-TOTAL.REAL)/TOTAL.REAL)
EREL
ERELSpl  <- (1/M)*sum(abs(TOTALYi1k.EstSpl-TOTAL.REAL)/TOTAL.REAL)
ERELSpl
EREL.LIN <- (1/M)*sum(abs(TOTALYi1k.EstLIN-TOTAL.REAL)/TOTAL.REAL)
EREL.LIN
ERELHT   <- (1/M)*sum(abs(TOTALYi1k.EstHT-TOTAL.REAL)/TOTAL.REAL)
ERELHT

#### Dr. Luis Fernando Contreras Cruz, lufecon@gmail.com ####

N <- 1000
M <- 1000
n <- 50
s1 <- 0.1
s2 <- 0.4
h1 <- 0.2943
h2 <- 0.135

Xk          <- matrix(0,N,1)
Yi1         <- matrix(0,N,1)
TOTALYi1k.EstSpl <- matrix(0,M,1)
TOTALYi1k.EstRLL <- matrix(0,M,1)
MatrizX     <- matrix(0,M,n)
XXU         <- matrix(0,M,M)
WUi        <- matrix(0,M,M)
MMi        <- matrix(0,M,M)

Xk <- runif(N)
m1 <- 1+2*(Xk-0.5)
m2 <- 1+2*((Xk-0.5)^2)
m3 <- 1+2*(Xk-0.5)+exp(-200*((Xk-0.5)^2))
m4 <- ifelse(Xk<=0.65,1+2*(Xk-0.5),0.65)
m5 <- ifelse(m1+rnorm(N,mean = 0, sd=s1)<=1.5,1,0)
m6 <- exp(-8*Xk)
m7 <- 2+sin(2*pi*Xk)
m8 <- 2+sin(8*pi*Xk)

Yi1 <- m6+ rnorm(N, mean = 0, sd = s2)
TOTAL.REAL <- sum(Yi1)

for (j in 1:M)
{
  XXU[,j] <- Xk-Xk[j]
}
XXUi <- cbind(rep(1,1000),XXU)
```

```

for (j in 1:M)
{
  WUi[,j] <- ifelse(abs(XXU[,j]/h2)<=1,(N/(n*h2))*0.75*(1-(XXU[,j]/h2)^2),0)
}

NODOS      <- quantile(Xk,probs = seq(0/25, 24/25, 1/25))
MatrizNODOS <- matrix(rep(NODOS,1000),1000,25,byrow=T)
MatrizXX    <- matrix(rep(Xk,25),1000,25,byrow=F)
  ALFA  <- 4.699
  ALFA2 <- 0.299
  MAT1  <- matrix(rep(1,1000),,1)
  X1    <- matrix(cbind(MAT1,Xk,MatrizXX-MatrizNODOS),1000,)
  XX1   <- ifelse(X1<0,X1==0,X1)
  Ws    <- diag(N/n,50)
  Aalfa1 <- diag(c(0,0,rep(1,25)),27,27)
  S_ALFA <- XX1%*%solve(t(XX1)%*%XX1+(ALFA2)*Aalfa1,t(XX1))
  GrLib  <- sum(diag(S_ALFA))
  GrLib

for (m in 1:M)
{
  MatrizX[m,] <- sample(1:N, n, replace = FALSE, prob = NULL)

  for (h in 1:M)
  {
    MMi[m,h] <- lm(Yi1[MatrizX[m,]]^XXUi[MatrizX[m,],c(1,h+1)],weights= WUi[MatrizX[m,],h])$coefficients[1]
  }

  TOTALYi1k.EstSpl[m] <- t(colSums(XX1))%*%(solve(t(XX1[MatrizX[m,],])%*%Ws%*%XX1[MatrizX[m,],]+
    (ALFA2)*Aalfa1,t(XX1[MatrizX[m,],])%*%Ws%*%Yi1[MatrizX[m,]]))
  TOTALYi1k.EstRLL[m] <- sum(MMi[m,])+(N/n)*sum(Yi1[MatrizX[m,]]-MMi[m,MatrizX[m,]])
}

ECMSpl <- (1/M)*sum((TOTALYi1k.EstSpl-TOTAL.REAL)^2)
ECMSpl
ECMRLl <- (1/M)*sum((TOTALYi1k.EstRLL-TOTAL.REAL)^2)
ECMRLl

razonECM4 <- ECMRLl/ECMSpl
round(razonECM4,2)

```

## Anexo B: Códigos de Programas en R usados en el Capítulo 5

```
#### Dr. Luis Fernando Contreras Cruz, lufecon@gmail.com ####

#####
# TEMA: SIMULACIÓN DE GRADOS DE LIBERTAD (7 Y 11) Y      #
# PENALIZACIÓN BAJO DOS DISEÑOS DE MUESTREO          #
# (MSR Y MPD) CON DOS TAMAÑOS DE MUESTRA (n1 y n2). #
#####

N <- 3000 # Tamaño de la población
n1 <- 150  # Tamaño de la muestra 1
n2 <- 300  # Tamaño de la muestra 2
g1_7 <- 7  # Grados de libertad equivalentes
g1_11 <- 11 # Grados de libertad equivalentes
J <- min(N/4,35) # Número de nodos usados en las dos poblaciones

rho <- 0.75 # Correlación entre las variables X y Z, Y e W.

X1 <- runif(N) # Población 1
Z1 <- runif(N) # Población auxiliar 1
X <- X1
Z <- rho*X + Z1*sqrt(1-rho^2) # Variable tamaño para X

write(X, file="DatosX_art.txt",ncol=1)
write(Z, file="DatosZ_art.txt",ncol=1)

Y2 <- rexp(N,2) # Población 2
W2 <- rexp(N,2) # Población auxiliar 2
Y <- Y2
W <- rho*Y + W2*sqrt(1-rho^2) # Variable tamaño para Y

write(Y, file="DatosY_art.txt",ncol=1)
write(W, file="DatosW_art.txt",ncol=1)

#####
## Cálculo de NODOS ##
#####

Nodos_X <- quantile(X,probs = seq(1/(J+1), J/(J+1), 1/(J+1)))
Nodos_Y <- quantile(Y,probs = seq(1/(J+1), J/(J+1), 1/(J+1)))

#####
## Matriz de Nodos ##
#####

Mat_NODOS_X <- matrix(rep(Nodos_X,N),N,J,byrow=T)
Matriz_X <- matrix(rep(X,J),N,J,byrow=F)

Mat_NODOS_Y <- matrix(rep(Nodos_Y,N),N,J,byrow=T)
Matriz_Y <- matrix(rep(Y,J),N,J,byrow=F)

#####
##### MATRIZ DISEÑO X e Y #####
#####

X_1 <- matrix(cbind(rep(1,N),X,X^2,X^3,(Matriz_X-Mat_NODOS_X)^3),N,)
```

## Anexos

---

```
X_u <- ifelse(X_1<0,X_1==0,X_1)

Y_1 <- matrix(cbind(rep(1,N),Y,Y^2,Y^3,(Matriz_Y-Mat_NODOS_Y)^3),N,)
Y_u <- ifelse(Y_1<0,Y_1==0,Y_1)

#####
#### Demmler-Reinsch Orthogonalization ####
#####

##### 1. Cholesky #####

XT.X <- t(Y_u)%*%Y_u
Chol <- chol(XT.X+10^(-10)*D) ##Chol= R, R^T R=XT.X ##

CHO <- t(Chol)%*%Chol

##### 2. SVD #####

D <- diag(c(rep(0,4),rep(1,J)))
H <- solve(t(Chol))%*%D%*%solve(Chol)

s <- svd(H)$d

##### 3. df #####

gl <- 11
df <- function(lambda)
{
  df.lambda <- sum(1/(1+lambda*s))
  equation <- gl-df.lambda

  #return(df.lambda)
  return(equation)
}

##### 4. Solve equation for lambda #####

(lamda_u <- uniroot(df, c(1e-10,1),tol = 1e-10)$root)

#####
#### Matriz de Suavizamiento ####
#####

A_lamda <- function(lamda,J)
{
  diag(c(rep(0,4),rep(lamda,J)))
}

#####
#### MÉTODO DE BISECCIÓN ####
#####

Biseccion <- function(ftn, x.l, x.r, tol = 1e-6)
{
  # aplica el algoritmo de bisección para encontrar x tal que ftn(x) == 0
  # asumimos que ftn es una función univariable
  # x.l y x.r deden contener la raíz buscada, es decir
  # x.l < x.r y ftn(x.l) * ftn(x.r) < 0
  # el algoritmo refina iterativamente x.l y x.r y termina cuando
  # x.r - x.l <= tol
```

## Anexos

---

```
# chequea las entradas
if (x.l >= x.r)
{
  cat("error: x.l >= x.r \n")
  return(NULL)
}

f.l <- ftn(x.l)
f.r <- ftn(x.r)

if (f.l == 0)
{
  return(x.l)
} else if (f.r == 0)
{
  return(x.r)
} else if (f.l * f.r > 0)
{
  cat("error: ftn(x.l) * ftn(x.r) > 0 \n")
  return(NULL)
}

# refina sucesivamente x.l y x.r
n <- 0
while ((x.r - x.l) > tol)
{
  x.m <- (x.l + x.r)/2
  f.m <- ftn(x.m)
  if (f.m == 0)
  {
    return(x.m)
  } else if (f.l * f.m < 0)
  {
    x.r <- x.m
    f.r <- f.m
  } else
  {
    x.l <- x.m
    f.l <- f.m
  }
}
n <- n + 1
#cat("en la iteración", n, "la raíz vive entre", x.l, "y", x.r, "\n")
}
# regresa (aproximadamente) la raíz
return((x.l + x.r)/2)
}

#####
#####

#####
### Población X ###
#####
### Con l_u=7 y 11 encontrar lamda_u tal que l_u = TRAZA[(X^T*X + A_lamda)^-1 X^T*X]

f1_X_7 <- function(lamda)
{
  XX <- t(X_u)%*%X_u
  return(gl_7-sum(diag(solve(XX + A_lamda(lamda,J))%*%XX)))
}

f1_X_11 <- function(lamda)
{
  XX <- t(X_u)%*%X_u
  return(gl_11-sum(diag(solve(XX + A_lamda(lamda,J))%*%XX)))
}
```



```

(lamda_u_X7 <- uniroot(f1_X_7, c(1e-10,1),tol = 1e-10)$root)
0.001291042

(lamda_u_X11 <- uniroot(f1_X_11, c(1e-10,1),tol = 1e-10)$root)
1.172817e-05

(lamda_u_X7 <- Biseccion(f1_X_7, 1e-10, 1, tol = 1e-10))
0.001291042

(lamda_u_X11 <- Biseccion(f1_X_11,1e-10 , 1, tol = 1e-10))
1.172822e-05

#####.Machine$double.eps , .Machine$double.xmin#####

#####
### Población Y ###
#####
### Con l_u=7 y 11 encontrar lamda_u tal que l_u = TRAZA[(Y^T*Y + A_lamda)^-1 Y^T*Y]

f_Y_7 <- function(lamda)
{
  YtY <- t(Y_u)%*%Y_u
  return(log(gl_7)-log(sum(diag(solve(YtY + A_lamda(lamda,J))%*%YtY))))
}

f_Y_11 <- function(lamda)
{
  YtY <- t(Y_u)%*%Y_u
  return(log(gl_11)-log(sum(diag(solve(YtY + A_lamda(lamda,J))%*%YtY))))
}

(lamda_u_Y7 <- uniroot(f_Y_7, c(1e-6,1), tol = 1e-10)$root)
0.4525088

(lamda_u_Y11 <- uniroot(f_Y_11, c(1e-6,1), tol = 1e-10)$root)
0.0009222457

(lamda_u_Y7 <- Biseccion(f_Y_7,1e-6,1,tol = 1e-10))
0.4525088

(lamda_u_Y11 <- Biseccion(f_Y_11,1e-6,1,tol = 1e-10))
0.0009222457

#####
# PROCESO DE SIMULACIÓN #
#####

#####
### Diseño de Muestreo SR y PD ###
#####

library(sampling)
pik1 <- inclusionprobabilities(Z,n1)
pikt1 <- UPMEpiktildefrompik(pik1)
W1 <- pikt1/(1-pikt1)
q1 <- UPMEqfromw(W1,n1)

pik2 <- inclusionprobabilities(Z,n2)

```

## Anexos

---

```
pikt2 <- UPMEpiktildefrompik(pikt2)
W2 <- pikt2/(1-pikt2)
q2 <- UPMEqfromw(W2,n2)

pikt3 <- inclusionprobabilities(W,n1)
pikt3 <- UPMEpiktildefrompik(pikt3)
W3 <- pikt3/(1-pikt3)
q3 <- UPMEqfromw(W3,n1)

pikt4 <- inclusionprobabilities(W,n2)
pikt4 <- UPMEpiktildefrompik(pikt4)
W4 <- pikt4/(1-pikt4)
q4 <- UPMEqfromw(W4,n2)
```

```
#####
```

```
M <- 5000 # Número de simulaciones
```

```
#####
```

```
gl_s_X7_n1 <- numeric(M)
lam_s_X7_n1 <- numeric(M)

gl_s_X11_n1 <- numeric(M)
lam_s_X11_n1 <- numeric(M)

gl_s_X7_n2 <- numeric(M)
lam_s_X7_n2 <- numeric(M)

gl_s_X11_n2 <- numeric(M)
lam_s_X11_n2 <- numeric(M)

gl_s_Y7_n1 <- numeric(M)
lam_s_Y7_n1 <- numeric(M)

gl_s_Y11_n1 <- numeric(M)
lam_s_Y11_n1 <- numeric(M)

gl_s_Y7_n2 <- numeric(M)
lam_s_Y7_n2 <- numeric(M)

gl_s_Y11_n2 <- numeric(M)
lam_s_Y11_n2 <- numeric(M)

gl_s_X7_n1_pd <- numeric(M)
lam_s_X7_n1_pd <- numeric(M)

gl_s_X11_n1_pd <- numeric(M)
lam_s_X11_n1_pd <- numeric(M)

gl_s_X7_n2_pd <- numeric(M)
lam_s_X7_n2_pd <- numeric(M)

gl_s_X11_n2_pd <- numeric(M)
lam_s_X11_n2_pd <- numeric(M)

gl_s_Y7_n1_pd <- numeric(M)
lam_s_Y7_n1_pd <- numeric(M)

gl_s_Y11_n1_pd <- numeric(M)
lam_s_Y11_n1_pd <- numeric(M)

gl_s_Y7_n2_pd <- numeric(M)
lam_s_Y7_n2_pd <- numeric(M)

gl_s_Y11_n2_pd <- numeric(M)
```

## Anexos

```
lam_s_Y11_n2_pd <- numeric(M)

##### Cuerpo principal #####

i <- 1
while(i <= M)
{
  X_s1 <- sample(1:N,n1)
  X_s2 <- sample(1:N,n1)
  X_s3 <- sample(1:N,n2)
  X_s4 <- sample(1:N,n2)
  Y_s1 <- sample(1:N,n1)
  Y_s2 <- sample(1:N,n1)
  Y_s3 <- sample(1:N,n2)
  Y_s4 <- sample(1:N,n2)

  X_s1_pd <- seq(N) [UPMEsfromq(q1)==1]
  X_s2_pd <- seq(N) [UPMEsfromq(q1)==1]
  X_s3_pd <- seq(N) [UPMEsfromq(q2)==1]
  X_s4_pd <- seq(N) [UPMEsfromq(q2)==1]
  Y_s1_pd <- seq(N) [UPMEsfromq(q3)==1]
  Y_s2_pd <- seq(N) [UPMEsfromq(q3)==1]
  Y_s3_pd <- seq(N) [UPMEsfromq(q4)==1]
  Y_s4_pd <- seq(N) [UPMEsfromq(q4)==1]

  XX_s1 <- t(X_u[X_s1,])%*%diag(N/n1,n1)%*%X_u[X_s1,]
  XX_s2 <- t(X_u[X_s2,])%*%diag(N/n1,n1)%*%X_u[X_s2,]
  XX_s3 <- t(X_u[X_s3,])%*%diag(N/n2,n2)%*%X_u[X_s3,]
  XX_s4 <- t(X_u[X_s4,])%*%diag(N/n2,n2)%*%X_u[X_s4,]
  YY_s1 <- t(Y_u[Y_s1,])%*%diag(N/n1,n1)%*%Y_u[Y_s1,]
  YY_s2 <- t(Y_u[Y_s2,])%*%diag(N/n1,n1)%*%Y_u[Y_s2,]
  YY_s3 <- t(Y_u[Y_s3,])%*%diag(N/n2,n2)%*%Y_u[Y_s3,]
  YY_s4 <- t(Y_u[Y_s4,])%*%diag(N/n2,n2)%*%Y_u[Y_s4,]

  XX_s1_pd <- t(X_u[X_s1_pd,])%*%diag(1/pik1[X_s1_pd])%*%X_u[X_s1_pd,]
  XX_s2_pd <- t(X_u[X_s2_pd,])%*%diag(1/pik1[X_s2_pd])%*%X_u[X_s2_pd,]
  XX_s3_pd <- t(X_u[X_s3_pd,])%*%diag(1/pik2[X_s3_pd])%*%X_u[X_s3_pd,]
  XX_s4_pd <- t(X_u[X_s4_pd,])%*%diag(1/pik2[X_s4_pd])%*%X_u[X_s4_pd,]
  YY_s1_pd <- t(Y_u[Y_s1_pd,])%*%diag(1/pik3[Y_s1_pd])%*%Y_u[Y_s1_pd,]
  YY_s2_pd <- t(Y_u[Y_s2_pd,])%*%diag(1/pik3[Y_s2_pd])%*%Y_u[Y_s2_pd,]
  YY_s3_pd <- t(Y_u[Y_s3_pd,])%*%diag(1/pik4[Y_s3_pd])%*%Y_u[Y_s3_pd,]
  YY_s4_pd <- t(Y_u[Y_s4_pd,])%*%diag(1/pik4[Y_s4_pd])%*%Y_u[Y_s4_pd,]

  gl_s_X7_n1[i] <- sum(diag(solve(XX_s1 + A_lamda(lamda_u_X7, J))%*%XX_s1))
  gl_s_X11_n1[i] <- sum(diag(solve(XX_s2 + A_lamda(lamda_u_X11, J))%*%XX_s2))
  gl_s_X7_n2[i] <- sum(diag(solve(XX_s3 + A_lamda(lamda_u_X7, J))%*%XX_s3))
  gl_s_X11_n2[i] <- sum(diag(solve(XX_s4 + A_lamda(lamda_u_X11, J))%*%XX_s4))
  gl_s_Y7_n1[i] <- sum(diag(solve(YY_s1 + A_lamda(lamda_u_Y7, J))%*%YY_s1))
  gl_s_Y11_n1[i] <- sum(diag(solve(YY_s2 + A_lamda(lamda_u_Y11, J))%*%YY_s2))
  gl_s_Y7_n2[i] <- sum(diag(solve(YY_s3 + A_lamda(lamda_u_Y7, J))%*%YY_s3))
  gl_s_Y11_n2[i] <- sum(diag(solve(YY_s4 + A_lamda(lamda_u_Y11, J))%*%YY_s4))

  gl_s_X7_n1_pd[i] <- sum(diag(solve(XX_s1_pd + A_lamda(lamda_u_X7, J))%*%XX_s1_pd))
  gl_s_X11_n1_pd[i] <- sum(diag(solve(XX_s2_pd + A_lamda(lamda_u_X11, J))%*%XX_s2_pd))
  gl_s_X7_n2_pd[i] <- sum(diag(solve(XX_s3_pd + A_lamda(lamda_u_X7, J))%*%XX_s3_pd))
  gl_s_X11_n2_pd[i] <- sum(diag(solve(XX_s4_pd + A_lamda(lamda_u_X11, J))%*%XX_s4_pd))
  gl_s_Y7_n1_pd[i] <- sum(diag(solve(YY_s1_pd + A_lamda(lamda_u_Y7, J))%*%YY_s1_pd))
  gl_s_Y11_n1_pd[i] <- sum(diag(solve(YY_s2_pd + A_lamda(lamda_u_Y11, J))%*%YY_s2_pd))
  gl_s_Y7_n2_pd[i] <- sum(diag(solve(YY_s3_pd + A_lamda(lamda_u_Y7, J))%*%YY_s3_pd))
  gl_s_Y11_n2_pd[i] <- sum(diag(solve(YY_s4_pd + A_lamda(lamda_u_Y11, J))%*%YY_s4_pd))
}
```

```

f1_s_X7_n1 <- function(lamda)
{
  return(gl_7-sum(diag(solve(XX_s1 + A_lamda(lamda,J))%*%XX_s1)))
}
f1_s_X11_n1 <- function(lamda)
{
  return(gl_11-sum(diag(solve(XX_s2 + A_lamda(lamda,J))%*%XX_s2)))
}
f1_s_X7_n2 <- function(lamda)
{
  return(gl_7-sum(diag(solve(XX_s3 + A_lamda(lamda,J))%*%XX_s3)))
}
f1_s_X11_n2 <- function(lamda)
{
  return(gl_11-sum(diag(solve(XX_s4 + A_lamda(lamda,J))%*%XX_s4)))
}

f1_s_Y7_n1 <- function(lamda)
{
  return(log(gl_7)-log(sum(diag(solve(YY_s1 + A_lamda(lamda,J))%*%YY_s1))))
}
f1_s_Y11_n1 <- function(lamda)
{
  return(log(gl_11)-log(sum(diag(solve(YY_s2 + A_lamda(lamda,J))%*%YY_s2))))
}
f1_s_Y7_n2 <- function(lamda)
{
  return(log(gl_7)-log(sum(diag(solve(YY_s3 + A_lamda(lamda,J))%*%YY_s3))))
}
f1_s_Y11_n2 <- function(lamda)
{
  return(log(gl_11)-log(sum(diag(solve(YY_s4 + A_lamda(lamda,J))%*%YY_s4))))
}

#####

f1_s_X7_n1_pd <- function(lamda)
{
  return(gl_7-sum(diag(solve(XX_s1_pd + A_lamda(lamda,J))%*%XX_s1_pd)))
}
f1_s_X11_n1_pd <- function(lamda)
{
  return(gl_11-sum(diag(solve(XX_s2_pd + A_lamda(lamda,J))%*%XX_s2_pd)))
}
f1_s_X7_n2_pd <- function(lamda)
{
  return(gl_7-sum(diag(solve(XX_s3_pd + A_lamda(lamda,J))%*%XX_s3_pd)))
}
f1_s_X11_n2_pd <- function(lamda)
{
  return(gl_11-sum(diag(solve(XX_s4_pd + A_lamda(lamda,J))%*%XX_s4_pd)))
}

f1_s_Y7_n1_pd <- function(lamda)
{
  return(log(gl_7)-log(sum(diag(solve(YY_s1_pd + A_lamda(lamda,J))%*%YY_s1_pd))))
}
f1_s_Y11_n1_pd <- function(lamda)
{
  return(log(gl_11)-log(sum(diag(solve(YY_s2_pd + A_lamda(lamda,J))%*%YY_s2_pd))))
}
f1_s_Y7_n2_pd <- function(lamda)
{
  return(log(gl_7)-log(sum(diag(solve(YY_s3_pd + A_lamda(lamda,J))%*%YY_s3_pd))))
}
f1_s_Y11_n2_pd <- function(lamda)
{
  return(log(gl_11)-log(sum(diag(solve(YY_s4_pd + A_lamda(lamda,J))%*%YY_s4_pd))))
}

```

## Anexos

---

```
}

lam_s_X7_n1[i] <- uniroot(f1_s_X7_n1, c(1e-10,1), tol = 1e-10)$root
lam_s_X11_n1[i] <- uniroot(f1_s_X11_n1, c(1e-10,1), tol = 1e-10)$root
lam_s_X7_n2[i] <- uniroot(f1_s_X7_n2, c(1e-10,1), tol = 1e-10)$root
lam_s_X11_n2[i] <- uniroot(f1_s_X11_n2, c(1e-10,1), tol = 1e-10)$root

lam_s_Y7_n1[i] <- uniroot(f1_s_Y7_n1, c(1e-6,1), tol = 1e-10)$root
lam_s_Y11_n1[i] <- uniroot(f1_s_Y11_n1, c(1e-6,1), tol = 1e-10)$root
lam_s_Y7_n2[i] <- uniroot(f1_s_Y7_n2, c(1e-6,1), tol = 1e-10)$root
lam_s_Y11_n2[i] <- uniroot(f1_s_Y11_n2, c(1e-6,1), tol = 1e-10)$root

lam_s_X7_n1_pd[i] <- uniroot(f1_s_X7_n1_pd, c(1e-10,1), tol = 1e-10)$root
lam_s_X11_n1_pd[i] <- uniroot(f1_s_X11_n1_pd, c(1e-10,1), tol = 1e-10)$root
lam_s_X7_n2_pd[i] <- uniroot(f1_s_X7_n2_pd, c(1e-10,1), tol = 1e-10)$root
lam_s_X11_n2_pd[i] <- uniroot(f1_s_X11_n2_pd, c(1e-10,1), tol = 1e-10)$root

lam_s_Y7_n1_pd[i] <- uniroot(f1_s_Y7_n1_pd, c(1e-6,1), tol = 1e-10)$root
lam_s_Y11_n1_pd[i] <- uniroot(f1_s_Y11_n1_pd, c(1e-6,1), tol = 1e-10)$root
lam_s_Y7_n2_pd[i] <- uniroot(f1_s_Y7_n2_pd, c(1e-6,1), tol = 1e-10)$root
lam_s_Y11_n2_pd[i] <- uniroot(f1_s_Y11_n2_pd, c(1e-6,1), tol = 1e-10)$root

i <- i+1
}

#### Dr. Luis Fernando Contreras Cruz, lufecon@gmail.com ####

p <- 1 ## p es el grado del polinomio.
Pob_2000.2005 <- read.table("C:/Documents and Settings/Luis Fernando/Esritorio/ESTANCIA DEMAT GTO/Aplicación.txt")
X <- Pob_2000.2005$V1 # Población por Municipios en México 2000
Y <- Pob_2000.2005$V4 # Población por Municipios en México 2005
Z <- Pob_2000.2005$V2 # Viviendas por Municipios en México 2000
W <- Pob_2000.2005$V3 # Ing. Prom. Mensual por Municipios en México 2000
t.Y <- sum(Y)
N <- length(X)
#n1 <- 122
n2 <- 150
gl_7 <- 7
gl_11 <- 11
J1 <- min(N/4,35)

#####
## Cálculo de NODOS ##

Nodos_X <- quantile(X,probs = seq(1/(J1+1), J1/(J1+1), 1/(J1+1)))

#####
## Matriz de Nodos ##

Mat_NODOS_X <- matrix(rep(Nodos_X,N),N,J1,byrow=T)
Matriz_X <- matrix(rep(X,J1),N,J1,byrow=F)

#####
##### MATRIZ DISEÑO X #####

X_1 <- matrix(cbind(rep(1,N),X^p,(Matriz_X-Mat_NODOS_X)^p),N,)
X_u <- ifelse(X_1<0,X_1==0,X_1)

t.X <- apply(X_u,2,sum)
```

## Anexos

---

```
#####
#### Matriz de Suavisamiento ####

A_lamda <- function(lamda,J)
{
  diag(c(rep(0,p+1),rep(lamda,J)))
}

#####
### Población por Municipios en México Censo 2000 ###
#####

### Con l_u=7 y 11 encontrar lamda_u tal que  $g.l = \text{TRAZA}[(X^T X + A_{lamda})^{-1} X^T X] = l_u$ 

f1_X_7 <- function(lamda)
{
  XX <- t(X_u)%*%X_u
  return(gl_7-sum(diag(solve(XX + A_lamda(lamda,J1))%*%XX)))
}

f1_X_11 <- function(lamda)
{
  XX <- t(X_u)%*%X_u
  return(gl_11-sum(diag(solve(XX + A_lamda(lamda,J1))%*%XX)))
}

(lamda_u_X7 <- uniroot(f1_X_7, c(1000000,30000000000))$root)
21,866,888,904

(lamda_u_X11 <- uniroot(f1_X_11, c(1000000,30000000000))$root)
1,415,549,050

#####
# PROCESO DE SIMULACIÓN #
#####

#####
### Diseño de Muestreo con Probabilidades Desiguales ###
#####

#####
library(sampling)

UPME <- function (pik, eps = 1e-10)
{arr <-1
  n = sum(pik)
  pikt = pik
  #arr = 1
  while (arr > eps)
  {
    w = (pikt)/(1 - pikt)
    q = UPMEqfromw(w, n)
    pikt1 = pikt + pik - UPMEpikfromq(q)
    arr = sum(abs(pikt - pikt1))
    pikt = pikt1
  }
  pikt
  print(arr)
}
```

## Anexos

---

```
pik1 <- inclusionprobabilities(W,170)
pikt1 <- UPMEpiktildefrompik(pik1)
W1 <- pik1/(1-pikt1)
q1 <- UPMEqfromw(W1,160)

pik2 <- inclusionprobabilities(W,n2)
pikt2 <- UPMEpiktildefrompik(pik2)
W2 <- pik2/(1-pikt2)
q2 <- UPMEqfromw(W2,n2)
#####

M <- 5000 # Número de simulaciones

#####

gl_s_X7_n2_sr <- numeric(M)
lam_s_X7_n2_sr <- numeric(M)
gl_u_s_X7_n2_sr <- numeric(M)
t_s_X7_n2_sr <- numeric(M)
t_u_X7_n2_sr <- numeric(M)

gl_s_X11_n2_sr <- numeric(M)
lam_s_X11_n2_sr <- numeric(M)
gl_u_s_X11_n2_sr <- numeric(M)
t_s_X11_n2_sr <- numeric(M)
t_u_X11_n2_sr <- numeric(M)

gl_s_X7_n2_pd <- numeric(M)
lam_s_X7_n2_pd <- numeric(M)
gl_u_s_X7_n2_pd <- numeric(M)
t_s_X7_n2_pd <- numeric(M)
t_u_X7_n2_pd <- numeric(M)

gl_s_X11_n2_pd <- numeric(M)
lam_s_X11_n2_pd <- numeric(M)
gl_u_s_X11_n2_pd <- numeric(M)
t_s_X11_n2_pd <- numeric(M)
t_u_X11_n2_pd <- numeric(M)

##### Cuerpo principal #####

i <- 1
while(i <= M)
{
  cat("i = ",i,"\n")

  s1 <- sample(1:N,n2)
  s2 <- sample(1:N,n2)
  s3 <- seq(N)[UPMEsfromq(q2)==1]
  s4 <- seq(N)[UPMEsfromq(q2)==1]

  X_s1 <- t(X_u[s1,])%*%diag(N/n2,n2)%*%X_u[s1,]
  X_s2 <- t(X_u[s2,])%*%diag(N/n2,n2)%*%X_u[s2,]
  X_s3 <- t(X_u[s3,])%*%diag(1/pik2[s3])%*%X_u[s3,]
  X_s4 <- t(X_u[s4,])%*%diag(1/pik2[s4])%*%X_u[s4,]

  gl_s_X7_n2_sr[i] <- sum(diag(solve(X_s1 + A_lamda(lamda_u_X7,J1))%*%X_s1))
  gl_s_X11_n2_sr[i] <- sum(diag(solve(X_s2 + A_lamda(lamda_u_X11,J1))%*%X_s2))
  gl_s_X7_n2_pd[i] <- sum(diag(solve(X_s3 + A_lamda(lamda_u_X7,J1))%*%X_s3))
  gl_s_X11_n2_pd[i] <- sum(diag(solve(X_s4 + A_lamda(lamda_u_X11,J1))%*%X_s4))

  f1_s_X7_n2_sr <- function(lamda)
```

```

    {
      return(gl_7-sum(diag(solve(X_s1 + A_lamda(lamda,J1))%*%X_s1)))
    }

f1_s_X11_n2.sr <- function(lamda)
{
  return(gl_11-sum(diag(solve(X_s2 + A_lamda(lamda,J1))%*%X_s2)))
}

f1_s_X7_n2 <- function(lamda)
{
  return(gl_7-sum(diag(solve(X_s3 + A_lamda(lamda,J1))%*%X_s3)))
}

f1_s_X11_n2 <- function(lamda)
{
  return(gl_11-sum(diag(solve(X_s4 + A_lamda(lamda,J1))%*%X_s4)))
}

a <- try(uniroot(f1_s_X7_n2.sr, c(1000000,30000000000)))
b <- try(uniroot(f1_s_X11_n2.sr, c(1000000,30000000000)))
c <- try(uniroot(f1_s_X7_n2, c(1000000,30000000000)))
d <- try(uniroot(f1_s_X11_n2, c(1000000,30000000000)))

if(class(a)=="list" & class(b)=="list" & class(c)=="list" & class(d)=="list")
{

  lam_s_X7_n2_sr[i] <- a$root
  lam_s_X11_n2_sr[i] <- b$root
  lam_s_X7_n2_pd[i] <- c$root
  lam_s_X11_n2_pd[i] <- d$root

  XX <- t(X_u)%*%X_u

  #gl_u_s_X7_n2_sr[i] <- sum(diag(solve(XX + A_lamda(lam_s_X7_n2_sr[i],J1))%*%XX))
  #gl_u_s_X11_n2_sr[i] <- sum(diag(solve(XX + A_lamda(lam_s_X11_n2_sr[i],J1))%*%XX))
  #gl_u_s_X7_n2_pd[i] <- sum(diag(solve(XX + A_lamda(lam_s_X7_n2_pd[i],J1))%*%XX))
  #gl_u_s_X11_n2_pd[i] <- sum(diag(solve(XX + A_lamda(lam_s_X11_n2_pd[i],J1))%*%XX))

  t_s_X7_n2_sr[i] <- t(t.X)%*%solve(X_s1 + A_lamda(lam_s_X7_n2_sr[i],J1))%*%t(X_u[s1,])%*%
    diag(N/n2,n2)%*%Y[s1]
  t_s_X11_n2_sr[i] <- t(t.X)%*%solve(X_s2 + A_lamda(lam_s_X11_n2_sr[i],J1))%*%t(X_u[s2,])%*%
    diag(N/n2,n2)%*%Y[s2]
  t_s_X7_n2_pd[i] <- t(t.X)%*%solve(X_s3 + A_lamda(lam_s_X7_n2_pd[i],J1))%*%t(X_u[s3,])%*%
    diag(1/pik2[s3])%*%Y[s3]
  t_s_X11_n2_pd[i] <- t(t.X)%*%solve(X_s4 + A_lamda(lam_s_X11_n2_pd[i],J1))%*%t(X_u[s4,])%*%
    diag(1/pik2[s4])%*%Y[s4]

  t_u_X7_n2_sr[i] <- t(t.X)%*%solve(X_s1 + A_lamda(lamda_u_X7,J1))%*%t(X_u[s1,])%*%
    diag(N/n2,n2)%*%Y[s1]
  t_u_X11_n2_sr[i] <- t(t.X)%*%solve(X_s2 + A_lamda(lamda_u_X11,J1))%*%t(X_u[s2,])%*%
    diag(N/n2,n2)%*%Y[s2]
  t_u_X7_n2_pd[i] <- t(t.X)%*%solve(X_s3 + A_lamda(lamda_u_X7,J1))%*%t(X_u[s3,])%*%
    diag(1/pik2[s3])%*%Y[s3]
  t_u_X11_n2_pd[i] <- t(t.X)%*%solve(X_s4 + A_lamda(lamda_u_X11,J1))%*%t(X_u[s4,])%*%
    diag(1/pik2[s4])%*%Y[s4]

  i <- i+1
}
}

```



## Anexo C: Códigos de Programas en R usados en el Capítulo 6

```
#### Dr. Luis Fernando Contreras Cruz, lufecon@gmail.com ####
```

```
#####  
### 3D ###  
#####
```

```
library(rgl)  
  
N <- 1000  
  
m1 <- function(x,y)  
{  
  5 + sin(2*pi*x)+ cos(2*pi*y)  
}  
  
m2 <- function(x,y)  
{  
  5 + cos(2*pi*abs(x-y)) - sin(2*pi*abs(x+y))  
}  
  
m3 <- function(x,y)  
{  
  5 + y*sin(2*pi*x) + cos(2*pi*x)  
}  
  
x <- y <- seq(0,1,length=N)  
  
z1 <- outer(x,y,"m1")  
z2 <- outer(x,y,"m2")  
z3 <- outer(x,y,"m3")  
  
persp3d(x,y,z1,col = "lightblue")  
persp3d(x,y,z2,col = "lightblue")  
persp3d(x,y,z3,col = "lightblue")
```

```
#####
```

```
N <- 1000  
X.1.2 <- matrix(0,N,2)  
  
for(i in 1:N) { X.1.2[i,]<-runif(2) }  
  
m1 <- function(X)  
{  
  X.1 <- X[,1]  
  X.2 <- X[,2]  
  
  m1 <- 5 + sin(2*pi*X.1)+ cos(2*pi*X.2)  
  return(m1)  
}  
  
m2 <- function(X)  
{  
  X.1 <- X[,1]  
  X.2 <- X[,2]  
  
  m2 <- 5 + cos(2*pi*abs(X.1-X.2)) - sin(2*pi*abs(X.1+X.2))  
  return(m2)  
}  
  
m3 <- function(X)
```

## Anexos

---

```
{
  X.1 <- X[,1]
  X.2 <- X[,2]

  m3 <- 5 + X.2*sin(2*pi*X.1) + cos(2*pi*X.1)
  return(m3)
}

Sig.1 <- 0.4
Sig.2 <- 1.0

X.1 <- X.1.2[,1]
X.2 <- X.1.2[,2]

y.1.1 <- m1(X.1.2) + Sig.1*rnorm(N)
y.2.1 <- m2(X.1.2) + Sig.1*rnorm(N)
y.3.1 <- m3(X.1.2) + Sig.1*rnorm(N)

y.1.2 <- m1(X.1.2) + Sig.2*rnorm(N)
y.2.2 <- m2(X.1.2) + Sig.2*rnorm(N)
y.3.2 <- m3(X.1.2) + Sig.2*rnorm(N)

t.1.1 <- sum(y.1.1)
t.2.1 <- sum(y.2.1)
t.3.1 <- sum(y.3.1)

t.1.2 <- sum(y.1.2)
t.2.2 <- sum(y.2.2)
t.3.2 <- sum(y.3.2)

#####
# VARYING COEFFICIENT MODELS #
#####

K <- 25
knots <- quantile(X.1,probs = seq(1/(K+1), K/(K+1), 1/(K+1)))

X_U <- cbind(rep(1,N),X.1,X.2,X.1*X.2)
X_U.LIN <- cbind(X_U,X.1^2,X.2^2,X.1^3,X.2^3,X.1^2*X.2,X.1*X.2^2)

Z_U <- outer(X.1,knots,"-")
Z_U <- Z_U*(Z_U>0)
C_U <- cbind(X_U,Z_U,X.2*Z_U)
dim(C_U)

Delta <- function(lambda1,lambda2)
{
  D <- diag(c(rep(0,ncol(X_U)), rep(lambda1,K), rep(lambda2,K)))
  return(D)
}

df <- sum(diag(solve(t(C_U)%*%C_U+ Delta(0,0))%*%t(C_U)%*%C_U))
df

D1 <- function(lambda1)
{
  D <- diag(c(0,0,rep(lambda1,K)))
  return(D)
}

D2 <- function(lambda2)
{
  D <- diag(c(0,0,0,rep(lambda2,K)))
  return(D)
}
```

## Anexos

---

```
C.1 <- C_U[,c(1,2,5:29)]
C.2 <- C_U[,c(1,3,4,30:54)]

adf.1 <- sum(diag(solve(t(C.1)%*%C.1+ D1(0))%*%t(C.1)%*%C.1))-1
adf.2 <- sum(diag(solve(t(C.2)%*%C.2+ D2(0))%*%t(C.2)%*%C.2))-1
adf.1
adf.2
adf.1 + adf.2    ## adf.1 + adf.2 + 1 = df ? ##

f1 <- function(l1)
{
  adf1<- 3
  return(adf1-(sum(diag(solve(t(C.1)%*%C.1+ D1(l1))%*%t(C.1)%*%C.1))-1))
}

f2 <- function(l2)
{
  adf2<- 3
  return(adf2-(sum(diag(solve(t(C.2)%*%C.2+ D2(l2))%*%t(C.2)%*%C.2))-1))
}

(lamda_1 <- uniroot(f1, c(0,1000))$root)
(lamda_2 <- uniroot(f2, c(0,1000))$root)

lambda.1 <- c(5.030301,1.590498,0.6462465,0.3093151,0.1661367,0.09703428,0.06030992,0.03932021,0.02662745,0.01856069)
lambda.2 <- c(8.547362,1.701828,0.5367981,0.2174824,0.1038409,0.05573747,0.03258389,0.02027538,0.01323529,0.008976242)

df <- sum(diag(solve(t(C_U)%*%C_U+ Delta(lambda.1[10],lambda.2[10]))%*%t(C_U)%*%C_U))
df

d1 <- rep(lambda.1,length(lambda.1))
d2 <- rep(lambda.2,rep(length(lambda.2),length(lambda.2)))

d <- matrix(cbind(d1,d2),,2)

H <- nrow(d)

#g <- apply(d,1,sum)
#which(g==g[10])

#####
# ADDITIVE MODELS #
#####

K <- 25
knots1 <- quantile(X.1,probs = seq(1/(K+1), K/(K+1), 1/(K+1)))
knots2 <- quantile(X.2,probs = seq(1/(K+1), K/(K+1), 1/(K+1)))

X2_U <- cbind(rep(1,N),X.1,X.2)
Z2_U <- outer(X.1,knots1,"-")
Z3_U <- outer(X.2,knots2,"-")
Z2_U <- Z2_U*(Z2_U>0)
Z3_U <- Z3_U*(Z3_U>0)

C2_U <- cbind(X2_U,Z2_U,Z3_U)
dim(C2_U)

Delta2 <- function(lambda1,lambda2)
{
  D <- diag(c(rep(0,ncol(X2_U)), rep(lambda1,K), rep(lambda2,K)))
```

## Anexos

---

```
return(D)
}

df2 <- sum(diag(solve(t(C2_U)%*%C2_U+ Delta2(0,0))%*%t(C2_U)%*%C2_U))
df2

D1.2 <- function(lambda1)
{
D <- diag(c(0,0,rep(lambda1,K)))
return(D)
}

D2.2 <- function(lambda2)
{
D <- diag(c(0,0,rep(lambda2,K)))
return(D)
}

C.1.2 <- C2_U[,c(1,2,4:28)]
C.2.2 <- C2_U[,c(1,3,29:53)]

adf.1.2 <- sum(diag(solve(t(C.1.2)%*%C.1.2+ D1.2(0))%*%t(C.1.2)%*%C.1.2))-1
adf.2.2 <- sum(diag(solve(t(C.2.2)%*%C.2.2+ D2.2(0))%*%t(C.2.2)%*%C.2.2))-1

adf.1.2
adf.2.2

adf.1.2 + adf.2.2    ## adf.1.2 + adf.2.2 + 1 = df2 ? ##

f1.2 <- function(l1)
{
adf1<- 12
return(adf1-(sum(diag(solve(t(C.1.2)%*%C.1.2+ D1.2(l1))%*%t(C.1.2)%*%C.1.2))-1))
}

f2.2 <- function(l2)
{
adf2<- 12
return(adf2-(sum(diag(solve(t(C.2.2)%*%C.2.2+ D2.2(l2))%*%t(C.2.2)%*%C.2.2))-1))
}

(lamda_1.2 <- uniroot(f1.2, c(0,1000))$root)
(lamda_2.2 <- uniroot(f2.2, c(0,1000))$root)

lambda.1.2 <- c(5.030301,1.590498,0.6462465,0.3093151,0.1661367,0.09703428,0.06030992,0.03932021,0.02662745,0.01856069)
lambda.2.2 <- c(4.962529,1.583656,0.648448,0.3115974,0.1677288,0.09807571,0.06104274,0.03984619,0.02699124,0.01881359)

df2 <- sum(diag(solve(t(C2_U)%*%C2_U+ Delta2(lambda.1.2[10],lambda.2.2[10]))%*%t(C2_U)%*%C2_U))
df2

d1.2 <- rep(lambda.1.2,length(lambda.1.2))
d2.2 <- rep(lambda.2.2,rep(length(lambda.2.2),length(lambda.2.2)))

d2 <- matrix(cbind(d1.2,d2.2),,2)

H2 <- nrow(d2)

#g2 <- apply(d2,1,sum)
#which(g2==g2[25])

#####
## Design SI ##
#####
```

## Anexos

---

```
n1 <- 100
n2 <- 200

N.1 <- N^2*(1/n1-1/N)
N.2 <- N^2*(1/n2-1/N)

Pi.inv.n1 <- diag(rep(N/n1,n1))
Pi.inv.n2 <- diag(rep(N/n2,n2))

M <- 2000

V1 <- numeric(0)
V2 <- numeric(0)
V3 <- numeric(0)
V4 <- numeric(0)
V5 <- numeric(0)
V6 <- numeric(0)
V7 <- numeric(0)
V8 <- numeric(0)
V9 <- numeric(0)
V10 <- numeric(0)
V11 <- numeric(0)
V12 <- numeric(0)

V13 <- numeric(0)
V14 <- numeric(0)
V15 <- numeric(0)
V16 <- numeric(0)
V17 <- numeric(0)
V18 <- numeric(0)
V19 <- numeric(0)
V20 <- numeric(0)
V21 <- numeric(0)
V22 <- numeric(0)
V23 <- numeric(0)
V24 <- numeric(0)

L1L2.y1.n1.1 <- matrix(0,M,2)
L1L2.y1.n2.1 <- matrix(0,M,2)
L1L2.y2.n1.1 <- matrix(0,M,2)
L1L2.y2.n2.1 <- matrix(0,M,2)
L1L2.y3.n1.1 <- matrix(0,M,2)
L1L2.y3.n2.1 <- matrix(0,M,2)
L1L2.y1.n1.2 <- matrix(0,M,2)
L1L2.y1.n2.2 <- matrix(0,M,2)
L1L2.y2.n1.2 <- matrix(0,M,2)
L1L2.y2.n2.2 <- matrix(0,M,2)
L1L2.y3.n1.2 <- matrix(0,M,2)
L1L2.y3.n2.2 <- matrix(0,M,2)

L1L2.y1.n1.1.ADI <- matrix(0,M,2)
L1L2.y1.n2.1.ADI <- matrix(0,M,2)
L1L2.y2.n1.1.ADI <- matrix(0,M,2)
L1L2.y2.n2.1.ADI <- matrix(0,M,2)
L1L2.y3.n1.1.ADI <- matrix(0,M,2)
L1L2.y3.n2.1.ADI <- matrix(0,M,2)
L1L2.y1.n1.2.ADI <- matrix(0,M,2)
L1L2.y1.n2.2.ADI <- matrix(0,M,2)
L1L2.y2.n1.2.ADI <- matrix(0,M,2)
L1L2.y2.n2.2.ADI <- matrix(0,M,2)
L1L2.y3.n1.2.ADI <- matrix(0,M,2)
L1L2.y3.n2.2.ADI <- matrix(0,M,2)

df.y1.n1.1 <- numeric(0)
df.y1.n2.1 <- numeric(0)
df.y2.n1.1 <- numeric(0)
df.y2.n2.1 <- numeric(0)
```

## Anexos

---

```
df.y3.n1.1 <- numeric(0)
df.y3.n2.1 <- numeric(0)
```

```
df.y1.n1.2 <- numeric(0)
df.y1.n2.2 <- numeric(0)
df.y2.n1.2 <- numeric(0)
df.y2.n2.2 <- numeric(0)
df.y3.n1.2 <- numeric(0)
df.y3.n2.2 <- numeric(0)
```

```
df.y1.n1.1.ADI <- numeric(0)
df.y1.n2.1.ADI <- numeric(0)
df.y2.n1.1.ADI <- numeric(0)
df.y2.n2.1.ADI <- numeric(0)
df.y3.n1.1.ADI <- numeric(0)
df.y3.n2.1.ADI <- numeric(0)
```

```
df.y1.n1.2.ADI <- numeric(0)
df.y1.n2.2.ADI <- numeric(0)
df.y2.n1.2.ADI <- numeric(0)
df.y2.n2.2.ADI <- numeric(0)
df.y3.n1.2.ADI <- numeric(0)
df.y3.n2.2.ADI <- numeric(0)
```

```
V1.hat <- numeric(0)
V2.hat <- numeric(0)
V3.hat <- numeric(0)
V4.hat <- numeric(0)
V5.hat <- numeric(0)
V6.hat <- numeric(0)
V7.hat <- numeric(0)
V8.hat <- numeric(0)
V9.hat <- numeric(0)
V10.hat <- numeric(0)
V11.hat <- numeric(0)
V12.hat <- numeric(0)
V13.hat <- numeric(0)
V14.hat <- numeric(0)
V15.hat <- numeric(0)
V16.hat <- numeric(0)
V17.hat <- numeric(0)
V18.hat <- numeric(0)
V19.hat <- numeric(0)
V20.hat <- numeric(0)
V21.hat <- numeric(0)
V22.hat <- numeric(0)
V23.hat <- numeric(0)
V24.hat <- numeric(0)
V25.hat <- numeric(0)
V26.hat <- numeric(0)
V27.hat <- numeric(0)
V28.hat <- numeric(0)
V29.hat <- numeric(0)
V30.hat <- numeric(0)
V31.hat <- numeric(0)
V32.hat <- numeric(0)
V33.hat <- numeric(0)
V34.hat <- numeric(0)
V35.hat <- numeric(0)
V36.hat <- numeric(0)
V37.hat <- numeric(0)
V38.hat <- numeric(0)
V39.hat <- numeric(0)
V40.hat <- numeric(0)
V41.hat <- numeric(0)
V42.hat <- numeric(0)
V43.hat <- numeric(0)
V44.hat <- numeric(0)
```

## Anexos

---

```
V45.hat <- numeric(0)
V46.hat <- numeric(0)
V47.hat <- numeric(0)
V48.hat <- numeric(0)
V49.hat <- numeric(0)
V50.hat <- numeric(0)
V51.hat <- numeric(0)
V52.hat <- numeric(0)
V53.hat <- numeric(0)
V54.hat <- numeric(0)
V55.hat <- numeric(0)
V56.hat <- numeric(0)
V57.hat <- numeric(0)
V58.hat <- numeric(0)
V59.hat <- numeric(0)
V60.hat <- numeric(0)
V61.hat <- numeric(0)
V62.hat <- numeric(0)
V63.hat <- numeric(0)
V64.hat <- numeric(0)
V65.hat <- numeric(0)
V66.hat <- numeric(0)
V67.hat <- numeric(0)
V68.hat <- numeric(0)
V69.hat <- numeric(0)
V70.hat <- numeric(0)
V71.hat <- numeric(0)
V72.hat <- numeric(0)
V73.hat <- numeric(0)
V74.hat <- numeric(0)
V75.hat <- numeric(0)
V76.hat <- numeric(0)
V77.hat <- numeric(0)
V78.hat <- numeric(0)
V79.hat <- numeric(0)
V80.hat <- numeric(0)
V81.hat <- numeric(0)
V82.hat <- numeric(0)
V83.hat <- numeric(0)
V84.hat <- numeric(0)
V85.hat <- numeric(0)
V86.hat <- numeric(0)
V87.hat <- numeric(0)
V88.hat <- numeric(0)
V89.hat <- numeric(0)
V90.hat <- numeric(0)
V91.hat <- numeric(0)
V92.hat <- numeric(0)
V93.hat <- numeric(0)
V94.hat <- numeric(0)
V95.hat <- numeric(0)
V96.hat <- numeric(0)
V97.hat <- numeric(0)
V98.hat <- numeric(0)
V99.hat <- numeric(0)
V100.hat <- numeric(0)
V101.hat <- numeric(0)
V102.hat <- numeric(0)
V103.hat <- numeric(0)
V104.hat <- numeric(0)
V105.hat <- numeric(0)
V106.hat <- numeric(0)
V107.hat <- numeric(0)
V108.hat <- numeric(0)
V109.hat <- numeric(0)
V110.hat <- numeric(0)
V111.hat <- numeric(0)
V112.hat <- numeric(0)
```

## Anexos

---

```
V113.hat <- numeric(0)
V114.hat <- numeric(0)
V115.hat <- numeric(0)
V116.hat <- numeric(0)
V117.hat <- numeric(0)
V118.hat <- numeric(0)
V119.hat <- numeric(0)
V120.hat <- numeric(0)

t_hat.y1.n1.1 <- numeric(0)
t_hat.y1.n2.1 <- numeric(0)
t_hat.y2.n1.1 <- numeric(0)
t_hat.y2.n2.1 <- numeric(0)
t_hat.y3.n1.1 <- numeric(0)
t_hat.y3.n2.1 <- numeric(0)
t_hat.y1.n1.2 <- numeric(0)
t_hat.y1.n2.2 <- numeric(0)
t_hat.y2.n1.2 <- numeric(0)
t_hat.y2.n2.2 <- numeric(0)
t_hat.y3.n1.2 <- numeric(0)
t_hat.y3.n2.2 <- numeric(0)

t_hat.y1.n1.LIN.1 <- numeric(0)
t_hat.y1.n2.LIN.1 <- numeric(0)
t_hat.y2.n1.LIN.1 <- numeric(0)
t_hat.y2.n2.LIN.1 <- numeric(0)
t_hat.y3.n1.LIN.1 <- numeric(0)
t_hat.y3.n2.LIN.1 <- numeric(0)
t_hat.y1.n1.LIN.2 <- numeric(0)
t_hat.y1.n2.LIN.2 <- numeric(0)
t_hat.y2.n1.LIN.2 <- numeric(0)
t_hat.y2.n2.LIN.2 <- numeric(0)
t_hat.y3.n1.LIN.2 <- numeric(0)
t_hat.y3.n2.LIN.2 <- numeric(0)

t_hat.y1.n1.1.ADI <- numeric(0)
t_hat.y1.n2.1.ADI <- numeric(0)
t_hat.y2.n1.1.ADI <- numeric(0)
t_hat.y2.n2.1.ADI <- numeric(0)
t_hat.y3.n1.1.ADI <- numeric(0)
t_hat.y3.n2.1.ADI <- numeric(0)
t_hat.y1.n1.2.ADI <- numeric(0)
t_hat.y1.n2.2.ADI <- numeric(0)
t_hat.y2.n1.2.ADI <- numeric(0)
t_hat.y2.n2.2.ADI <- numeric(0)
t_hat.y3.n1.2.ADI <- numeric(0)
t_hat.y3.n2.2.ADI <- numeric(0)

Coverage.y1.n1.1<- numeric(0)
Coverage.y1.n2.1<- numeric(0)
Coverage.y2.n1.1<- numeric(0)
Coverage.y2.n2.1<- numeric(0)
Coverage.y3.n1.1<- numeric(0)
Coverage.y3.n2.1<- numeric(0)

Coverage.y1.n1.2<- numeric(0)
Coverage.y1.n2.2<- numeric(0)
Coverage.y2.n1.2<- numeric(0)
Coverage.y2.n2.2<- numeric(0)
Coverage.y3.n1.2<- numeric(0)
Coverage.y3.n2.2<- numeric(0)

Coverage.y1.n1.1.g_W<- numeric(0)
Coverage.y1.n2.1.g_W<- numeric(0)
Coverage.y2.n1.1.g_W<- numeric(0)
Coverage.y2.n2.1.g_W<- numeric(0)
Coverage.y3.n1.1.g_W<- numeric(0)
```



## Anexos

---

```
Coverage.y3.n2.1.g_W<- numeric(0)

Coverage.y1.n1.2.g_W<- numeric(0)
Coverage.y1.n2.2.g_W<- numeric(0)
Coverage.y2.n1.2.g_W<- numeric(0)
Coverage.y2.n2.2.g_W<- numeric(0)
Coverage.y3.n1.2.g_W<- numeric(0)
Coverage.y3.n2.2.g_W<- numeric(0)

Coverage.y1.n1.1.CV<- numeric(0)
Coverage.y1.n2.1.CV<- numeric(0)
Coverage.y2.n1.1.CV<- numeric(0)
Coverage.y2.n2.1.CV<- numeric(0)
Coverage.y3.n1.1.CV<- numeric(0)
Coverage.y3.n2.1.CV<- numeric(0)

Coverage.y1.n1.2.CV<- numeric(0)
Coverage.y1.n2.2.CV<- numeric(0)
Coverage.y2.n1.2.CV<- numeric(0)
Coverage.y2.n2.2.CV<- numeric(0)
Coverage.y3.n1.2.CV<- numeric(0)
Coverage.y3.n2.2.CV<- numeric(0)

Coverage.y1.n1.1.CV.g_W<- numeric(0)
Coverage.y1.n2.1.CV.g_W<- numeric(0)
Coverage.y2.n1.1.CV.g_W<- numeric(0)
Coverage.y2.n2.1.CV.g_W<- numeric(0)
Coverage.y3.n1.1.CV.g_W<- numeric(0)
Coverage.y3.n2.1.CV.g_W<- numeric(0)

Coverage.y1.n1.2.CV.g_W<- numeric(0)
Coverage.y1.n2.2.CV.g_W<- numeric(0)
Coverage.y2.n1.2.CV.g_W<- numeric(0)
Coverage.y2.n2.2.CV.g_W<- numeric(0)
Coverage.y3.n1.2.CV.g_W<- numeric(0)
Coverage.y3.n2.2.CV.g_W<- numeric(0)

Coverage.y1.n1.1.ADI <- numeric(0)
Coverage.y1.n2.1.ADI <- numeric(0)
Coverage.y2.n1.1.ADI <- numeric(0)
Coverage.y2.n2.1.ADI <- numeric(0)
Coverage.y3.n1.1.ADI <- numeric(0)
Coverage.y3.n2.1.ADI <- numeric(0)

Coverage.y1.n1.2.ADI <- numeric(0)
Coverage.y1.n2.2.ADI <- numeric(0)
Coverage.y2.n1.2.ADI <- numeric(0)
Coverage.y2.n2.2.ADI <- numeric(0)
Coverage.y3.n1.2.ADI <- numeric(0)
Coverage.y3.n2.2.ADI <- numeric(0)

Coverage.y1.n1.1.g_W.ADI <- numeric(0)
Coverage.y1.n2.1.g_W.ADI <- numeric(0)
Coverage.y2.n1.1.g_W.ADI <- numeric(0)
Coverage.y2.n2.1.g_W.ADI <- numeric(0)
Coverage.y3.n1.1.g_W.ADI <- numeric(0)
Coverage.y3.n2.1.g_W.ADI <- numeric(0)

Coverage.y1.n1.2.g_W.ADI <- numeric(0)
Coverage.y1.n2.2.g_W.ADI <- numeric(0)
Coverage.y2.n1.2.g_W.ADI <- numeric(0)
Coverage.y2.n2.2.g_W.ADI <- numeric(0)
Coverage.y3.n1.2.g_W.ADI <- numeric(0)
Coverage.y3.n2.2.g_W.ADI <- numeric(0)

Coverage.y1.n1.1.CV.ADI <- numeric(0)
Coverage.y1.n2.1.CV.ADI <- numeric(0)
Coverage.y2.n1.1.CV.ADI <- numeric(0)
```

## Anexos

---

```
Coverage.y2.n2.1.CV.ADI <- numeric(0)
Coverage.y3.n1.1.CV.ADI <- numeric(0)
Coverage.y3.n2.1.CV.ADI <- numeric(0)

Coverage.y1.n1.2.CV.ADI <- numeric(0)
Coverage.y1.n2.2.CV.ADI <- numeric(0)
Coverage.y2.n1.2.CV.ADI <- numeric(0)
Coverage.y2.n2.2.CV.ADI <- numeric(0)
Coverage.y3.n1.2.CV.ADI <- numeric(0)
Coverage.y3.n2.2.CV.ADI <- numeric(0)

Coverage.y1.n1.1.CV.g_W.ADI <- numeric(0)
Coverage.y1.n2.1.CV.g_W.ADI <- numeric(0)
Coverage.y2.n1.1.CV.g_W.ADI <- numeric(0)
Coverage.y2.n2.1.CV.g_W.ADI <- numeric(0)
Coverage.y3.n1.1.CV.g_W.ADI <- numeric(0)
Coverage.y3.n2.1.CV.g_W.ADI <- numeric(0)

Coverage.y1.n1.2.CV.g_W.ADI <- numeric(0)
Coverage.y1.n2.2.CV.g_W.ADI <- numeric(0)
Coverage.y2.n1.2.CV.g_W.ADI <- numeric(0)
Coverage.y2.n2.2.CV.g_W.ADI <- numeric(0)
Coverage.y3.n1.2.CV.g_W.ADI <- numeric(0)
Coverage.y3.n2.2.CV.g_W.ADI <- numeric(0)

Coverage.y1.n1.1.LIN<- numeric(0)
Coverage.y1.n2.1.LIN<- numeric(0)
Coverage.y2.n1.1.LIN<- numeric(0)
Coverage.y2.n2.1.LIN<- numeric(0)
Coverage.y3.n1.1.LIN<- numeric(0)
Coverage.y3.n2.1.LIN<- numeric(0)

Coverage.y1.n1.2.LIN<- numeric(0)
Coverage.y1.n2.2.LIN<- numeric(0)
Coverage.y2.n1.2.LIN<- numeric(0)
Coverage.y2.n2.2.LIN<- numeric(0)
Coverage.y3.n1.2.LIN<- numeric(0)
Coverage.y3.n2.2.LIN<- numeric(0)

Coverage.y1.n1.1.g_W.LIN<- numeric(0)
Coverage.y1.n2.1.g_W.LIN<- numeric(0)
Coverage.y2.n1.1.g_W.LIN<- numeric(0)
Coverage.y2.n2.1.g_W.LIN<- numeric(0)
Coverage.y3.n1.1.g_W.LIN<- numeric(0)
Coverage.y3.n2.1.g_W.LIN<- numeric(0)

Coverage.y1.n1.2.g_W.LIN<- numeric(0)
Coverage.y1.n2.2.g_W.LIN<- numeric(0)
Coverage.y2.n1.2.g_W.LIN<- numeric(0)
Coverage.y2.n2.2.g_W.LIN<- numeric(0)
Coverage.y3.n1.2.g_W.LIN<- numeric(0)
Coverage.y3.n2.2.g_W.LIN<- numeric(0)

der.y1.n1.1 <- numeric(0)
izq.y1.n1.1 <- numeric(0)
der.y1.n2.1 <- numeric(0)
izq.y1.n2.1 <- numeric(0)
der.y2.n1.1 <- numeric(0)
izq.y2.n1.1 <- numeric(0)
der.y2.n2.1 <- numeric(0)
izq.y2.n2.1 <- numeric(0)
der.y3.n1.1 <- numeric(0)
izq.y3.n1.1 <- numeric(0)
der.y3.n2.1 <- numeric(0)
izq.y3.n2.1 <- numeric(0)

der.y1.n1.2 <- numeric(0)
izq.y1.n1.2 <- numeric(0)
```

```
der.y1.n2.2 <- numeric(0)
izq.y1.n2.2 <- numeric(0)
der.y2.n1.2 <- numeric(0)
izq.y2.n1.2 <- numeric(0)
der.y2.n2.2 <- numeric(0)
izq.y2.n2.2 <- numeric(0)
der.y3.n1.2 <- numeric(0)
izq.y3.n1.2 <- numeric(0)
der.y3.n2.2 <- numeric(0)
izq.y3.n2.2 <- numeric(0)

der.y1.n1.1.CV <- numeric(0)
izq.y1.n1.1.CV <- numeric(0)
der.y1.n2.1.CV <- numeric(0)
izq.y1.n2.1.CV <- numeric(0)
der.y2.n1.1.CV <- numeric(0)
izq.y2.n1.1.CV <- numeric(0)
der.y2.n2.1.CV <- numeric(0)
izq.y2.n2.1.CV <- numeric(0)
der.y3.n1.1.CV <- numeric(0)
izq.y3.n1.1.CV <- numeric(0)
der.y3.n2.1.CV <- numeric(0)
izq.y3.n2.1.CV <- numeric(0)

der.y1.n1.2.CV <- numeric(0)
izq.y1.n1.2.CV <- numeric(0)
der.y1.n2.2.CV <- numeric(0)
izq.y1.n2.2.CV <- numeric(0)
der.y2.n1.2.CV <- numeric(0)
izq.y2.n1.2.CV <- numeric(0)
der.y2.n2.2.CV <- numeric(0)
izq.y2.n2.2.CV <- numeric(0)
der.y3.n1.2.CV <- numeric(0)
izq.y3.n1.2.CV <- numeric(0)
der.y3.n2.2.CV <- numeric(0)
izq.y3.n2.2.CV <- numeric(0)

der.y1.n1.1.CV.g_W <- numeric(0)
izq.y1.n1.1.CV.g_W <- numeric(0)
der.y1.n2.1.CV.g_W <- numeric(0)
izq.y1.n2.1.CV.g_W <- numeric(0)
der.y2.n1.1.CV.g_W <- numeric(0)
izq.y2.n1.1.CV.g_W <- numeric(0)
der.y2.n2.1.CV.g_W <- numeric(0)
izq.y2.n2.1.CV.g_W <- numeric(0)
der.y3.n1.1.CV.g_W <- numeric(0)
izq.y3.n1.1.CV.g_W <- numeric(0)
der.y3.n2.1.CV.g_W <- numeric(0)
izq.y3.n2.1.CV.g_W <- numeric(0)

der.y1.n1.2.CV.g_W <- numeric(0)
izq.y1.n1.2.CV.g_W <- numeric(0)
der.y1.n2.2.CV.g_W <- numeric(0)
izq.y1.n2.2.CV.g_W <- numeric(0)
der.y2.n1.2.CV.g_W <- numeric(0)
izq.y2.n1.2.CV.g_W <- numeric(0)
der.y2.n2.2.CV.g_W <- numeric(0)
izq.y2.n2.2.CV.g_W <- numeric(0)
der.y3.n1.2.CV.g_W <- numeric(0)
izq.y3.n1.2.CV.g_W <- numeric(0)
der.y3.n2.2.CV.g_W <- numeric(0)
izq.y3.n2.2.CV.g_W <- numeric(0)

der.y1.n1.1.g_W <- numeric(0)
izq.y1.n1.1.g_W <- numeric(0)
der.y1.n2.1.g_W <- numeric(0)
izq.y1.n2.1.g_W <- numeric(0)
der.y2.n1.1.g_W <- numeric(0)
```

## Anexos

---

```
izq.y2.n1.1.g_W <- numeric(0)
der.y2.n2.1.g_W <- numeric(0)
izq.y2.n2.1.g_W <- numeric(0)
der.y3.n1.1.g_W <- numeric(0)
izq.y3.n1.1.g_W <- numeric(0)
der.y3.n2.1.g_W <- numeric(0)
izq.y3.n2.1.g_W <- numeric(0)
```

```
der.y1.n1.2.g_W <- numeric(0)
izq.y1.n1.2.g_W <- numeric(0)
der.y1.n2.2.g_W <- numeric(0)
izq.y1.n2.2.g_W <- numeric(0)
der.y2.n1.2.g_W <- numeric(0)
izq.y2.n1.2.g_W <- numeric(0)
der.y2.n2.2.g_W <- numeric(0)
izq.y2.n2.2.g_W <- numeric(0)
der.y3.n1.2.g_W <- numeric(0)
izq.y3.n1.2.g_W <- numeric(0)
der.y3.n2.2.g_W <- numeric(0)
izq.y3.n2.2.g_W <- numeric(0)
```

```
der.y1.n1.1.ADI <- numeric(0)
izq.y1.n1.1.ADI <- numeric(0)
der.y1.n2.1.ADI <- numeric(0)
izq.y1.n2.1.ADI <- numeric(0)
der.y2.n1.1.ADI <- numeric(0)
izq.y2.n1.1.ADI <- numeric(0)
der.y2.n2.1.ADI <- numeric(0)
izq.y2.n2.1.ADI <- numeric(0)
der.y3.n1.1.ADI <- numeric(0)
izq.y3.n1.1.ADI <- numeric(0)
der.y3.n2.1.ADI <- numeric(0)
izq.y3.n2.1.ADI <- numeric(0)
```

```
der.y1.n1.2.ADI <- numeric(0)
izq.y1.n1.2.ADI <- numeric(0)
der.y1.n2.2.ADI <- numeric(0)
izq.y1.n2.2.ADI <- numeric(0)
der.y2.n1.2.ADI <- numeric(0)
izq.y2.n1.2.ADI <- numeric(0)
der.y2.n2.2.ADI <- numeric(0)
izq.y2.n2.2.ADI <- numeric(0)
der.y3.n1.2.ADI <- numeric(0)
izq.y3.n1.2.ADI <- numeric(0)
der.y3.n2.2.ADI <- numeric(0)
izq.y3.n2.2.ADI <- numeric(0)
```

```
der.y1.n1.1.CV.ADI <- numeric(0)
izq.y1.n1.1.CV.ADI <- numeric(0)
der.y1.n2.1.CV.ADI <- numeric(0)
izq.y1.n2.1.CV.ADI <- numeric(0)
der.y2.n1.1.CV.ADI <- numeric(0)
izq.y2.n1.1.CV.ADI <- numeric(0)
der.y2.n2.1.CV.ADI <- numeric(0)
izq.y2.n2.1.CV.ADI <- numeric(0)
der.y3.n1.1.CV.ADI <- numeric(0)
izq.y3.n1.1.CV.ADI <- numeric(0)
der.y3.n2.1.CV.ADI <- numeric(0)
izq.y3.n2.1.CV.ADI <- numeric(0)
```

```
der.y1.n1.2.CV.ADI <- numeric(0)
izq.y1.n1.2.CV.ADI <- numeric(0)
der.y1.n2.2.CV.ADI <- numeric(0)
izq.y1.n2.2.CV.ADI <- numeric(0)
der.y2.n1.2.CV.ADI <- numeric(0)
izq.y2.n1.2.CV.ADI <- numeric(0)
der.y2.n2.2.CV.ADI <- numeric(0)
izq.y2.n2.2.CV.ADI <- numeric(0)
```

## Anexos

---

```
der.y3.n1.2.CV.ADI <- numeric(0)
izq.y3.n1.2.CV.ADI <- numeric(0)
der.y3.n2.2.CV.ADI <- numeric(0)
izq.y3.n2.2.CV.ADI <- numeric(0)

der.y1.n1.1.CV.g_W.ADI <- numeric(0)
izq.y1.n1.1.CV.g_W.ADI <- numeric(0)
der.y1.n2.1.CV.g_W.ADI <- numeric(0)
izq.y1.n2.1.CV.g_W.ADI <- numeric(0)
der.y2.n1.1.CV.g_W.ADI <- numeric(0)
izq.y2.n1.1.CV.g_W.ADI <- numeric(0)
der.y2.n2.1.CV.g_W.ADI <- numeric(0)
izq.y2.n2.1.CV.g_W.ADI <- numeric(0)
der.y3.n1.1.CV.g_W.ADI <- numeric(0)
izq.y3.n1.1.CV.g_W.ADI <- numeric(0)
der.y3.n2.1.CV.g_W.ADI <- numeric(0)
izq.y3.n2.1.CV.g_W.ADI <- numeric(0)

der.y1.n1.2.CV.g_W.ADI <- numeric(0)
izq.y1.n1.2.CV.g_W.ADI <- numeric(0)
der.y1.n2.2.CV.g_W.ADI <- numeric(0)
izq.y1.n2.2.CV.g_W.ADI <- numeric(0)
der.y2.n1.2.CV.g_W.ADI <- numeric(0)
izq.y2.n1.2.CV.g_W.ADI <- numeric(0)
der.y2.n2.2.CV.g_W.ADI <- numeric(0)
izq.y2.n2.2.CV.g_W.ADI <- numeric(0)
der.y3.n1.2.CV.g_W.ADI <- numeric(0)
izq.y3.n1.2.CV.g_W.ADI <- numeric(0)
der.y3.n2.2.CV.g_W.ADI <- numeric(0)
izq.y3.n2.2.CV.g_W.ADI <- numeric(0)

der.y1.n1.1.g_W.ADI <- numeric(0)
izq.y1.n1.1.g_W.ADI <- numeric(0)
der.y1.n2.1.g_W.ADI <- numeric(0)
izq.y1.n2.1.g_W.ADI <- numeric(0)
der.y2.n1.1.g_W.ADI <- numeric(0)
izq.y2.n1.1.g_W.ADI <- numeric(0)
der.y2.n2.1.g_W.ADI <- numeric(0)
izq.y2.n2.1.g_W.ADI <- numeric(0)
der.y3.n1.1.g_W.ADI <- numeric(0)
izq.y3.n1.1.g_W.ADI <- numeric(0)
der.y3.n2.1.g_W.ADI <- numeric(0)
izq.y3.n2.1.g_W.ADI <- numeric(0)

der.y1.n1.2.g_W.ADI <- numeric(0)
izq.y1.n1.2.g_W.ADI <- numeric(0)
der.y1.n2.2.g_W.ADI <- numeric(0)
izq.y1.n2.2.g_W.ADI <- numeric(0)
der.y2.n1.2.g_W.ADI <- numeric(0)
izq.y2.n1.2.g_W.ADI <- numeric(0)
der.y2.n2.2.g_W.ADI <- numeric(0)
izq.y2.n2.2.g_W.ADI <- numeric(0)
der.y3.n1.2.g_W.ADI <- numeric(0)
izq.y3.n1.2.g_W.ADI <- numeric(0)
der.y3.n2.2.g_W.ADI <- numeric(0)
izq.y3.n2.2.g_W.ADI <- numeric(0)

der.y1.n1.1.LIN <- numeric(0)
izq.y1.n1.1.LIN <- numeric(0)
der.y1.n2.1.LIN <- numeric(0)
izq.y1.n2.1.LIN <- numeric(0)
der.y2.n1.1.LIN <- numeric(0)
izq.y2.n1.1.LIN <- numeric(0)
der.y2.n2.1.LIN <- numeric(0)
izq.y2.n2.1.LIN <- numeric(0)
der.y3.n1.1.LIN <- numeric(0)
izq.y3.n1.1.LIN <- numeric(0)
der.y3.n2.1.LIN <- numeric(0)
```

```

izq.y3.n2.1.LIN <- numeric(0)

der.y1.n1.2.LIN <- numeric(0)
izq.y1.n1.2.LIN <- numeric(0)
der.y1.n2.2.LIN <- numeric(0)
izq.y1.n2.2.LIN <- numeric(0)
der.y2.n1.2.LIN <- numeric(0)
izq.y2.n1.2.LIN <- numeric(0)
der.y2.n2.2.LIN <- numeric(0)
izq.y2.n2.2.LIN <- numeric(0)
der.y3.n1.2.LIN <- numeric(0)
izq.y3.n1.2.LIN <- numeric(0)
der.y3.n2.2.LIN <- numeric(0)
izq.y3.n2.2.LIN <- numeric(0)

der.y1.n1.1.g_W.LIN <- numeric(0)
izq.y1.n1.1.g_W.LIN <- numeric(0)
der.y1.n2.1.g_W.LIN <- numeric(0)
izq.y1.n2.1.g_W.LIN <- numeric(0)
der.y2.n1.1.g_W.LIN <- numeric(0)
izq.y2.n1.1.g_W.LIN <- numeric(0)
der.y2.n2.1.g_W.LIN <- numeric(0)
izq.y2.n2.1.g_W.LIN <- numeric(0)
der.y3.n1.1.g_W.LIN <- numeric(0)
izq.y3.n1.1.g_W.LIN <- numeric(0)
der.y3.n2.1.g_W.LIN <- numeric(0)
izq.y3.n2.1.g_W.LIN <- numeric(0)

der.y1.n1.2.g_W.LIN <- numeric(0)
izq.y1.n1.2.g_W.LIN <- numeric(0)
der.y1.n2.2.g_W.LIN <- numeric(0)
izq.y1.n2.2.g_W.LIN <- numeric(0)
der.y2.n1.2.g_W.LIN <- numeric(0)
izq.y2.n1.2.g_W.LIN <- numeric(0)
der.y2.n2.2.g_W.LIN <- numeric(0)
izq.y2.n2.2.g_W.LIN <- numeric(0)
der.y3.n1.2.g_W.LIN <- numeric(0)
izq.y3.n1.2.g_W.LIN <- numeric(0)
der.y3.n2.2.g_W.LIN <- numeric(0)
izq.y3.n2.2.g_W.LIN <- numeric(0)

t.C <- matrix(apply(C_U,2,sum),1,)
t.C2 <- matrix(apply(C2_U,2,sum),1,)

for(k in 1:M)
{
  s1 <- sample(1:N,n1)
  s2 <- sample(1:N,n2)

#####
### VARYING COEFFICIENT MODELS ###
#####

  for (j in 1:H)
  {
    w.s1 <- function(lambda1,lambda2)
    {
      w_s1 <- C_U[s1,]*%solve(t(C_U[s1,])%*%Pi.inv.n1)%*C_U[s1,] +
        Delta(lambda1,lambda2)%*%t(C_U[s1,])%*%Pi.inv.n1
      return(t(w_s1))
    }

    w.s2 <- function(lambda1,lambda2)
    {
      w_s2 <- C_U[s2,]*%solve(t(C_U[s2,])%*%Pi.inv.n2)%*C_U[s2,] +
        Delta(lambda1,lambda2)%*%t(C_U[s2,])%*%Pi.inv.n2
    }
  }
}

```

```

    return(t(w_s2))
  }

  rr.1 <- 1/(1-diag(w.s1(d[j,1],d[j,2])))
  W.s.1 <- w.s1(d[j,1],d[j,2])-diag(diag(w.s1(d[j,1],d[j,2])))

  rr.2 <- 1/(1-diag(w.s2(d[j,1],d[j,2])))
  W.s.2 <- w.s2(d[j,1],d[j,2])-diag(diag(w.s2(d[j,1],d[j,2])))

tt.y1.1.1 <- matrix(rep(y.1.1[s1],n1),n1,n1)
tt.y1.2.1 <- matrix(rep(y.1.1[s2],n2),n2,n2)
tt.y2.1.1 <- matrix(rep(y.2.1[s1],n1),n1,n1)
tt.y2.2.1 <- matrix(rep(y.2.1[s2],n2),n2,n2)
tt.y3.1.1 <- matrix(rep(y.3.1[s1],n1),n1,n1)
tt.y3.2.1 <- matrix(rep(y.3.1[s2],n2),n2,n2)

tt.y1.1.2 <- matrix(rep(y.1.2[s1],n1),n1,n1)
tt.y1.2.2 <- matrix(rep(y.1.2[s2],n2),n2,n2)
tt.y2.1.2 <- matrix(rep(y.2.2[s1],n1),n1,n1)
tt.y2.2.2 <- matrix(rep(y.2.2[s2],n2),n2,n2)
tt.y3.1.2 <- matrix(rep(y.3.2[s1],n1),n1,n1)
tt.y3.2.2 <- matrix(rep(y.3.2[s2],n2),n2,n2)

W.t.y1.1.1 <- tt.y1.1.1-diag(diag(tt.y1.1.1))
W.t.y1.2.1 <- tt.y1.2.1-diag(diag(tt.y1.2.1))
W.t.y2.1.1 <- tt.y2.1.1-diag(diag(tt.y2.1.1))
W.t.y2.2.1 <- tt.y2.2.1-diag(diag(tt.y2.2.1))
W.t.y3.1.1 <- tt.y3.1.1-diag(diag(tt.y3.1.1))
W.t.y3.2.1 <- tt.y3.2.1-diag(diag(tt.y3.2.1))

W.t.y1.1.2 <- tt.y1.1.2-diag(diag(tt.y1.1.2))
W.t.y1.2.2 <- tt.y1.2.2-diag(diag(tt.y1.2.2))
W.t.y2.1.2 <- tt.y2.1.2-diag(diag(tt.y2.1.2))
W.t.y2.2.2 <- tt.y2.2.2-diag(diag(tt.y2.2.2))
W.t.y3.1.2 <- tt.y3.1.2-diag(diag(tt.y3.1.2))
W.t.y3.2.2 <- tt.y3.2.2-diag(diag(tt.y3.2.2))

v1 <- (y.1.1[s1] - rr.1*apply(W.s.1*W.t.y1.1.1,2,sum))
v2 <- (y.1.1[s2] - rr.2*apply(W.s.2*W.t.y1.2.1,2,sum))
v3 <- (y.2.1[s1] - rr.1*apply(W.s.1*W.t.y2.1.1,2,sum))
v4 <- (y.2.1[s2] - rr.2*apply(W.s.2*W.t.y2.2.1,2,sum))
v5 <- (y.3.1[s1] - rr.1*apply(W.s.1*W.t.y3.1.1,2,sum))
v6 <- (y.3.1[s2] - rr.2*apply(W.s.2*W.t.y3.2.1,2,sum))

v7 <- (y.1.2[s1] - rr.1*apply(W.s.1*W.t.y1.1.2,2,sum))
v8 <- (y.1.2[s2] - rr.2*apply(W.s.2*W.t.y1.2.2,2,sum))
v9 <- (y.2.2[s1] - rr.1*apply(W.s.1*W.t.y2.1.2,2,sum))
v10 <- (y.2.2[s2] - rr.2*apply(W.s.2*W.t.y2.2.2,2,sum))
v11 <- (y.3.2[s1] - rr.1*apply(W.s.1*W.t.y3.1.2,2,sum))
v12 <- (y.3.2[s2] - rr.2*apply(W.s.2*W.t.y3.2.2,2,sum))

V1[j] <- N.1*var(c(v1))
V2[j] <- N.2*var(c(v2))
V3[j] <- N.1*var(c(v3))
V4[j] <- N.2*var(c(v4))
V5[j] <- N.1*var(c(v5))
V6[j] <- N.2*var(c(v6))

V7[j] <- N.1*var(c(v7))
V8[j] <- N.2*var(c(v8))
V9[j] <- N.1*var(c(v9))
V10[j] <- N.2*var(c(v10))
V11[j] <- N.1*var(c(v11))
V12[j] <- N.2*var(c(v12))

```

```

}

L1L2.y1.n1.1[k,] <- d[which(V1==min(V1)),]
L1L2.y1.n2.1[k,] <- d[which(V2==min(V2)),]
L1L2.y2.n1.1[k,] <- d[which(V3==min(V3)),]
L1L2.y2.n2.1[k,] <- d[which(V4==min(V4)),]
L1L2.y3.n1.1[k,] <- d[which(V5==min(V5)),]
L1L2.y3.n2.1[k,] <- d[which(V6==min(V6)),]

L1L2.y1.n1.2[k,] <- d[which(V7==min(V7)),]
L1L2.y1.n2.2[k,] <- d[which(V8==min(V8)),]
L1L2.y2.n1.2[k,] <- d[which(V9==min(V9)),]
L1L2.y2.n2.2[k,] <- d[which(V10==min(V10)),]
L1L2.y3.n1.2[k,] <- d[which(V11==min(V11)),]
L1L2.y3.n2.2[k,] <- d[which(V12==min(V12)),]

V1.hat[k] <- V1[which(V1==min(V1))]
V2.hat[k] <- V2[which(V2==min(V2))]
V3.hat[k] <- V3[which(V3==min(V3))]
V4.hat[k] <- V4[which(V4==min(V4))]
V5.hat[k] <- V5[which(V5==min(V5))]
V6.hat[k] <- V6[which(V6==min(V6))]
V7.hat[k] <- V7[which(V7==min(V7))]
V8.hat[k] <- V8[which(V8==min(V8))]
V9.hat[k] <- V9[which(V9==min(V9))]
V10.hat[k] <- V10[which(V10==min(V10))]
V11.hat[k] <- V11[which(V11==min(V11))]
V12.hat[k] <- V12[which(V12==min(V12))]

#####
### ADDITIVE MODELS ###
#####

for (j in 1:H2)
{
  w.s1.ad <- function(lambda1,lambda2)
  {
    w_s1 <- C2_U[s1,]%*%solve(t(C2_U[s1,])%*%Pi.inv.n1%*%C2_U[s1,] +
      Delta2(lambda1,lambda2))%*%t(C2_U[s1,])%*%Pi.inv.n1
    return(t(w_s1))
  }

  w.s2.ad <- function(lambda1,lambda2)
  {
    w_s2 <- C2_U[s2,]%*%solve(t(C2_U[s2,])%*%Pi.inv.n2%*%C2_U[s2,] +
      Delta2(lambda1,lambda2))%*%t(C2_U[s2,])%*%Pi.inv.n2
    return(t(w_s2))
  }

  rr.1 <- 1/(1-diag(w.s1.ad(d2[j,1],d2[j,2])))
  W.s.1 <- w.s1.ad(d2[j,1],d2[j,2])-diag(diag(w.s1.ad(d2[j,1],d2[j,2])))

  rr.2 <- 1/(1-diag(w.s2.ad(d2[j,1],d2[j,2])))
  W.s.2 <- w.s2.ad(d2[j,1],d2[j,2])-diag(diag(w.s2.ad(d2[j,1],d2[j,2])))

  tt.y1.1.1 <- matrix(rep(y.1.1[s1],n1),n1,n1)
  tt.y1.2.1 <- matrix(rep(y.1.1[s2],n2),n2,n2)
  tt.y2.1.1 <- matrix(rep(y.2.1[s1],n1),n1,n1)
  tt.y2.2.1 <- matrix(rep(y.2.1[s2],n2),n2,n2)
  tt.y3.1.1 <- matrix(rep(y.3.1[s1],n1),n1,n1)
  tt.y3.2.1 <- matrix(rep(y.3.1[s2],n2),n2,n2)

  tt.y1.1.2 <- matrix(rep(y.1.2[s1],n1),n1,n1)
  tt.y1.2.2 <- matrix(rep(y.1.2[s2],n2),n2,n2)

```



```

tt.y2.1.2 <- matrix(rep(y.2.2[s1],n1),n1,n1)
tt.y2.2.2 <- matrix(rep(y.2.2[s2],n2),n2,n2)
tt.y3.1.2 <- matrix(rep(y.3.2[s1],n1),n1,n1)
tt.y3.2.2 <- matrix(rep(y.3.2[s2],n2),n2,n2)

W.t.y1.1.1 <- tt.y1.1.1-diag(diag(tt.y1.1.1))
W.t.y1.2.1 <- tt.y1.2.1-diag(diag(tt.y1.2.1))
W.t.y2.1.1 <- tt.y2.1.1-diag(diag(tt.y2.1.1))
W.t.y2.2.1 <- tt.y2.2.1-diag(diag(tt.y2.2.1))
W.t.y3.1.1 <- tt.y3.1.1-diag(diag(tt.y3.1.1))
W.t.y3.2.1 <- tt.y3.2.1-diag(diag(tt.y3.2.1))

W.t.y1.1.2 <- tt.y1.1.2-diag(diag(tt.y1.1.2))
W.t.y1.2.2 <- tt.y1.2.2-diag(diag(tt.y1.2.2))
W.t.y2.1.2 <- tt.y2.1.2-diag(diag(tt.y2.1.2))
W.t.y2.2.2 <- tt.y2.2.2-diag(diag(tt.y2.2.2))
W.t.y3.1.2 <- tt.y3.1.2-diag(diag(tt.y3.1.2))
W.t.y3.2.2 <- tt.y3.2.2-diag(diag(tt.y3.2.2))

v1 <- (y.1.1[s1] - rr.1*apply(W.s.1*W.t.y1.1.1,2,sum))
v2 <- (y.1.1[s2] - rr.2*apply(W.s.2*W.t.y1.2.1,2,sum))
v3 <- (y.2.1[s1] - rr.1*apply(W.s.1*W.t.y2.1.1,2,sum))
v4 <- (y.2.1[s2] - rr.2*apply(W.s.2*W.t.y2.2.1,2,sum))
v5 <- (y.3.1[s1] - rr.1*apply(W.s.1*W.t.y3.1.1,2,sum))
v6 <- (y.3.1[s2] - rr.2*apply(W.s.2*W.t.y3.2.1,2,sum))

v7 <- (y.1.2[s1] - rr.1*apply(W.s.1*W.t.y1.1.2,2,sum))
v8 <- (y.1.2[s2] - rr.2*apply(W.s.2*W.t.y1.2.2,2,sum))
v9 <- (y.2.2[s1] - rr.1*apply(W.s.1*W.t.y2.1.2,2,sum))
v10 <- (y.2.2[s2] - rr.2*apply(W.s.2*W.t.y2.2.2,2,sum))
v11 <- (y.3.2[s1] - rr.1*apply(W.s.1*W.t.y3.1.2,2,sum))
v12 <- (y.3.2[s2] - rr.2*apply(W.s.2*W.t.y3.2.2,2,sum))

V13[j] <- N.1*var(c(v1))
V14[j] <- N.2*var(c(v2))
V15[j] <- N.1*var(c(v3))
V16[j] <- N.2*var(c(v4))
V17[j] <- N.1*var(c(v5))
V18[j] <- N.2*var(c(v6))

V19[j] <- N.1*var(c(v7))
V20[j] <- N.2*var(c(v8))
V21[j] <- N.1*var(c(v9))
V22[j] <- N.2*var(c(v10))
V23[j] <- N.1*var(c(v11))
V24[j] <- N.2*var(c(v12))
}

L1L2.y1.n1.1.ADI[k,] <- d2[which(V13==min(V13)),]
L1L2.y1.n2.1.ADI[k,] <- d2[which(V14==min(V14)),]
L1L2.y2.n1.1.ADI[k,] <- d2[which(V15==min(V15)),]
L1L2.y2.n2.1.ADI[k,] <- d2[which(V16==min(V16)),]
L1L2.y3.n1.1.ADI[k,] <- d2[which(V17==min(V17)),]
L1L2.y3.n2.1.ADI[k,] <- d2[which(V18==min(V18)),]

L1L2.y1.n1.2.ADI[k,] <- d2[which(V19==min(V19)),]
L1L2.y1.n2.2.ADI[k,] <- d2[which(V20==min(V20)),]
L1L2.y2.n1.2.ADI[k,] <- d2[which(V21==min(V21)),]
L1L2.y2.n2.2.ADI[k,] <- d2[which(V22==min(V22)),]
L1L2.y3.n1.2.ADI[k,] <- d2[which(V23==min(V23)),]
L1L2.y3.n2.2.ADI[k,] <- d2[which(V24==min(V24)),]

V13.hat[k] <- V13[which(V13==min(V13))]
V14.hat[k] <- V14[which(V14==min(V14))]

```

## Anexos

---

```
V15.hat[k] <- V15[which(V15==min(V15))]
V16.hat[k] <- V16[which(V16==min(V16))]
V17.hat[k] <- V17[which(V17==min(V17))]
V18.hat[k] <- V18[which(V18==min(V18))]
V19.hat[k] <- V19[which(V19==min(V19))]
V20.hat[k] <- V20[which(V20==min(V20))]
V21.hat[k] <- V21[which(V21==min(V21))]
V22.hat[k] <- V22[which(V22==min(V22))]
V23.hat[k] <- V23[which(V23==min(V23))]
V24.hat[k] <- V24[which(V24==min(V24))]

#####
## DEGREES FREEDOM ##
#####

#### VCM ####

AF.s1 <- function(lam1,lam2)
{
  af.s1 <- solve(t(C_U[s1,])%*%Pi.inv.n1%*%C_U[s1,]+
    Delta(lam1,lam2))%*%t(C_U[s1,])%*%Pi.inv.n1
  return(af.s1)
}

AF.s2 <- function(lam1,lam2)
{
  af.s2 <- solve(t(C_U[s2,])%*%Pi.inv.n2%*%C_U[s2,]+
    Delta(lam1,lam2))%*%t(C_U[s2,])%*%Pi.inv.n2
  return(af.s2)
}

df.y1.n1.1[k] <- sum(diag(AF.s1(L1L2.y1.n1.1[k,1],L1L2.y1.n1.1[k,2]))%*%C_U[s1,])
df.y1.n2.1[k] <- sum(diag(AF.s2(L1L2.y1.n2.1[k,1],L1L2.y1.n2.1[k,2]))%*%C_U[s2,])
df.y2.n1.1[k] <- sum(diag(AF.s1(L1L2.y2.n1.1[k,1],L1L2.y2.n1.1[k,2]))%*%C_U[s1,])
df.y2.n2.1[k] <- sum(diag(AF.s2(L1L2.y2.n2.1[k,1],L1L2.y2.n2.1[k,2]))%*%C_U[s2,])
df.y3.n1.1[k] <- sum(diag(AF.s1(L1L2.y3.n1.1[k,1],L1L2.y3.n1.1[k,2]))%*%C_U[s1,])
df.y3.n2.1[k] <- sum(diag(AF.s2(L1L2.y3.n2.1[k,1],L1L2.y3.n2.1[k,2]))%*%C_U[s2,])

df.y1.n1.2[k] <- sum(diag(AF.s1(L1L2.y1.n1.2[k,1],L1L2.y1.n1.2[k,2]))%*%C_U[s1,])
df.y1.n2.2[k] <- sum(diag(AF.s2(L1L2.y1.n2.2[k,1],L1L2.y1.n2.2[k,2]))%*%C_U[s2,])
df.y2.n1.2[k] <- sum(diag(AF.s1(L1L2.y2.n1.2[k,1],L1L2.y2.n1.2[k,2]))%*%C_U[s1,])
df.y2.n2.2[k] <- sum(diag(AF.s2(L1L2.y2.n2.2[k,1],L1L2.y2.n2.2[k,2]))%*%C_U[s2,])
df.y3.n1.2[k] <- sum(diag(AF.s1(L1L2.y3.n1.2[k,1],L1L2.y3.n1.2[k,2]))%*%C_U[s1,])
df.y3.n2.2[k] <- sum(diag(AF.s2(L1L2.y3.n2.2[k,1],L1L2.y3.n2.2[k,2]))%*%C_U[s2,])

#### AM ####

AF.s1.ADI <- function(lam1,lam2)
{
  af.s1 <- solve(t(C2_U[s1,])%*%Pi.inv.n1%*%C2_U[s1,]+ Delta2(lam1,lam2))%*%t(C2_U[s1,])%*%Pi.inv.n1
  return(af.s1)
}

AF.s2.ADI <- function(lam1,lam2)
{
  af.s2 <- solve(t(C2_U[s2,])%*%Pi.inv.n2%*%C2_U[s2,]+ Delta2(lam1,lam2))%*%t(C2_U[s2,])%*%Pi.inv.n2
  return(af.s2)
}

df.y1.n1.1.ADI[k] <- sum(diag(AF.s1.ADI(L1L2.y1.n1.1.ADI[k,1],L1L2.y1.n1.1.ADI[k,2]))%*%C2_U[s1,])
df.y1.n2.1.ADI[k] <- sum(diag(AF.s2.ADI(L1L2.y1.n2.1.ADI[k,1],L1L2.y1.n2.1.ADI[k,2]))%*%C2_U[s2,])
```

## Anexos

```
df.y2.n1.1.ADI[k] <- sum(diag(AF.s1.ADI(L1L2.y2.n1.1.ADI[k,1],L1L2.y2.n1.1.ADI[k,2]))**C2_U[s1,])
df.y2.n2.1.ADI[k] <- sum(diag(AF.s2.ADI(L1L2.y2.n2.1.ADI[k,1],L1L2.y2.n2.1.ADI[k,2]))**C2_U[s2,])
df.y3.n1.1.ADI[k] <- sum(diag(AF.s1.ADI(L1L2.y3.n1.1.ADI[k,1],L1L2.y3.n1.1.ADI[k,2]))**C2_U[s1,])
df.y3.n2.1.ADI[k] <- sum(diag(AF.s2.ADI(L1L2.y3.n2.1.ADI[k,1],L1L2.y3.n2.1.ADI[k,2]))**C2_U[s2,])

df.y1.n1.2.ADI[k] <- sum(diag(AF.s1.ADI(L1L2.y1.n1.2.ADI[k,1],L1L2.y1.n1.2.ADI[k,2]))**C2_U[s1,])
df.y1.n2.2.ADI[k] <- sum(diag(AF.s2.ADI(L1L2.y1.n2.2.ADI[k,1],L1L2.y1.n2.2.ADI[k,2]))**C2_U[s2,])
df.y2.n1.2.ADI[k] <- sum(diag(AF.s1.ADI(L1L2.y2.n1.2.ADI[k,1],L1L2.y2.n1.2.ADI[k,2]))**C2_U[s1,])
df.y2.n2.2.ADI[k] <- sum(diag(AF.s2.ADI(L1L2.y2.n2.2.ADI[k,1],L1L2.y2.n2.2.ADI[k,2]))**C2_U[s2,])
df.y3.n1.2.ADI[k] <- sum(diag(AF.s1.ADI(L1L2.y3.n1.2.ADI[k,1],L1L2.y3.n1.2.ADI[k,2]))**C2_U[s1,])
df.y3.n2.2.ADI[k] <- sum(diag(AF.s2.ADI(L1L2.y3.n2.2.ADI[k,1],L1L2.y3.n2.2.ADI[k,2]))**C2_U[s2,])
```

#### VCM ####

```
mhat_U.y1.n1.1 <- C_U**AF.s1(L1L2.y1.n1.1[k,1],L1L2.y1.n1.1[k,2])**cbind(y.1.1[s1])
mhat_s.y1.n1.1 <- C_U[s1,]**AF.s1(L1L2.y1.n1.1[k,1],L1L2.y1.n1.1[k,2])**cbind(y.1.1[s1])
mhat_U.y1.n2.1 <- C_U**AF.s2(L1L2.y1.n2.1[k,1],L1L2.y1.n2.1[k,2])**cbind(y.1.1[s2])
mhat_s.y1.n2.1 <- C_U[s2,]**AF.s2(L1L2.y1.n2.1[k,1],L1L2.y1.n2.1[k,2])**cbind(y.1.1[s2])

mhat_U.y2.n1.1 <- C_U**AF.s1(L1L2.y2.n1.1[k,1],L1L2.y2.n1.1[k,2])**cbind(y.2.1[s1])
mhat_s.y2.n1.1 <- C_U[s1,]**AF.s1(L1L2.y2.n1.1[k,1],L1L2.y2.n1.1[k,2])**cbind(y.2.1[s1])
mhat_U.y2.n2.1 <- C_U**AF.s2(L1L2.y2.n2.1[k,1],L1L2.y2.n2.1[k,2])**cbind(y.2.1[s2])
mhat_s.y2.n2.1 <- C_U[s2,]**AF.s2(L1L2.y2.n2.1[k,1],L1L2.y2.n2.1[k,2])**cbind(y.2.1[s2])

mhat_U.y3.n1.1 <- C_U**AF.s1(L1L2.y3.n1.1[k,1],L1L2.y3.n1.1[k,2])**cbind(y.3.1[s1])
mhat_s.y3.n1.1 <- C_U[s1,]**AF.s1(L1L2.y3.n1.1[k,1],L1L2.y3.n1.1[k,2])**cbind(y.3.1[s1])
mhat_U.y3.n2.1 <- C_U**AF.s2(L1L2.y3.n2.1[k,1],L1L2.y3.n2.1[k,2])**cbind(y.3.1[s2])
mhat_s.y3.n2.1 <- C_U[s2,]**AF.s2(L1L2.y3.n2.1[k,1],L1L2.y3.n2.1[k,2])**cbind(y.3.1[s2])
```

```
mhat_U.y1.n1.2 <- C_U**AF.s1(L1L2.y1.n1.2[k,1],L1L2.y1.n1.2[k,2])**cbind(y.1.2[s1])
mhat_s.y1.n1.2 <- C_U[s1,]**AF.s1(L1L2.y1.n1.2[k,1],L1L2.y1.n1.2[k,2])**cbind(y.1.2[s1])
mhat_U.y1.n2.2 <- C_U**AF.s2(L1L2.y1.n2.2[k,1],L1L2.y1.n2.2[k,2])**cbind(y.1.2[s2])
mhat_s.y1.n2.2 <- C_U[s2,]**AF.s2(L1L2.y1.n2.2[k,1],L1L2.y1.n2.2[k,2])**cbind(y.1.2[s2])
```

```
mhat_U.y2.n1.2 <- C_U**AF.s1(L1L2.y2.n1.2[k,1],L1L2.y2.n1.2[k,2])**cbind(y.2.2[s1])
mhat_s.y2.n1.2 <- C_U[s1,]**AF.s1(L1L2.y2.n1.2[k,1],L1L2.y2.n1.2[k,2])**cbind(y.2.2[s1])
mhat_U.y2.n2.2 <- C_U**AF.s2(L1L2.y2.n2.2[k,1],L1L2.y2.n2.2[k,2])**cbind(y.2.2[s2])
mhat_s.y2.n2.2 <- C_U[s2,]**AF.s2(L1L2.y2.n2.2[k,1],L1L2.y2.n2.2[k,2])**cbind(y.2.2[s2])
```

```
mhat_U.y3.n1.2 <- C_U**AF.s1(L1L2.y3.n1.2[k,1],L1L2.y3.n1.2[k,2])**cbind(y.3.2[s1])
mhat_s.y3.n1.2 <- C_U[s1,]**AF.s1(L1L2.y3.n1.2[k,1],L1L2.y3.n1.2[k,2])**cbind(y.3.2[s1])
mhat_U.y3.n2.2 <- C_U**AF.s2(L1L2.y3.n2.2[k,1],L1L2.y3.n2.2[k,2])**cbind(y.3.2[s2])
mhat_s.y3.n2.2 <- C_U[s2,]**AF.s2(L1L2.y3.n2.2[k,1],L1L2.y3.n2.2[k,2])**cbind(y.3.2[s2])
```

#### AM ####

```
mhat_U.y1.n1.1.ADI <- C2_U**AF.s1.ADI(L1L2.y1.n1.1.ADI[k,1],L1L2.y1.n1.1.ADI[k,2])**cbind(y.1.1[s1])
mhat_s.y1.n1.1.ADI <- C2_U[s1,]**AF.s1.ADI(L1L2.y1.n1.1.ADI[k,1],L1L2.y1.n1.1.ADI[k,2])**cbind(y.1.1[s1])
mhat_U.y1.n2.1.ADI <- C2_U**AF.s2.ADI(L1L2.y1.n2.1.ADI[k,1],L1L2.y1.n2.1.ADI[k,2])**cbind(y.1.1[s2])
mhat_s.y1.n2.1.ADI <- C2_U[s2,]**AF.s2.ADI(L1L2.y1.n2.1.ADI[k,1],L1L2.y1.n2.1.ADI[k,2])**cbind(y.1.1[s2])
```

```
mhat_U.y2.n1.1.ADI <- C2_U**AF.s1.ADI(L1L2.y2.n1.1.ADI[k,1],L1L2.y2.n1.1.ADI[k,2])**cbind(y.2.1[s1])
mhat_s.y2.n1.1.ADI <- C2_U[s1,]**AF.s1.ADI(L1L2.y2.n1.1.ADI[k,1],L1L2.y2.n1.1.ADI[k,2])**cbind(y.2.1[s1])
mhat_U.y2.n2.1.ADI <- C2_U**AF.s2.ADI(L1L2.y2.n2.1.ADI[k,1],L1L2.y2.n2.1.ADI[k,2])**cbind(y.2.1[s2])
mhat_s.y2.n2.1.ADI <- C2_U[s2,]**AF.s2.ADI(L1L2.y2.n2.1.ADI[k,1],L1L2.y2.n2.1.ADI[k,2])**cbind(y.2.1[s2])
```

```
mhat_U.y3.n1.1.ADI <- C2_U**AF.s1.ADI(L1L2.y3.n1.1.ADI[k,1],L1L2.y3.n1.1.ADI[k,2])**cbind(y.3.1[s1])
mhat_s.y3.n1.1.ADI <- C2_U[s1,]**AF.s1.ADI(L1L2.y3.n1.1.ADI[k,1],L1L2.y3.n1.1.ADI[k,2])**cbind(y.3.1[s1])
mhat_U.y3.n2.1.ADI <- C2_U**AF.s2.ADI(L1L2.y3.n2.1.ADI[k,1],L1L2.y3.n2.1.ADI[k,2])**cbind(y.3.1[s2])
mhat_s.y3.n2.1.ADI <- C2_U[s2,]**AF.s2.ADI(L1L2.y3.n2.1.ADI[k,1],L1L2.y3.n2.1.ADI[k,2])**cbind(y.3.1[s2])
```

```
mhat_U.y1.n1.2.ADI <- C2_U**AF.s1.ADI(L1L2.y1.n1.2.ADI[k,1],L1L2.y1.n1.2.ADI[k,2])**cbind(y.1.2[s1])
```

## Anexos

```
mhat_s.y1.n1.2.ADI <- C2_U[s1,]*%AF.s1.ADI(L1L2.y1.n1.2.ADI[k,1],L1L2.y1.n1.2.ADI[k,2])*%cbind(y.1.2[s1])
mhat_U.y1.n2.2.ADI <- C2_U*s2.ADI(L1L2.y1.n2.2.ADI[k,1],L1L2.y1.n2.2.ADI[k,2])*%cbind(y.1.2[s2])
mhat_s.y1.n2.2.ADI <- C2_U[s2,]*%AF.s2.ADI(L1L2.y1.n2.2.ADI[k,1],L1L2.y1.n2.2.ADI[k,2])*%cbind(y.1.2[s2])

mhat_U.y2.n1.2.ADI <- C2_U*s2.ADI(L1L2.y2.n1.2.ADI[k,1],L1L2.y2.n1.2.ADI[k,2])*%cbind(y.2.2[s1])
mhat_s.y2.n1.2.ADI <- C2_U[s1,]*%AF.s1.ADI(L1L2.y2.n1.2.ADI[k,1],L1L2.y2.n1.2.ADI[k,2])*%cbind(y.2.2[s1])
mhat_U.y2.n2.2.ADI <- C2_U*s2.ADI(L1L2.y2.n2.2.ADI[k,1],L1L2.y2.n2.2.ADI[k,2])*%cbind(y.2.2[s2])
mhat_s.y2.n2.2.ADI <- C2_U[s2,]*%AF.s2.ADI(L1L2.y2.n2.2.ADI[k,1],L1L2.y2.n2.2.ADI[k,2])*%cbind(y.2.2[s2])

mhat_U.y3.n1.2.ADI <- C2_U*s2.ADI(L1L2.y3.n1.2.ADI[k,1],L1L2.y3.n1.2.ADI[k,2])*%cbind(y.3.2[s1])
mhat_s.y3.n1.2.ADI <- C2_U[s1,]*%AF.s1.ADI(L1L2.y3.n1.2.ADI[k,1],L1L2.y3.n1.2.ADI[k,2])*%cbind(y.3.2[s1])
mhat_U.y3.n2.2.ADI <- C2_U*s2.ADI(L1L2.y3.n2.2.ADI[k,1],L1L2.y3.n2.2.ADI[k,2])*%cbind(y.3.2[s2])
mhat_s.y3.n2.2.ADI <- C2_U[s2,]*%AF.s2.ADI(L1L2.y3.n2.2.ADI[k,1],L1L2.y3.n2.2.ADI[k,2])*%cbind(y.3.2[s2])

#### LM ####

L.s1 <- function(n)
{
  XX <- X_U.LIN[,1:n]
  L.s1 <- solve(t(XX[s1,])*%Pi.inv.n1)*%XX[s1,])*%t(XX[s1,])*%Pi.inv.n1
  return(L.s1)
}

L.s2 <- function(n)
{
  XX <- X_U.LIN[,1:n]
  L.s2 <- solve(t(XX[s2,])*%Pi.inv.n2)*%XX[s2,])*%t(XX[s2,])*%Pi.inv.n2
  return(L.s2)
}

mhat_U.y1.n1.LIN.1 <- X_U.LIN*s1(10)*%cbind(y.1.1[s1])
mhat_s.y1.n1.LIN.1 <- X_U.LIN[s1,]*%L.s1(10)*%cbind(y.1.1[s1])
mhat_U.y1.n2.LIN.1 <- X_U.LIN*s2(10)*%cbind(y.1.1[s2])
mhat_s.y1.n2.LIN.1 <- X_U.LIN[s2,]*%L.s2(10)*%cbind(y.1.1[s2])

mhat_U.y2.n1.LIN.1 <- X_U.LIN*s1(10)*%cbind(y.2.1[s1])
mhat_s.y2.n1.LIN.1 <- X_U.LIN[s1,]*%L.s1(10)*%cbind(y.2.1[s1])
mhat_U.y2.n2.LIN.1 <- X_U.LIN*s2(10)*%cbind(y.2.1[s2])
mhat_s.y2.n2.LIN.1 <- X_U.LIN[s2,]*%L.s2(10)*%cbind(y.2.1[s2])

mhat_U.y3.n1.LIN.1 <- X_U.LIN*s1(10)*%cbind(y.3.1[s1])
mhat_s.y3.n1.LIN.1 <- X_U.LIN[s1,]*%L.s1(10)*%cbind(y.3.1[s1])
mhat_U.y3.n2.LIN.1 <- X_U.LIN*s2(10)*%cbind(y.3.1[s2])
mhat_s.y3.n2.LIN.1 <- X_U.LIN[s2,]*%L.s2(10)*%cbind(y.3.1[s2])

mhat_U.y1.n1.LIN.2 <- X_U.LIN*s1(10)*%cbind(y.1.2[s1])
mhat_s.y1.n1.LIN.2 <- X_U.LIN[s1,]*%L.s1(10)*%cbind(y.1.2[s1])
mhat_U.y1.n2.LIN.2 <- X_U.LIN*s2(10)*%cbind(y.1.2[s2])
mhat_s.y1.n2.LIN.2 <- X_U.LIN[s2,]*%L.s2(10)*%cbind(y.1.2[s2])

mhat_U.y2.n1.LIN.2 <- X_U.LIN*s1(10)*%cbind(y.2.2[s1])
mhat_s.y2.n1.LIN.2 <- X_U.LIN[s1,]*%L.s1(10)*%cbind(y.2.2[s1])
mhat_U.y2.n2.LIN.2 <- X_U.LIN*s2(10)*%cbind(y.2.2[s2])
mhat_s.y2.n2.LIN.2 <- X_U.LIN[s2,]*%L.s2(10)*%cbind(y.2.2[s2])

mhat_U.y3.n1.LIN.2 <- X_U.LIN*s1(10)*%cbind(y.3.2[s1])
mhat_s.y3.n1.LIN.2 <- X_U.LIN[s1,]*%L.s1(10)*%cbind(y.3.2[s1])
mhat_U.y3.n2.LIN.2 <- X_U.LIN*s2(10)*%cbind(y.3.2[s2])
mhat_s.y3.n2.LIN.2 <- X_U.LIN[s2,]*%L.s2(10)*%cbind(y.3.2[s2])

### VCM ###   ### without g-weights, with g-weights   ####

t.C.s1 <- (N/n1)*matrix(apply(C_U[s1,],2,sum),1,)
t.C.s2 <- (N/n2)*matrix(apply(C_U[s2,],2,sum),1,)
```

```

G.k.s1 <- function(lam1,lam2)
{
  G.k.s1 <- 1+(t.C-t.C.s1)%%solve(t(C_U[s1,])%%Pi.inv.n1)%%C_U[s1,] + Delta(lam1,lam2))%%t(C_U[s1,])
  return(t(G.k.s1))
}

G.k.s2 <- function(lam1,lam2)
{
  G.k.s2 <- 1+(t.C-t.C.s2)%%solve(t(C_U[s2,])%%Pi.inv.n2)%%C_U[s2,] + Delta(lam1,lam2))%%t(C_U[s2,])
  return(t(G.k.s2))
}

h1 <- (y.1.1[s1] - mhat_s.y1.n1.1)
h2 <- (y.1.1[s2] - mhat_s.y1.n2.1)
h3 <- (y.2.1[s1] - mhat_s.y2.n1.1)
h4 <- (y.2.1[s2] - mhat_s.y2.n2.1)
h5 <- (y.3.1[s1] - mhat_s.y3.n1.1)
h6 <- (y.3.1[s2] - mhat_s.y3.n2.1)

h7 <- (y.1.2[s1] - mhat_s.y1.n1.2)
h8 <- (y.1.2[s2] - mhat_s.y1.n2.2)
h9 <- (y.2.2[s1] - mhat_s.y2.n1.2)
h10 <- (y.2.2[s2] - mhat_s.y2.n2.2)
h11 <- (y.3.2[s1] - mhat_s.y3.n1.2)
h12 <- (y.3.2[s2] - mhat_s.y3.n2.2)

h13 <- G.k.s1(L1L2.y1.n1.1[k,1],L1L2.y1.n1.1[k,2])*(y.1.1[s1] - mhat_s.y1.n1.1)
h14 <- G.k.s2(L1L2.y1.n2.1[k,1],L1L2.y1.n2.1[k,2])*(y.1.1[s2] - mhat_s.y1.n2.1)
h15 <- G.k.s1(L1L2.y2.n1.1[k,1],L1L2.y2.n1.1[k,2])*(y.2.1[s1] - mhat_s.y2.n1.1)
h16 <- G.k.s2(L1L2.y2.n2.1[k,1],L1L2.y2.n2.1[k,2])*(y.2.1[s2] - mhat_s.y2.n2.1)
h17 <- G.k.s1(L1L2.y3.n1.1[k,1],L1L2.y3.n1.1[k,2])*(y.3.1[s1] - mhat_s.y3.n1.1)
h18 <- G.k.s2(L1L2.y3.n2.1[k,1],L1L2.y3.n2.1[k,2])*(y.3.1[s2] - mhat_s.y3.n2.1)

h19 <- G.k.s1(L1L2.y1.n1.2[k,1],L1L2.y1.n1.2[k,2])*(y.1.2[s1] - mhat_s.y1.n1.2)
h20 <- G.k.s2(L1L2.y1.n2.2[k,1],L1L2.y1.n2.2[k,2])*(y.1.2[s2] - mhat_s.y1.n2.2)
h21 <- G.k.s1(L1L2.y2.n1.2[k,1],L1L2.y2.n1.2[k,2])*(y.2.2[s1] - mhat_s.y2.n1.2)
h22 <- G.k.s2(L1L2.y2.n2.2[k,1],L1L2.y2.n2.2[k,2])*(y.2.2[s2] - mhat_s.y2.n2.2)
h23 <- G.k.s1(L1L2.y3.n1.2[k,1],L1L2.y3.n1.2[k,2])*(y.3.2[s1] - mhat_s.y3.n1.2)
h24 <- G.k.s2(L1L2.y3.n2.2[k,1],L1L2.y3.n2.2[k,2])*(y.3.2[s2] - mhat_s.y3.n2.2)

V25.hat[k] <- N.1*var(h1)
V26.hat[k] <- N.2*var(h2)
V27.hat[k] <- N.1*var(h3)
V28.hat[k] <- N.2*var(h4)
V29.hat[k] <- N.1*var(h5)
V30.hat[k] <- N.2*var(h6)
V31.hat[k] <- N.1*var(h7)
V32.hat[k] <- N.2*var(h8)
V33.hat[k] <- N.1*var(h9)
V34.hat[k] <- N.2*var(h10)
V35.hat[k] <- N.1*var(h11)
V36.hat[k] <- N.2*var(h12)

V37.hat[k] <- N.1*var(h13)
V38.hat[k] <- N.2*var(h14)
V39.hat[k] <- N.1*var(h15)
V40.hat[k] <- N.2*var(h16)
V41.hat[k] <- N.1*var(h17)
V42.hat[k] <- N.2*var(h18)
V43.hat[k] <- N.1*var(h19)
V44.hat[k] <- N.2*var(h20)
V45.hat[k] <- N.1*var(h21)

```

## Anexos

---

```
V46.hat[k] <- N.2*var(h22)
V47.hat[k] <- N.1*var(h23)
V48.hat[k] <- N.2*var(h24)
```

```
### AM ###      ### without g-weights, with g-weights ###
```

```
t.C2.s1 <- (N/n1)*matrix(apply(C2_U[s1,],2,sum),1,)
t.C2.s2 <- (N/n2)*matrix(apply(C2_U[s2,],2,sum),1,)
```

```
G.k.s1.ADI <- function(lam1,lam2)
{
  G.k.s1 <- 1+(t.C2-t.C2.s1)%*%solve(t(C2_U[s1,])%*%Pi.inv.n1%*%C2_U[s1,] +
    Delta2(lam1,lam2))%*%t(C2_U[s1,])
  return(t(G.k.s1))
}
```

```
G.k.s2.ADI <- function(lam1,lam2)
{
  G.k.s2 <- 1+(t.C2-t.C2.s2)%*%solve(t(C2_U[s2,])%*%Pi.inv.n2%*%C2_U[s2,] +
    Delta2(lam1,lam2))%*%t(C2_U[s2,])
  return(t(G.k.s2))
}
```

```
h25 <- (y.1.1[s1] - mhat_s.y1.n1.1.ADI)
h26 <- (y.1.1[s2] - mhat_s.y1.n2.1.ADI)
h27 <- (y.2.1[s1] - mhat_s.y2.n1.1.ADI)
h28 <- (y.2.1[s2] - mhat_s.y2.n2.1.ADI)
h29 <- (y.3.1[s1] - mhat_s.y3.n1.1.ADI)
h30 <- (y.3.1[s2] - mhat_s.y3.n2.1.ADI)
```

```
h31 <- (y.1.2[s1] - mhat_s.y1.n1.2.ADI)
h32 <- (y.1.2[s2] - mhat_s.y1.n2.2.ADI)
h33 <- (y.2.2[s1] - mhat_s.y2.n1.2.ADI)
h34 <- (y.2.2[s2] - mhat_s.y2.n2.2.ADI)
h35 <- (y.3.2[s1] - mhat_s.y3.n1.2.ADI)
h36 <- (y.3.2[s2] - mhat_s.y3.n2.2.ADI)
```

```
h37 <- G.k.s1.ADI(L1L2.y1.n1.1.ADI[k,1],L1L2.y1.n1.1.ADI[k,2])*(y.1.1[s1] - mhat_s.y1.n1.1.ADI)
h38 <- G.k.s2.ADI(L1L2.y1.n2.1.ADI[k,1],L1L2.y1.n2.1.ADI[k,2])*(y.1.1[s2] - mhat_s.y1.n2.1.ADI)
h39 <- G.k.s1.ADI(L1L2.y2.n1.1.ADI[k,1],L1L2.y2.n1.1.ADI[k,2])*(y.2.1[s1] - mhat_s.y2.n1.1.ADI)
h40 <- G.k.s2.ADI(L1L2.y2.n2.1.ADI[k,1],L1L2.y2.n2.1.ADI[k,2])*(y.2.1[s2] - mhat_s.y2.n2.1.ADI)
h41 <- G.k.s1.ADI(L1L2.y3.n1.1.ADI[k,1],L1L2.y3.n1.1.ADI[k,2])*(y.3.1[s1] - mhat_s.y3.n1.1.ADI)
h42 <- G.k.s2.ADI(L1L2.y3.n2.1.ADI[k,1],L1L2.y3.n2.1.ADI[k,2])*(y.3.1[s2] - mhat_s.y3.n2.1.ADI)
```

```
h43 <- G.k.s1.ADI(L1L2.y1.n1.2.ADI[k,1],L1L2.y1.n1.2.ADI[k,2])*(y.1.2[s1] - mhat_s.y1.n1.2.ADI)
h44 <- G.k.s2.ADI(L1L2.y1.n2.2.ADI[k,1],L1L2.y1.n2.2.ADI[k,2])*(y.1.2[s2] - mhat_s.y1.n2.2.ADI)
h45 <- G.k.s1.ADI(L1L2.y2.n1.2.ADI[k,1],L1L2.y2.n1.2.ADI[k,2])*(y.2.2[s1] - mhat_s.y2.n1.2.ADI)
h46 <- G.k.s2.ADI(L1L2.y2.n2.2.ADI[k,1],L1L2.y2.n2.2.ADI[k,2])*(y.2.2[s2] - mhat_s.y2.n2.2.ADI)
h47 <- G.k.s1.ADI(L1L2.y3.n1.2.ADI[k,1],L1L2.y3.n1.2.ADI[k,2])*(y.3.2[s1] - mhat_s.y3.n1.2.ADI)
h48 <- G.k.s2.ADI(L1L2.y3.n2.2.ADI[k,1],L1L2.y3.n2.2.ADI[k,2])*(y.3.2[s2] - mhat_s.y3.n2.2.ADI)
```

```
V49.hat[k] <- N.1*var(h25)
V50.hat[k] <- N.2*var(h26)
V51.hat[k] <- N.1*var(h27)
V52.hat[k] <- N.2*var(h28)
V53.hat[k] <- N.1*var(h29)
V54.hat[k] <- N.2*var(h30)
V55.hat[k] <- N.1*var(h31)
V56.hat[k] <- N.2*var(h32)
V57.hat[k] <- N.1*var(h33)
V58.hat[k] <- N.2*var(h34)
V59.hat[k] <- N.1*var(h35)
```

```

V60.hat[k] <- N.2*var(h36)

V61.hat[k] <- N.1*var(h37)
V62.hat[k] <- N.2*var(h38)
V63.hat[k] <- N.1*var(h39)
V64.hat[k] <- N.2*var(h40)
V65.hat[k] <- N.1*var(h41)
V66.hat[k] <- N.2*var(h42)
V67.hat[k] <- N.1*var(h43)
V68.hat[k] <- N.2*var(h44)
V69.hat[k] <- N.1*var(h45)
V70.hat[k] <- N.2*var(h46)
V71.hat[k] <- N.1*var(h47)
V72.hat[k] <- N.2*var(h48)

### VCM ###      ### with g-weigths and Cross-Validation #####

rr.1 <- 1/(1-diag(w.s1(L1L2.y1.n1.1[k,1],L1L2.y1.n1.1[k,2])))
rr.2 <- 1/(1-diag(w.s2(L1L2.y1.n2.1[k,1],L1L2.y1.n2.1[k,2])))
rr.3 <- 1/(1-diag(w.s1(L1L2.y2.n1.1[k,1],L1L2.y2.n1.1[k,2])))
rr.4 <- 1/(1-diag(w.s2(L1L2.y2.n2.1[k,1],L1L2.y2.n2.1[k,2])))
rr.5 <- 1/(1-diag(w.s1(L1L2.y3.n1.1[k,1],L1L2.y3.n1.1[k,2])))
rr.6 <- 1/(1-diag(w.s2(L1L2.y3.n2.1[k,1],L1L2.y3.n2.1[k,2])))

rr.7 <- 1/(1-diag(w.s1(L1L2.y1.n1.2[k,1],L1L2.y1.n1.2[k,2])))
rr.8 <- 1/(1-diag(w.s2(L1L2.y1.n2.2[k,1],L1L2.y1.n2.2[k,2])))
rr.9 <- 1/(1-diag(w.s1(L1L2.y2.n1.2[k,1],L1L2.y2.n1.2[k,2])))
rr.10 <- 1/(1-diag(w.s2(L1L2.y2.n2.2[k,1],L1L2.y2.n2.2[k,2])))
rr.11 <- 1/(1-diag(w.s1(L1L2.y3.n1.2[k,1],L1L2.y3.n1.2[k,2])))
rr.12 <- 1/(1-diag(w.s2(L1L2.y3.n2.2[k,1],L1L2.y3.n2.2[k,2])))

W.s.1 <- w.s1(L1L2.y1.n1.1[k,1],L1L2.y1.n1.1[k,2])-diag(diag(w.s1(L1L2.y1.n1.1[k,1],L1L2.y1.n1.1[k,2])))
W.s.2 <- w.s2(L1L2.y1.n2.1[k,1],L1L2.y1.n2.1[k,2])-diag(diag(w.s2(L1L2.y1.n2.1[k,1],L1L2.y1.n2.1[k,2])))
W.s.3 <- w.s1(L1L2.y2.n1.1[k,1],L1L2.y2.n1.1[k,2])-diag(diag(w.s1(L1L2.y2.n1.1[k,1],L1L2.y2.n1.1[k,2])))
W.s.4 <- w.s2(L1L2.y2.n2.1[k,1],L1L2.y2.n2.1[k,2])-diag(diag(w.s2(L1L2.y2.n2.1[k,1],L1L2.y2.n2.1[k,2])))
W.s.5 <- w.s1(L1L2.y3.n1.1[k,1],L1L2.y3.n1.1[k,2])-diag(diag(w.s1(L1L2.y3.n1.1[k,1],L1L2.y3.n1.1[k,2])))
W.s.6 <- w.s2(L1L2.y3.n2.1[k,1],L1L2.y3.n2.1[k,2])-diag(diag(w.s2(L1L2.y3.n2.1[k,1],L1L2.y3.n2.1[k,2])))

W.s.7 <- w.s1(L1L2.y1.n1.2[k,1],L1L2.y1.n1.2[k,2])-diag(diag(w.s1(L1L2.y1.n1.2[k,1],L1L2.y1.n1.2[k,2])))
W.s.8 <- w.s2(L1L2.y1.n2.2[k,1],L1L2.y1.n2.2[k,2])-diag(diag(w.s2(L1L2.y1.n2.2[k,1],L1L2.y1.n2.2[k,2])))
W.s.9 <- w.s1(L1L2.y2.n1.2[k,1],L1L2.y2.n1.2[k,2])-diag(diag(w.s1(L1L2.y2.n1.2[k,1],L1L2.y2.n1.2[k,2])))
W.s.10 <- w.s2(L1L2.y2.n2.2[k,1],L1L2.y2.n2.2[k,2])-diag(diag(w.s2(L1L2.y2.n2.2[k,1],L1L2.y2.n2.2[k,2])))
W.s.11 <- w.s1(L1L2.y3.n1.2[k,1],L1L2.y3.n1.2[k,2])-diag(diag(w.s1(L1L2.y3.n1.2[k,1],L1L2.y3.n1.2[k,2])))
W.s.12 <- w.s2(L1L2.y3.n2.2[k,1],L1L2.y3.n2.2[k,2])-diag(diag(w.s2(L1L2.y3.n2.2[k,1],L1L2.y3.n2.2[k,2])))

tt.y1.1.1 <- matrix(rep(y.1.1[s1],n1),n1,n1)
tt.y1.2.1 <- matrix(rep(y.1.1[s2],n2),n2,n2)
tt.y2.1.1 <- matrix(rep(y.2.1[s1],n1),n1,n1)
tt.y2.2.1 <- matrix(rep(y.2.1[s2],n2),n2,n2)
tt.y3.1.1 <- matrix(rep(y.3.1[s1],n1),n1,n1)
tt.y3.2.1 <- matrix(rep(y.3.1[s2],n2),n2,n2)

tt.y1.1.2 <- matrix(rep(y.1.2[s1],n1),n1,n1)
tt.y1.2.2 <- matrix(rep(y.1.2[s2],n2),n2,n2)
tt.y2.1.2 <- matrix(rep(y.2.2[s1],n1),n1,n1)
tt.y2.2.2 <- matrix(rep(y.2.2[s2],n2),n2,n2)
tt.y3.1.2 <- matrix(rep(y.3.2[s1],n1),n1,n1)
tt.y3.2.2 <- matrix(rep(y.3.2[s2],n2),n2,n2)

W.t.y1.1.1 <- tt.y1.1.1-diag(diag(tt.y1.1.1))
W.t.y1.2.1 <- tt.y1.2.1-diag(diag(tt.y1.2.1))
W.t.y2.1.1 <- tt.y2.1.1-diag(diag(tt.y2.1.1))
W.t.y2.2.1 <- tt.y2.2.1-diag(diag(tt.y2.2.1))
W.t.y3.1.1 <- tt.y3.1.1-diag(diag(tt.y3.1.1))

```

```

W.t.y3.2.1 <- tt.y3.2.1-diag(diag(tt.y3.2.1))

W.t.y1.1.2 <- tt.y1.1.2-diag(diag(tt.y1.1.2))
W.t.y1.2.2 <- tt.y1.2.2-diag(diag(tt.y1.2.2))
W.t.y2.1.2 <- tt.y2.1.2-diag(diag(tt.y2.1.2))
W.t.y2.2.2 <- tt.y2.2.2-diag(diag(tt.y2.2.2))
W.t.y3.1.2 <- tt.y3.1.2-diag(diag(tt.y3.1.2))
W.t.y3.2.2 <- tt.y3.2.2-diag(diag(tt.y3.2.2))

h49 <- (y.1.1[s1] - rr.1*apply(W.s.1*W.t.y1.1.1,2,sum))
h50 <- (y.1.1[s2] - rr.2*apply(W.s.2*W.t.y1.2.1,2,sum))
h51 <- (y.2.1[s1] - rr.3*apply(W.s.3*W.t.y2.1.1,2,sum))
h52 <- (y.2.1[s2] - rr.4*apply(W.s.4*W.t.y2.2.1,2,sum))
h53 <- (y.3.1[s1] - rr.5*apply(W.s.5*W.t.y3.1.1,2,sum))
h54 <- (y.3.1[s2] - rr.6*apply(W.s.6*W.t.y3.2.1,2,sum))

h55 <- (y.1.2[s1] - rr.7*apply(W.s.7*W.t.y1.1.2,2,sum))
h56 <- (y.1.2[s2] - rr.8*apply(W.s.8*W.t.y1.2.2,2,sum))
h57 <- (y.2.2[s1] - rr.9*apply(W.s.9*W.t.y2.1.2,2,sum))
h58 <- (y.2.2[s2] - rr.10*apply(W.s.10*W.t.y2.2.2,2,sum))
h59 <- (y.3.2[s1] - rr.11*apply(W.s.11*W.t.y3.1.2,2,sum))
h60 <- (y.3.2[s2] - rr.12*apply(W.s.12*W.t.y3.2.2,2,sum))

h61 <- G.k.s1(L1L2.y1.n1.1[k,1],L1L2.y1.n1.1[k,2])*h49
h62 <- G.k.s2(L1L2.y1.n2.1[k,1],L1L2.y1.n2.1[k,2])*h50
h63 <- G.k.s1(L1L2.y2.n1.1[k,1],L1L2.y2.n1.1[k,2])*h51
h64 <- G.k.s2(L1L2.y2.n2.1[k,1],L1L2.y2.n2.1[k,2])*h52
h65 <- G.k.s1(L1L2.y3.n1.1[k,1],L1L2.y3.n1.1[k,2])*h53
h66 <- G.k.s2(L1L2.y3.n2.1[k,1],L1L2.y3.n2.1[k,2])*h54

h67 <- G.k.s1(L1L2.y1.n1.2[k,1],L1L2.y1.n1.2[k,2])*h55
h68 <- G.k.s2(L1L2.y1.n2.2[k,1],L1L2.y1.n2.2[k,2])*h56
h69 <- G.k.s1(L1L2.y2.n1.2[k,1],L1L2.y2.n1.2[k,2])*h57
h70 <- G.k.s2(L1L2.y2.n2.2[k,1],L1L2.y2.n2.2[k,2])*h58
h71 <- G.k.s1(L1L2.y3.n1.2[k,1],L1L2.y3.n1.2[k,2])*h59
h72 <- G.k.s2(L1L2.y3.n2.2[k,1],L1L2.y3.n2.2[k,2])*h60

V73.hat[k] <- N.1*var(h61)
V74.hat[k] <- N.2*var(h62)
V75.hat[k] <- N.1*var(h63)
V76.hat[k] <- N.2*var(h64)
V77.hat[k] <- N.1*var(h65)
V78.hat[k] <- N.2*var(h66)
V79.hat[k] <- N.1*var(h67)
V80.hat[k] <- N.2*var(h68)
V81.hat[k] <- N.1*var(h69)
V82.hat[k] <- N.2*var(h70)
V83.hat[k] <- N.1*var(h71)
V84.hat[k] <- N.2*var(h72)

### AM ###      ### with g-weighths and Cross-Validation ####

rr.1 <- 1/(1-diag(w.s1.ad(L1L2.y1.n1.1.ADI[k,1],L1L2.y1.n1.1.ADI[k,2])))
rr.2 <- 1/(1-diag(w.s2.ad(L1L2.y1.n2.1.ADI[k,1],L1L2.y1.n2.1.ADI[k,2])))
rr.3 <- 1/(1-diag(w.s1.ad(L1L2.y2.n1.1.ADI[k,1],L1L2.y2.n1.1.ADI[k,2])))
rr.4 <- 1/(1-diag(w.s2.ad(L1L2.y2.n2.1.ADI[k,1],L1L2.y2.n2.1.ADI[k,2])))
rr.5 <- 1/(1-diag(w.s1.ad(L1L2.y3.n1.1.ADI[k,1],L1L2.y3.n1.1.ADI[k,2])))
rr.6 <- 1/(1-diag(w.s2.ad(L1L2.y3.n2.1.ADI[k,1],L1L2.y3.n2.1.ADI[k,2])))

rr.7 <- 1/(1-diag(w.s1.ad(L1L2.y1.n1.2.ADI[k,1],L1L2.y1.n1.2.ADI[k,2])))
rr.8 <- 1/(1-diag(w.s2.ad(L1L2.y1.n2.2.ADI[k,1],L1L2.y1.n2.2.ADI[k,2])))
rr.9 <- 1/(1-diag(w.s1.ad(L1L2.y2.n1.2.ADI[k,1],L1L2.y2.n1.2.ADI[k,2])))
rr.10 <- 1/(1-diag(w.s2.ad(L1L2.y2.n2.2.ADI[k,1],L1L2.y2.n2.2.ADI[k,2])))

```



```

rr.11 <- 1/(1-diag(w.s1.ad(L1L2.y3.n1.2.ADI[k,1],L1L2.y3.n1.2.ADI[k,2])))
rr.12 <- 1/(1-diag(w.s2.ad(L1L2.y3.n2.2.ADI[k,1],L1L2.y3.n2.2.ADI[k,2])))

W.s.1 <- w.s1.ad(L1L2.y1.n1.1.ADI[k,1],L1L2.y1.n1.1.ADI[k,2])-
diag(diag(w.s1.ad(L1L2.y1.n1.1.ADI[k,1],L1L2.y1.n1.1.ADI[k,2])))
W.s.2 <- w.s2.ad(L1L2.y1.n2.1.ADI[k,1],L1L2.y1.n2.1.ADI[k,2])-
diag(diag(w.s2.ad(L1L2.y1.n2.1.ADI[k,1],L1L2.y1.n2.1.ADI[k,2])))
W.s.3 <- w.s1.ad(L1L2.y2.n1.1.ADI[k,1],L1L2.y2.n1.1.ADI[k,2])-
diag(diag(w.s1.ad(L1L2.y2.n1.1.ADI[k,1],L1L2.y2.n1.1.ADI[k,2])))
W.s.4 <- w.s2.ad(L1L2.y2.n2.1.ADI[k,1],L1L2.y2.n2.1.ADI[k,2])-
diag(diag(w.s2.ad(L1L2.y2.n2.1.ADI[k,1],L1L2.y2.n2.1.ADI[k,2])))
W.s.5 <- w.s1.ad(L1L2.y3.n1.1.ADI[k,1],L1L2.y3.n1.1.ADI[k,2])-
diag(diag(w.s1.ad(L1L2.y3.n1.1.ADI[k,1],L1L2.y3.n1.1.ADI[k,2])))
W.s.6 <- w.s2.ad(L1L2.y3.n2.1.ADI[k,1],L1L2.y3.n2.1.ADI[k,2])-
diag(diag(w.s2.ad(L1L2.y3.n2.1.ADI[k,1],L1L2.y3.n2.1.ADI[k,2])))

W.s.7 <- w.s1.ad(L1L2.y1.n1.2.ADI[k,1],L1L2.y1.n1.2.ADI[k,2])-
diag(diag(w.s1.ad(L1L2.y1.n1.2.ADI[k,1],L1L2.y1.n1.2.ADI[k,2])))
W.s.8 <- w.s2.ad(L1L2.y1.n2.2.ADI[k,1],L1L2.y1.n2.2.ADI[k,2])-
diag(diag(w.s2.ad(L1L2.y1.n2.2.ADI[k,1],L1L2.y1.n2.2.ADI[k,2])))
W.s.9 <- w.s1.ad(L1L2.y2.n1.2.ADI[k,1],L1L2.y2.n1.2.ADI[k,2])-
diag(diag(w.s1.ad(L1L2.y2.n1.2.ADI[k,1],L1L2.y2.n1.2.ADI[k,2])))
W.s.10 <- w.s2.ad(L1L2.y2.n2.2.ADI[k,1],L1L2.y2.n2.2.ADI[k,2])-
diag(diag(w.s2.ad(L1L2.y2.n2.2.ADI[k,1],L1L2.y2.n2.2.ADI[k,2])))
W.s.11 <- w.s1.ad(L1L2.y3.n1.2.ADI[k,1],L1L2.y3.n1.2.ADI[k,2])-
diag(diag(w.s1.ad(L1L2.y3.n1.2.ADI[k,1],L1L2.y3.n1.2.ADI[k,2])))
W.s.12 <- w.s2.ad(L1L2.y3.n2.2.ADI[k,1],L1L2.y3.n2.2.ADI[k,2])-
diag(diag(w.s2.ad(L1L2.y3.n2.2.ADI[k,1],L1L2.y3.n2.2.ADI[k,2])))

tt.y1.1.1 <- matrix(rep(y.1.1[s1],n1),n1,n1)
tt.y1.2.1 <- matrix(rep(y.1.1[s2],n2),n2,n2)
tt.y2.1.1 <- matrix(rep(y.2.1[s1],n1),n1,n1)
tt.y2.2.1 <- matrix(rep(y.2.1[s2],n2),n2,n2)
tt.y3.1.1 <- matrix(rep(y.3.1[s1],n1),n1,n1)
tt.y3.2.1 <- matrix(rep(y.3.1[s2],n2),n2,n2)

tt.y1.1.2 <- matrix(rep(y.1.2[s1],n1),n1,n1)
tt.y1.2.2 <- matrix(rep(y.1.2[s2],n2),n2,n2)
tt.y2.1.2 <- matrix(rep(y.2.2[s1],n1),n1,n1)
tt.y2.2.2 <- matrix(rep(y.2.2[s2],n2),n2,n2)
tt.y3.1.2 <- matrix(rep(y.3.2[s1],n1),n1,n1)
tt.y3.2.2 <- matrix(rep(y.3.2[s2],n2),n2,n2)

W.t.y1.1.1 <- tt.y1.1.1-diag(diag(tt.y1.1.1))
W.t.y1.2.1 <- tt.y1.2.1-diag(diag(tt.y1.2.1))
W.t.y2.1.1 <- tt.y2.1.1-diag(diag(tt.y2.1.1))
W.t.y2.2.1 <- tt.y2.2.1-diag(diag(tt.y2.2.1))
W.t.y3.1.1 <- tt.y3.1.1-diag(diag(tt.y3.1.1))
W.t.y3.2.1 <- tt.y3.2.1-diag(diag(tt.y3.2.1))

W.t.y1.1.2 <- tt.y1.1.2-diag(diag(tt.y1.1.2))
W.t.y1.2.2 <- tt.y1.2.2-diag(diag(tt.y1.2.2))
W.t.y2.1.2 <- tt.y2.1.2-diag(diag(tt.y2.1.2))
W.t.y2.2.2 <- tt.y2.2.2-diag(diag(tt.y2.2.2))
W.t.y3.1.2 <- tt.y3.1.2-diag(diag(tt.y3.1.2))
W.t.y3.2.2 <- tt.y3.2.2-diag(diag(tt.y3.2.2))

h73 <- (y.1.1[s1] - rr.1*apply(W.s.1*W.t.y1.1.1,2,sum))
h74 <- (y.1.1[s2] - rr.2*apply(W.s.2*W.t.y1.2.1,2,sum))
h75 <- (y.2.1[s1] - rr.3*apply(W.s.3*W.t.y2.1.1,2,sum))
h76 <- (y.2.1[s2] - rr.4*apply(W.s.4*W.t.y2.2.1,2,sum))
h77 <- (y.3.1[s1] - rr.5*apply(W.s.5*W.t.y3.1.1,2,sum))
h78 <- (y.3.1[s2] - rr.6*apply(W.s.6*W.t.y3.2.1,2,sum))

```

```

h79 <- (y.1.2[s1] - rr.7*apply(W.s.7*W.t.y1.1.2,2,sum))
h80 <- (y.1.2[s2] - rr.8*apply(W.s.8*W.t.y1.2.2,2,sum))
h81 <- (y.2.2[s1] - rr.9*apply(W.s.9*W.t.y2.1.2,2,sum))
h82 <- (y.2.2[s2] - rr.10*apply(W.s.10*W.t.y2.2.2,2,sum))
h83 <- (y.3.2[s1] - rr.11*apply(W.s.11*W.t.y3.1.2,2,sum))
h84 <- (y.3.2[s2] - rr.12*apply(W.s.12*W.t.y3.2.2,2,sum))

h85 <- G.k.s1.ADI(L1L2.y1.n1.1.ADI[k,1],L1L2.y1.n1.1.ADI[k,2])*h73
h86 <- G.k.s2.ADI(L1L2.y1.n2.1.ADI[k,1],L1L2.y1.n2.1.ADI[k,2])*h74
h87 <- G.k.s1.ADI(L1L2.y2.n1.1.ADI[k,1],L1L2.y2.n1.1.ADI[k,2])*h75
h88 <- G.k.s2.ADI(L1L2.y2.n2.1.ADI[k,1],L1L2.y2.n2.1.ADI[k,2])*h76
h89 <- G.k.s1.ADI(L1L2.y3.n1.1.ADI[k,1],L1L2.y3.n1.1.ADI[k,2])*h77
h90 <- G.k.s2.ADI(L1L2.y3.n2.1.ADI[k,1],L1L2.y3.n2.1.ADI[k,2])*h78

h91 <- G.k.s1.ADI(L1L2.y1.n1.2.ADI[k,1],L1L2.y1.n1.2.ADI[k,2])*h79
h92 <- G.k.s2.ADI(L1L2.y1.n2.2.ADI[k,1],L1L2.y1.n2.2.ADI[k,2])*h80
h93 <- G.k.s1.ADI(L1L2.y2.n1.2.ADI[k,1],L1L2.y2.n1.2.ADI[k,2])*h81
h94 <- G.k.s2.ADI(L1L2.y2.n2.2.ADI[k,1],L1L2.y2.n2.2.ADI[k,2])*h82
h95 <- G.k.s1.ADI(L1L2.y3.n1.2.ADI[k,1],L1L2.y3.n1.2.ADI[k,2])*h83
h96 <- G.k.s2.ADI(L1L2.y3.n2.2.ADI[k,1],L1L2.y3.n2.2.ADI[k,2])*h84

```

```

V85.hat[k] <- N.1*var(h85)
V86.hat[k] <- N.2*var(h86)
V87.hat[k] <- N.1*var(h87)
V88.hat[k] <- N.2*var(h88)
V89.hat[k] <- N.1*var(h89)
V90.hat[k] <- N.2*var(h90)
V91.hat[k] <- N.1*var(h91)
V92.hat[k] <- N.2*var(h92)
V93.hat[k] <- N.1*var(h93)
V94.hat[k] <- N.2*var(h94)
V95.hat[k] <- N.1*var(h95)
V96.hat[k] <- N.2*var(h96)

```

##### LINEAR MODEL, without g-weights, with g-weights #####

```

L.k.s1 <- function(n)
{
  XX <- X_U.LIN[,1:n]
  t.X <- matrix(apply(XX,2,sum),1,)
  t.X.s1 <- (N/n1)*matrix(apply(XX[s1,],2,sum),1,)

  L.k.s1 <- 1+(t.X-t.X.s1)%%solve(t(XX[s1,])%*%Pi.inv.n1%*%XX[s1,])%*%t(XX[s1,])
  return(t(L.k.s1))
}

L.k.s2 <- function(n)
{
  XX <- X_U.LIN[,1:n]
  t.X <- matrix(apply(XX,2,sum),1,)
  t.X.s2 <- (N/n2)*matrix(apply(XX[s2,],2,sum),1,)

  L.k.s2 <- 1+(t.X-t.X.s2)%%solve(t(XX[s2,])%*%Pi.inv.n2%*%XX[s2,])%*%t(XX[s2,])
  return(t(L.k.s2))
}

h97 <- (y.1.1[s1] - mhat_s.y1.n1.LIN.1)
h98 <- (y.1.1[s2] - mhat_s.y1.n2.LIN.1)
h99 <- (y.2.1[s1] - mhat_s.y2.n1.LIN.1)
h100 <- (y.2.1[s2] - mhat_s.y2.n2.LIN.1)
h101 <- (y.3.1[s1] - mhat_s.y3.n1.LIN.1)
h102 <- (y.3.1[s2] - mhat_s.y3.n2.LIN.1)

```

```

h103 <- (y.1.2[s1] - mhat_s.y1.n1.LIN.2)
h104 <- (y.1.2[s2] - mhat_s.y1.n2.LIN.2)
h105 <- (y.2.2[s1] - mhat_s.y2.n1.LIN.2)
h106 <- (y.2.2[s2] - mhat_s.y2.n2.LIN.2)
h107 <- (y.3.2[s1] - mhat_s.y3.n1.LIN.2)
h108 <- (y.3.2[s2] - mhat_s.y3.n2.LIN.2)

```

```

V97.hat[k] <- N.1*var(h97)
V98.hat[k] <- N.2*var(h98)
V99.hat[k] <- N.1*var(h99)
V100.hat[k] <- N.2*var(h100)
V101.hat[k] <- N.1*var(h101)
V102.hat[k] <- N.2*var(h102)
V103.hat[k] <- N.1*var(h103)
V104.hat[k] <- N.2*var(h104)
V105.hat[k] <- N.1*var(h105)
V106.hat[k] <- N.2*var(h106)
V107.hat[k] <- N.1*var(h107)
V108.hat[k] <- N.2*var(h108)

```

```

h109 <- L.k.s1(10)*h97
h110 <- L.k.s2(10)*h98
h111 <- L.k.s1(10)*h99
h112 <- L.k.s2(10)*h100
h113 <- L.k.s1(10)*h101
h114 <- L.k.s2(10)*h102
h115 <- L.k.s1(10)*h103
h116 <- L.k.s2(10)*h104
h117 <- L.k.s1(10)*h105
h118 <- L.k.s2(10)*h106
h119 <- L.k.s1(10)*h107
h120 <- L.k.s2(10)*h108

```

```

V109.hat[k] <- N.1*var(h109)
V110.hat[k] <- N.2*var(h110)
V111.hat[k] <- N.1*var(h111)
V112.hat[k] <- N.2*var(h112)
V113.hat[k] <- N.1*var(h113)
V114.hat[k] <- N.2*var(h114)
V115.hat[k] <- N.1*var(h115)
V116.hat[k] <- N.2*var(h116)
V117.hat[k] <- N.1*var(h117)
V118.hat[k] <- N.2*var(h118)
V119.hat[k] <- N.1*var(h119)
V120.hat[k] <- N.2*var(h120)

```

```

#####
##### Estimation of population total #####
#####

```

```

t_hat.y1.n1.1[k] <- sum(mhat_U.y1.n1.1)+(N/n1)*sum((y.1.1[s1]-mhat_s.y1.n1.1))
t_hat.y1.n2.1[k] <- sum(mhat_U.y1.n2.1)+(N/n2)*sum((y.1.1[s2]-mhat_s.y1.n2.1))
t_hat.y2.n1.1[k] <- sum(mhat_U.y2.n1.1)+(N/n1)*sum((y.2.1[s1]-mhat_s.y2.n1.1))
t_hat.y2.n2.1[k] <- sum(mhat_U.y2.n2.1)+(N/n2)*sum((y.2.1[s2]-mhat_s.y2.n2.1))
t_hat.y3.n1.1[k] <- sum(mhat_U.y3.n1.1)+(N/n1)*sum((y.3.1[s1]-mhat_s.y3.n1.1))
t_hat.y3.n2.1[k] <- sum(mhat_U.y3.n2.1)+(N/n2)*sum((y.3.1[s2]-mhat_s.y3.n2.1))

```

```

t_hat.y1.n1.2[k] <- sum(mhat_U.y1.n1.2)+(N/n1)*sum((y.1.2[s1]-mhat_s.y1.n1.2))
t_hat.y1.n2.2[k] <- sum(mhat_U.y1.n2.2)+(N/n2)*sum((y.1.2[s2]-mhat_s.y1.n2.2))
t_hat.y2.n1.2[k] <- sum(mhat_U.y2.n1.2)+(N/n1)*sum((y.2.2[s1]-mhat_s.y2.n1.2))
t_hat.y2.n2.2[k] <- sum(mhat_U.y2.n2.2)+(N/n2)*sum((y.2.2[s2]-mhat_s.y2.n2.2))

```

## Anexos

```
t_hat.y3.n1.2[k] <- sum(mhat_U.y3.n1.2)+(N/n1)*sum((y.3.2[s1]-mhat_s.y3.n1.2))
t_hat.y3.n2.2[k] <- sum(mhat_U.y3.n2.2)+(N/n2)*sum((y.3.2[s2]-mhat_s.y3.n2.2))

t_hat.y1.n1.LIN.1[k] <- sum(mhat_U.y1.n1.LIN.1)+(N/n1)*sum((y.1.1[s1]-mhat_s.y1.n1.LIN.1))
t_hat.y1.n2.LIN.1[k] <- sum(mhat_U.y1.n2.LIN.1)+(N/n2)*sum((y.1.1[s2]-mhat_s.y1.n2.LIN.1))
t_hat.y2.n1.LIN.1[k] <- sum(mhat_U.y2.n1.LIN.1)+(N/n1)*sum((y.2.1[s1]-mhat_s.y2.n1.LIN.1))
t_hat.y2.n2.LIN.1[k] <- sum(mhat_U.y2.n2.LIN.1)+(N/n2)*sum((y.2.1[s2]-mhat_s.y2.n2.LIN.1))
t_hat.y3.n1.LIN.1[k] <- sum(mhat_U.y3.n1.LIN.1)+(N/n1)*sum((y.3.1[s1]-mhat_s.y3.n1.LIN.1))
t_hat.y3.n2.LIN.1[k] <- sum(mhat_U.y3.n2.LIN.1)+(N/n2)*sum((y.3.1[s2]-mhat_s.y3.n2.LIN.1))

t_hat.y1.n1.LIN.2[k] <- sum(mhat_U.y1.n1.LIN.2)+(N/n1)*sum((y.1.2[s1]-mhat_s.y1.n1.LIN.2))
t_hat.y1.n2.LIN.2[k] <- sum(mhat_U.y1.n2.LIN.2)+(N/n2)*sum((y.1.2[s2]-mhat_s.y1.n2.LIN.2))
t_hat.y2.n1.LIN.2[k] <- sum(mhat_U.y2.n1.LIN.2)+(N/n1)*sum((y.2.2[s1]-mhat_s.y2.n1.LIN.2))
t_hat.y2.n2.LIN.2[k] <- sum(mhat_U.y2.n2.LIN.2)+(N/n2)*sum((y.2.2[s2]-mhat_s.y2.n2.LIN.2))
t_hat.y3.n1.LIN.2[k] <- sum(mhat_U.y3.n1.LIN.2)+(N/n1)*sum((y.3.2[s1]-mhat_s.y3.n1.LIN.2))
t_hat.y3.n2.LIN.2[k] <- sum(mhat_U.y3.n2.LIN.2)+(N/n2)*sum((y.3.2[s2]-mhat_s.y3.n2.LIN.2))

t_hat.y1.n1.1.ADI[k] <- sum(mhat_U.y1.n1.1.ADI)+(N/n1)*sum((y.1.1[s1]-mhat_s.y1.n1.1.ADI))
t_hat.y1.n2.1.ADI[k] <- sum(mhat_U.y1.n2.1.ADI)+(N/n2)*sum((y.1.1[s2]-mhat_s.y1.n2.1.ADI))
t_hat.y2.n1.1.ADI[k] <- sum(mhat_U.y2.n1.1.ADI)+(N/n1)*sum((y.2.1[s1]-mhat_s.y2.n1.1.ADI))
t_hat.y2.n2.1.ADI[k] <- sum(mhat_U.y2.n2.1.ADI)+(N/n2)*sum((y.2.1[s2]-mhat_s.y2.n2.1.ADI))
t_hat.y3.n1.1.ADI[k] <- sum(mhat_U.y3.n1.1.ADI)+(N/n1)*sum((y.3.1[s1]-mhat_s.y3.n1.1.ADI))
t_hat.y3.n2.1.ADI[k] <- sum(mhat_U.y3.n2.1.ADI)+(N/n2)*sum((y.3.1[s2]-mhat_s.y3.n2.1.ADI))

t_hat.y1.n1.2.ADI[k] <- sum(mhat_U.y1.n1.2.ADI)+(N/n1)*sum((y.1.2[s1]-mhat_s.y1.n1.2.ADI))
t_hat.y1.n2.2.ADI[k] <- sum(mhat_U.y1.n2.2.ADI)+(N/n2)*sum((y.1.2[s2]-mhat_s.y1.n2.2.ADI))
t_hat.y2.n1.2.ADI[k] <- sum(mhat_U.y2.n1.2.ADI)+(N/n1)*sum((y.2.2[s1]-mhat_s.y2.n1.2.ADI))
t_hat.y2.n2.2.ADI[k] <- sum(mhat_U.y2.n2.2.ADI)+(N/n2)*sum((y.2.2[s2]-mhat_s.y2.n2.2.ADI))
t_hat.y3.n1.2.ADI[k] <- sum(mhat_U.y3.n1.2.ADI)+(N/n1)*sum((y.3.2[s1]-mhat_s.y3.n1.2.ADI))
t_hat.y3.n2.2.ADI[k] <- sum(mhat_U.y3.n2.2.ADI)+(N/n2)*sum((y.3.2[s2]-mhat_s.y3.n2.2.ADI))

#####
##### Confidence Interval 95% of t #####
##### Coverage Probability #####
#####

### VCM ###

#### Cross-Validation ####

L.y1.n1.1.CV <- t_hat.y1.n1.1[k]-1.96*sqrt(V1.hat[k])
U.y1.n1.1.CV <- t_hat.y1.n1.1[k]+1.96*sqrt(V1.hat[k])
L.y1.n2.1.CV <- t_hat.y1.n2.1[k]-1.96*sqrt(V2.hat[k])
U.y1.n2.1.CV <- t_hat.y1.n2.1[k]+1.96*sqrt(V2.hat[k])
L.y2.n1.1.CV <- t_hat.y2.n1.1[k]-1.96*sqrt(V3.hat[k])
U.y2.n1.1.CV <- t_hat.y2.n1.1[k]+1.96*sqrt(V3.hat[k])
L.y2.n2.1.CV <- t_hat.y2.n2.1[k]-1.96*sqrt(V4.hat[k])
U.y2.n2.1.CV <- t_hat.y2.n2.1[k]+1.96*sqrt(V4.hat[k])
L.y3.n1.1.CV <- t_hat.y3.n1.1[k]-1.96*sqrt(V5.hat[k])
U.y3.n1.1.CV <- t_hat.y3.n1.1[k]+1.96*sqrt(V5.hat[k])
L.y3.n2.1.CV <- t_hat.y3.n2.1[k]-1.96*sqrt(V6.hat[k])
U.y3.n2.1.CV <- t_hat.y3.n2.1[k]+1.96*sqrt(V6.hat[k])

L.y1.n1.2.CV <- t_hat.y1.n1.2[k]-1.96*sqrt(V7.hat[k])
U.y1.n1.2.CV <- t_hat.y1.n1.2[k]+1.96*sqrt(V7.hat[k])
L.y1.n2.2.CV <- t_hat.y1.n2.2[k]-1.96*sqrt(V8.hat[k])
U.y1.n2.2.CV <- t_hat.y1.n2.2[k]+1.96*sqrt(V8.hat[k])
L.y2.n1.2.CV <- t_hat.y2.n1.2[k]-1.96*sqrt(V9.hat[k])
U.y2.n1.2.CV <- t_hat.y2.n1.2[k]+1.96*sqrt(V9.hat[k])
L.y2.n2.2.CV <- t_hat.y2.n2.2[k]-1.96*sqrt(V10.hat[k])
U.y2.n2.2.CV <- t_hat.y2.n2.2[k]+1.96*sqrt(V10.hat[k])
L.y3.n1.2.CV <- t_hat.y3.n1.2[k]-1.96*sqrt(V11.hat[k])
U.y3.n1.2.CV <- t_hat.y3.n1.2[k]+1.96*sqrt(V11.hat[k])
```

## Anexos

---

```
L.y3.n2.2.CV <- t_hat.y3.n2.2[k]-1.96*sqrt(V12.hat[k])
U.y3.n2.2.CV <- t_hat.y3.n2.2[k]+1.96*sqrt(V12.hat[k])

Coverage.y1.n1.1.CV[k] <- ifelse(t.1.1>L.y1.n1.1.CV & t.1.1<U.y1.n1.1.CV,1,0)
Coverage.y1.n2.1.CV[k] <- ifelse(t.1.1>L.y1.n2.1.CV & t.1.1<U.y1.n2.1.CV,1,0)
Coverage.y2.n1.1.CV[k] <- ifelse(t.2.1>L.y2.n1.1.CV & t.2.1<U.y2.n1.1.CV,1,0)
Coverage.y2.n2.1.CV[k] <- ifelse(t.2.1>L.y2.n2.1.CV & t.2.1<U.y2.n2.1.CV,1,0)
Coverage.y3.n1.1.CV[k] <- ifelse(t.3.1>L.y3.n1.1.CV & t.3.1<U.y3.n1.1.CV,1,0)
Coverage.y3.n2.1.CV[k] <- ifelse(t.3.1>L.y3.n2.1.CV & t.3.1<U.y3.n2.1.CV,1,0)

Coverage.y1.n1.2.CV[k] <- ifelse(t.1.2>L.y1.n1.2.CV & t.1.2<U.y1.n1.2.CV,1,0)
Coverage.y1.n2.2.CV[k] <- ifelse(t.1.2>L.y1.n2.2.CV & t.1.2<U.y1.n2.2.CV,1,0)
Coverage.y2.n1.2.CV[k] <- ifelse(t.2.2>L.y2.n1.2.CV & t.2.2<U.y2.n1.2.CV,1,0)
Coverage.y2.n2.2.CV[k] <- ifelse(t.2.2>L.y2.n2.2.CV & t.2.2<U.y2.n2.2.CV,1,0)
Coverage.y3.n1.2.CV[k] <- ifelse(t.3.2>L.y3.n1.2.CV & t.3.2<U.y3.n1.2.CV,1,0)
Coverage.y3.n2.2.CV[k] <- ifelse(t.3.2>L.y3.n2.2.CV & t.3.2<U.y3.n2.2.CV,1,0)

der.y1.n1.1.CV[k] <- ifelse(t.1.1 > U.y1.n1.1.CV,1,0)
izq.y1.n1.1.CV[k] <- ifelse(t.1.1 < L.y1.n1.1.CV,1,0)
der.y1.n2.1.CV[k] <- ifelse(t.1.1 > U.y1.n2.1.CV,1,0)
izq.y1.n2.1.CV[k] <- ifelse(t.1.1 < L.y1.n2.1.CV,1,0)
der.y2.n1.1.CV[k] <- ifelse(t.2.1 > U.y2.n1.1.CV,1,0)
izq.y2.n1.1.CV[k] <- ifelse(t.2.1 < L.y2.n1.1.CV,1,0)
der.y2.n2.1.CV[k] <- ifelse(t.2.1 > U.y2.n2.1.CV,1,0)
izq.y2.n2.1.CV[k] <- ifelse(t.2.1 < L.y2.n2.1.CV,1,0)
der.y3.n1.1.CV[k] <- ifelse(t.3.1 > U.y3.n1.1.CV,1,0)
izq.y3.n1.1.CV[k] <- ifelse(t.3.1 < L.y3.n1.1.CV,1,0)
der.y3.n2.1.CV[k] <- ifelse(t.3.1 > U.y3.n2.1.CV,1,0)
izq.y3.n2.1.CV[k] <- ifelse(t.3.1 < L.y3.n2.1.CV,1,0)

der.y1.n1.2.CV[k] <- ifelse(t.1.2 > U.y1.n1.2.CV,1,0)
izq.y1.n1.2.CV[k] <- ifelse(t.1.2 < L.y1.n1.2.CV,1,0)
der.y1.n2.2.CV[k] <- ifelse(t.1.2 > U.y1.n2.2.CV,1,0)
izq.y1.n2.2.CV[k] <- ifelse(t.1.2 < L.y1.n2.2.CV,1,0)
der.y2.n1.2.CV[k] <- ifelse(t.2.2 > U.y2.n1.2.CV,1,0)
izq.y2.n1.2.CV[k] <- ifelse(t.2.2 < L.y2.n1.2.CV,1,0)
der.y2.n2.2.CV[k] <- ifelse(t.2.2 > U.y2.n2.2.CV,1,0)
izq.y2.n2.2.CV[k] <- ifelse(t.2.2 < L.y2.n2.2.CV,1,0)
der.y3.n1.2.CV[k] <- ifelse(t.3.2 > U.y3.n1.2.CV,1,0)
izq.y3.n1.2.CV[k] <- ifelse(t.3.2 < L.y3.n1.2.CV,1,0)
der.y3.n2.2.CV[k] <- ifelse(t.3.2 > U.y3.n2.2.CV,1,0)
izq.y3.n2.2.CV[k] <- ifelse(t.3.2 < L.y3.n2.2.CV,1,0)

##### g-weights #####

L.y1.n1.1.g_W <- t_hat.y1.n1.1[k]-1.96*sqrt(V37.hat[k])
U.y1.n1.1.g_W <- t_hat.y1.n1.1[k]+1.96*sqrt(V37.hat[k])
L.y1.n2.1.g_W <- t_hat.y1.n2.1[k]-1.96*sqrt(V38.hat[k])
U.y1.n2.1.g_W <- t_hat.y1.n2.1[k]+1.96*sqrt(V38.hat[k])
L.y2.n1.1.g_W <- t_hat.y2.n1.1[k]-1.96*sqrt(V39.hat[k])
U.y2.n1.1.g_W <- t_hat.y2.n1.1[k]+1.96*sqrt(V39.hat[k])
L.y2.n2.1.g_W <- t_hat.y2.n2.1[k]-1.96*sqrt(V40.hat[k])
U.y2.n2.1.g_W <- t_hat.y2.n2.1[k]+1.96*sqrt(V40.hat[k])
L.y3.n1.1.g_W <- t_hat.y3.n1.1[k]-1.96*sqrt(V41.hat[k])
U.y3.n1.1.g_W <- t_hat.y3.n1.1[k]+1.96*sqrt(V41.hat[k])
L.y3.n2.1.g_W <- t_hat.y3.n2.1[k]-1.96*sqrt(V42.hat[k])
U.y3.n2.1.g_W <- t_hat.y3.n2.1[k]+1.96*sqrt(V42.hat[k])

L.y1.n1.2.g_W <- t_hat.y1.n1.2[k]-1.96*sqrt(V43.hat[k])
U.y1.n1.2.g_W <- t_hat.y1.n1.2[k]+1.96*sqrt(V43.hat[k])
L.y1.n2.2.g_W <- t_hat.y1.n2.2[k]-1.96*sqrt(V44.hat[k])
U.y1.n2.2.g_W <- t_hat.y1.n2.2[k]+1.96*sqrt(V44.hat[k])
L.y2.n1.2.g_W <- t_hat.y2.n1.2[k]-1.96*sqrt(V45.hat[k])
```

## Anexos

---

```
U.y2.n1.2.g_W <- t_hat.y2.n1.2[k]+1.96*sqrt(V45.hat[k])
L.y2.n2.2.g_W <- t_hat.y2.n2.2[k]-1.96*sqrt(V46.hat[k])
U.y2.n2.2.g_W <- t_hat.y2.n2.2[k]+1.96*sqrt(V46.hat[k])
L.y3.n1.2.g_W <- t_hat.y3.n1.2[k]-1.96*sqrt(V47.hat[k])
U.y3.n1.2.g_W <- t_hat.y3.n1.2[k]+1.96*sqrt(V47.hat[k])
L.y3.n2.2.g_W <- t_hat.y3.n2.2[k]-1.96*sqrt(V48.hat[k])
U.y3.n2.2.g_W <- t_hat.y3.n2.2[k]+1.96*sqrt(V48.hat[k])

Coverage.y1.n1.1.g_W[k] <- ifelse(t.1.1>L.y1.n1.1.g_W & t.1.1<U.y1.n1.1.g_W ,1,0)
Coverage.y1.n2.1.g_W[k] <- ifelse(t.1.1>L.y1.n2.1.g_W & t.1.1<U.y1.n2.1.g_W ,1,0)
Coverage.y2.n1.1.g_W[k] <- ifelse(t.2.1>L.y2.n1.1.g_W & t.2.1<U.y2.n1.1.g_W ,1,0)
Coverage.y2.n2.1.g_W[k] <- ifelse(t.2.1>L.y2.n2.1.g_W & t.2.1<U.y2.n2.1.g_W ,1,0)
Coverage.y3.n1.1.g_W[k] <- ifelse(t.3.1>L.y3.n1.1.g_W & t.3.1<U.y3.n1.1.g_W ,1,0)
Coverage.y3.n2.1.g_W[k] <- ifelse(t.3.1>L.y3.n2.1.g_W & t.3.1<U.y3.n2.1.g_W ,1,0)

Coverage.y1.n1.2.g_W[k] <- ifelse(t.1.2>L.y1.n1.2.g_W & t.1.2<U.y1.n1.2.g_W ,1,0)
Coverage.y1.n2.2.g_W[k] <- ifelse(t.1.2>L.y1.n2.2.g_W & t.1.2<U.y1.n2.2.g_W ,1,0)
Coverage.y2.n1.2.g_W[k] <- ifelse(t.2.2>L.y2.n1.2.g_W & t.2.2<U.y2.n1.2.g_W ,1,0)
Coverage.y2.n2.2.g_W[k] <- ifelse(t.2.2>L.y2.n2.2.g_W & t.2.2<U.y2.n2.2.g_W ,1,0)
Coverage.y3.n1.2.g_W[k] <- ifelse(t.3.2>L.y3.n1.2.g_W & t.3.2<U.y3.n1.2.g_W ,1,0)
Coverage.y3.n2.2.g_W[k] <- ifelse(t.3.2>L.y3.n2.2.g_W & t.3.2<U.y3.n2.2.g_W ,1,0)

der.y1.n1.1.g_W[k] <- ifelse(t.1.1 > U.y1.n1.1.g_W,1,0)
izq.y1.n1.1.g_W[k] <- ifelse(t.1.1 < L.y1.n1.1.g_W,1,0)
der.y1.n2.1.g_W[k] <- ifelse(t.1.1 > U.y1.n2.1.g_W,1,0)
izq.y1.n2.1.g_W[k] <- ifelse(t.1.1 < L.y1.n2.1.g_W,1,0)
der.y2.n1.1.g_W[k] <- ifelse(t.2.1 > U.y2.n1.1.g_W,1,0)
izq.y2.n1.1.g_W[k] <- ifelse(t.2.1 < L.y2.n1.1.g_W,1,0)
der.y2.n2.1.g_W[k] <- ifelse(t.2.1 > U.y2.n2.1.g_W,1,0)
izq.y2.n2.1.g_W[k] <- ifelse(t.2.1 < L.y2.n2.1.g_W,1,0)
der.y3.n1.1.g_W[k] <- ifelse(t.3.1 > U.y3.n1.1.g_W,1,0)
izq.y3.n1.1.g_W[k] <- ifelse(t.3.1 < L.y3.n1.1.g_W,1,0)
der.y3.n2.1.g_W[k] <- ifelse(t.3.1 > U.y3.n2.1.g_W,1,0)
izq.y3.n2.1.g_W[k] <- ifelse(t.3.1 < L.y3.n2.1.g_W,1,0)

der.y1.n1.2.g_W[k] <- ifelse(t.1.2 > U.y1.n1.2.g_W,1,0)
izq.y1.n1.2.g_W[k] <- ifelse(t.1.2 < L.y1.n1.2.g_W,1,0)
der.y1.n2.2.g_W[k] <- ifelse(t.1.2 > U.y1.n2.2.g_W,1,0)
izq.y1.n2.2.g_W[k] <- ifelse(t.1.2 < L.y1.n2.2.g_W,1,0)
der.y2.n1.2.g_W[k] <- ifelse(t.2.2 > U.y2.n1.2.g_W,1,0)
izq.y2.n1.2.g_W[k] <- ifelse(t.2.2 < L.y2.n1.2.g_W,1,0)
der.y2.n2.2.g_W[k] <- ifelse(t.2.2 > U.y2.n2.2.g_W,1,0)
izq.y2.n2.2.g_W[k] <- ifelse(t.2.2 < L.y2.n2.2.g_W,1,0)
der.y3.n1.2.g_W[k] <- ifelse(t.3.2 > U.y3.n1.2.g_W,1,0)
izq.y3.n1.2.g_W[k] <- ifelse(t.3.2 < L.y3.n1.2.g_W,1,0)
der.y3.n2.2.g_W[k] <- ifelse(t.3.2 > U.y3.n2.2.g_W,1,0)
izq.y3.n2.2.g_W[k] <- ifelse(t.3.2 < L.y3.n2.2.g_W,1,0)

##### Classic #####

L.y1.n1.1 <- t_hat.y1.n1.1[k]-1.96*sqrt(V25.hat[k])
U.y1.n1.1 <- t_hat.y1.n1.1[k]+1.96*sqrt(V25.hat[k])
L.y1.n2.1 <- t_hat.y1.n2.1[k]-1.96*sqrt(V26.hat[k])
U.y1.n2.1 <- t_hat.y1.n2.1[k]+1.96*sqrt(V26.hat[k])
L.y2.n1.1 <- t_hat.y2.n1.1[k]-1.96*sqrt(V27.hat[k])
U.y2.n1.1 <- t_hat.y2.n1.1[k]+1.96*sqrt(V27.hat[k])
L.y2.n2.1 <- t_hat.y2.n2.1[k]-1.96*sqrt(V28.hat[k])
U.y2.n2.1 <- t_hat.y2.n2.1[k]+1.96*sqrt(V28.hat[k])
L.y3.n1.1 <- t_hat.y3.n1.1[k]-1.96*sqrt(V29.hat[k])
U.y3.n1.1 <- t_hat.y3.n1.1[k]+1.96*sqrt(V29.hat[k])
L.y3.n2.1 <- t_hat.y3.n2.1[k]-1.96*sqrt(V30.hat[k])
```

## Anexos

---

```
U.y3.n2.1 <- t_hat.y3.n2.1[k]+1.96*sqrt(V30.hat[k])

L.y1.n1.2 <- t_hat.y1.n1.2[k]-1.96*sqrt(V31.hat[k])
U.y1.n1.2 <- t_hat.y1.n1.2[k]+1.96*sqrt(V31.hat[k])
L.y1.n2.2 <- t_hat.y1.n2.2[k]-1.96*sqrt(V32.hat[k])
U.y1.n2.2 <- t_hat.y1.n2.2[k]+1.96*sqrt(V32.hat[k])
L.y2.n1.2 <- t_hat.y2.n1.2[k]-1.96*sqrt(V33.hat[k])
U.y2.n1.2 <- t_hat.y2.n1.2[k]+1.96*sqrt(V33.hat[k])
L.y2.n2.2 <- t_hat.y2.n2.2[k]-1.96*sqrt(V34.hat[k])
U.y2.n2.2 <- t_hat.y2.n2.2[k]+1.96*sqrt(V34.hat[k])
L.y3.n1.2 <- t_hat.y3.n1.2[k]-1.96*sqrt(V35.hat[k])
U.y3.n1.2 <- t_hat.y3.n1.2[k]+1.96*sqrt(V35.hat[k])
L.y3.n2.2 <- t_hat.y3.n2.2[k]-1.96*sqrt(V36.hat[k])
U.y3.n2.2 <- t_hat.y3.n2.2[k]+1.96*sqrt(V36.hat[k])

Coverage.y1.n1.1[k] <- ifelse(t.1.1>L.y1.n1.1 & t.1.1<U.y1.n1.1,1,0)
Coverage.y1.n2.1[k] <- ifelse(t.1.1>L.y1.n2.1 & t.1.1<U.y1.n2.1,1,0)
Coverage.y2.n1.1[k] <- ifelse(t.2.1>L.y2.n1.1 & t.2.1<U.y2.n1.1,1,0)
Coverage.y2.n2.1[k] <- ifelse(t.2.1>L.y2.n2.1 & t.2.1<U.y2.n2.1,1,0)
Coverage.y3.n1.1[k] <- ifelse(t.3.1>L.y3.n1.1 & t.3.1<U.y3.n1.1,1,0)
Coverage.y3.n2.1[k] <- ifelse(t.3.1>L.y3.n2.1 & t.3.1<U.y3.n2.1,1,0)

Coverage.y1.n1.2[k] <- ifelse(t.1.2>L.y1.n1.2 & t.1.2<U.y1.n1.2,1,0)
Coverage.y1.n2.2[k] <- ifelse(t.1.2>L.y1.n2.2 & t.1.2<U.y1.n2.2,1,0)
Coverage.y2.n1.2[k] <- ifelse(t.2.2>L.y2.n1.2 & t.2.2<U.y2.n1.2,1,0)
Coverage.y2.n2.2[k] <- ifelse(t.2.2>L.y2.n2.2 & t.2.2<U.y2.n2.2,1,0)
Coverage.y3.n1.2[k] <- ifelse(t.3.2>L.y3.n1.2 & t.3.2<U.y3.n1.2,1,0)
Coverage.y3.n2.2[k] <- ifelse(t.3.2>L.y3.n2.2 & t.3.2<U.y3.n2.2,1,0)

der.y1.n1.1[k] <- ifelse(t.1.1 > U.y1.n1.1,1,0)
izq.y1.n1.1[k] <- ifelse(t.1.1 < L.y1.n1.1,1,0)
der.y1.n2.1[k] <- ifelse(t.1.1 > U.y1.n2.1,1,0)
izq.y1.n2.1[k] <- ifelse(t.1.1 < L.y1.n2.1,1,0)
der.y2.n1.1[k] <- ifelse(t.2.1 > U.y2.n1.1,1,0)
izq.y2.n1.1[k] <- ifelse(t.2.1 < L.y2.n1.1,1,0)
der.y2.n2.1[k] <- ifelse(t.2.1 > U.y2.n2.1,1,0)
izq.y2.n2.1[k] <- ifelse(t.2.1 < L.y2.n2.1,1,0)
der.y3.n1.1[k] <- ifelse(t.3.1 > U.y3.n1.1,1,0)
izq.y3.n1.1[k] <- ifelse(t.3.1 < L.y3.n1.1,1,0)
der.y3.n2.1[k] <- ifelse(t.3.1 > U.y3.n2.1,1,0)
izq.y3.n2.1[k] <- ifelse(t.3.1 < L.y3.n2.1,1,0)

der.y1.n1.2[k] <- ifelse(t.1.2 > U.y1.n1.2,1,0)
izq.y1.n1.2[k] <- ifelse(t.1.2 < L.y1.n1.2,1,0)
der.y1.n2.2[k] <- ifelse(t.1.2 > U.y1.n2.2,1,0)
izq.y1.n2.2[k] <- ifelse(t.1.2 < L.y1.n2.2,1,0)
der.y2.n1.2[k] <- ifelse(t.2.2 > U.y2.n1.2,1,0)
izq.y2.n1.2[k] <- ifelse(t.2.2 < L.y2.n1.2,1,0)
der.y2.n2.2[k] <- ifelse(t.2.2 > U.y2.n2.2,1,0)
izq.y2.n2.2[k] <- ifelse(t.2.2 < L.y2.n2.2,1,0)
der.y3.n1.2[k] <- ifelse(t.3.2 > U.y3.n1.2,1,0)
izq.y3.n1.2[k] <- ifelse(t.3.2 < L.y3.n1.2,1,0)
der.y3.n2.2[k] <- ifelse(t.3.2 > U.y3.n2.2,1,0)
izq.y3.n2.2[k] <- ifelse(t.3.2 < L.y3.n2.2,1,0)

##### g-weights and cross-validation #####

L.y1.n1.1.CV.g_W <- t_hat.y1.n1.1[k]-1.96*sqrt(V73.hat[k])
U.y1.n1.1.CV.g_W <- t_hat.y1.n1.1[k]+1.96*sqrt(V73.hat[k])
L.y1.n2.1.CV.g_W <- t_hat.y1.n2.1[k]-1.96*sqrt(V74.hat[k])
U.y1.n2.1.CV.g_W <- t_hat.y1.n2.1[k]+1.96*sqrt(V74.hat[k])
L.y2.n1.1.CV.g_W <- t_hat.y2.n1.1[k]-1.96*sqrt(V75.hat[k])
U.y2.n1.1.CV.g_W <- t_hat.y2.n1.1[k]+1.96*sqrt(V75.hat[k])
```

```

L.y2.n2.1.CV.g_W <- t_hat.y2.n2.1[k]-1.96*sqrt(V76.hat[k])
U.y2.n2.1.CV.g_W <- t_hat.y2.n2.1[k]+1.96*sqrt(V76.hat[k])
L.y3.n1.1.CV.g_W <- t_hat.y3.n1.1[k]-1.96*sqrt(V77.hat[k])
U.y3.n1.1.CV.g_W <- t_hat.y3.n1.1[k]+1.96*sqrt(V77.hat[k])
L.y3.n2.1.CV.g_W <- t_hat.y3.n2.1[k]-1.96*sqrt(V78.hat[k])
U.y3.n2.1.CV.g_W <- t_hat.y3.n2.1[k]+1.96*sqrt(V78.hat[k])

L.y1.n1.2.CV.g_W <- t_hat.y1.n1.2[k]-1.96*sqrt(V79.hat[k])
U.y1.n1.2.CV.g_W <- t_hat.y1.n1.2[k]+1.96*sqrt(V79.hat[k])
L.y1.n2.2.CV.g_W <- t_hat.y1.n2.2[k]-1.96*sqrt(V80.hat[k])
U.y1.n2.2.CV.g_W <- t_hat.y1.n2.2[k]+1.96*sqrt(V80.hat[k])
L.y2.n1.2.CV.g_W <- t_hat.y2.n1.2[k]-1.96*sqrt(V81.hat[k])
U.y2.n1.2.CV.g_W <- t_hat.y2.n1.2[k]+1.96*sqrt(V81.hat[k])
L.y2.n2.2.CV.g_W <- t_hat.y2.n2.2[k]-1.96*sqrt(V82.hat[k])
U.y2.n2.2.CV.g_W <- t_hat.y2.n2.2[k]+1.96*sqrt(V82.hat[k])
L.y3.n1.2.CV.g_W <- t_hat.y3.n1.2[k]-1.96*sqrt(V83.hat[k])
U.y3.n1.2.CV.g_W <- t_hat.y3.n1.2[k]+1.96*sqrt(V83.hat[k])
L.y3.n2.2.CV.g_W <- t_hat.y3.n2.2[k]-1.96*sqrt(V84.hat[k])
U.y3.n2.2.CV.g_W <- t_hat.y3.n2.2[k]+1.96*sqrt(V84.hat[k])

Coverage.y1.n1.1.CV.g_W[k] <- ifelse(t.1.1>L.y1.n1.1.CV.g_W & t.1.1<U.y1.n1.1.CV.g_W,1,0)
Coverage.y1.n2.1.CV.g_W[k] <- ifelse(t.1.1>L.y1.n2.1.CV.g_W & t.1.1<U.y1.n2.1.CV.g_W,1,0)
Coverage.y2.n1.1.CV.g_W[k] <- ifelse(t.2.1>L.y2.n1.1.CV.g_W & t.2.1<U.y2.n1.1.CV.g_W,1,0)
Coverage.y2.n2.1.CV.g_W[k] <- ifelse(t.2.1>L.y2.n2.1.CV.g_W & t.2.1<U.y2.n2.1.CV.g_W,1,0)
Coverage.y3.n1.1.CV.g_W[k] <- ifelse(t.3.1>L.y3.n1.1.CV.g_W & t.3.1<U.y3.n1.1.CV.g_W,1,0)
Coverage.y3.n2.1.CV.g_W[k] <- ifelse(t.3.1>L.y3.n2.1.CV.g_W & t.3.1<U.y3.n2.1.CV.g_W,1,0)

Coverage.y1.n1.2.CV.g_W[k] <- ifelse(t.1.2>L.y1.n1.2.CV.g_W & t.1.2<U.y1.n1.2.CV.g_W,1,0)
Coverage.y1.n2.2.CV.g_W[k] <- ifelse(t.1.2>L.y1.n2.2.CV.g_W & t.1.2<U.y1.n2.2.CV.g_W,1,0)
Coverage.y2.n1.2.CV.g_W[k] <- ifelse(t.2.2>L.y2.n1.2.CV.g_W & t.2.2<U.y2.n1.2.CV.g_W,1,0)
Coverage.y2.n2.2.CV.g_W[k] <- ifelse(t.2.2>L.y2.n2.2.CV.g_W & t.2.2<U.y2.n2.2.CV.g_W,1,0)
Coverage.y3.n1.2.CV.g_W[k] <- ifelse(t.3.2>L.y3.n1.2.CV.g_W & t.3.2<U.y3.n1.2.CV.g_W,1,0)
Coverage.y3.n2.2.CV.g_W[k] <- ifelse(t.3.2>L.y3.n2.2.CV.g_W & t.3.2<U.y3.n2.2.CV.g_W,1,0)

der.y1.n1.1.CV.g_W[k] <- ifelse(t.1.1 > U.y1.n1.1.CV.g_W,1,0)
izq.y1.n1.1.CV.g_W[k] <- ifelse(t.1.1 < L.y1.n1.1.CV.g_W,1,0)
der.y1.n2.1.CV.g_W[k] <- ifelse(t.1.1 > U.y1.n2.1.CV.g_W,1,0)
izq.y1.n2.1.CV.g_W[k] <- ifelse(t.1.1 < L.y1.n2.1.CV.g_W,1,0)
der.y2.n1.1.CV.g_W[k] <- ifelse(t.2.1 > U.y2.n1.1.CV.g_W,1,0)
izq.y2.n1.1.CV.g_W[k] <- ifelse(t.2.1 < L.y2.n1.1.CV.g_W,1,0)
der.y2.n2.1.CV.g_W[k] <- ifelse(t.2.1 > U.y2.n2.1.CV.g_W,1,0)
izq.y2.n2.1.CV.g_W[k] <- ifelse(t.2.1 < L.y2.n2.1.CV.g_W,1,0)
der.y3.n1.1.CV.g_W[k] <- ifelse(t.3.1 > U.y3.n1.1.CV.g_W,1,0)
izq.y3.n1.1.CV.g_W[k] <- ifelse(t.3.1 < L.y3.n1.1.CV.g_W,1,0)
der.y3.n2.1.CV.g_W[k] <- ifelse(t.3.1 > U.y3.n2.1.CV.g_W,1,0)
izq.y3.n2.1.CV.g_W[k] <- ifelse(t.3.1 < L.y3.n2.1.CV.g_W,1,0)

der.y1.n1.2.CV.g_W[k] <- ifelse(t.1.2 > U.y1.n1.2.CV.g_W,1,0)
izq.y1.n1.2.CV.g_W[k] <- ifelse(t.1.2 < L.y1.n1.2.CV.g_W,1,0)
der.y1.n2.2.CV.g_W[k] <- ifelse(t.1.2 > U.y1.n2.2.CV.g_W,1,0)
izq.y1.n2.2.CV.g_W[k] <- ifelse(t.1.2 < L.y1.n2.2.CV.g_W,1,0)
der.y2.n1.2.CV.g_W[k] <- ifelse(t.2.2 > U.y2.n1.2.CV.g_W,1,0)
izq.y2.n1.2.CV.g_W[k] <- ifelse(t.2.2 < L.y2.n1.2.CV.g_W,1,0)
der.y2.n2.2.CV.g_W[k] <- ifelse(t.2.2 > U.y2.n2.2.CV.g_W,1,0)
izq.y2.n2.2.CV.g_W[k] <- ifelse(t.2.2 < L.y2.n2.2.CV.g_W,1,0)
der.y3.n1.2.CV.g_W[k] <- ifelse(t.3.2 > U.y3.n1.2.CV.g_W,1,0)
izq.y3.n1.2.CV.g_W[k] <- ifelse(t.3.2 < L.y3.n1.2.CV.g_W,1,0)
der.y3.n2.2.CV.g_W[k] <- ifelse(t.3.2 > U.y3.n2.2.CV.g_W,1,0)
izq.y3.n2.2.CV.g_W[k] <- ifelse(t.3.2 < L.y3.n2.2.CV.g_W,1,0)

```

### ADDITIVE MODELS ###



#### Cross-Validation ####

```
L.y1.n1.1.CV <- t_hat.y1.n1.1.ADI[k]-1.96*sqrt(V13.hat[k])
U.y1.n1.1.CV <- t_hat.y1.n1.1.ADI[k]+1.96*sqrt(V13.hat[k])
L.y1.n2.1.CV <- t_hat.y1.n2.1.ADI[k]-1.96*sqrt(V14.hat[k])
U.y1.n2.1.CV <- t_hat.y1.n2.1.ADI[k]+1.96*sqrt(V14.hat[k])
L.y2.n1.1.CV <- t_hat.y2.n1.1.ADI[k]-1.96*sqrt(V15.hat[k])
U.y2.n1.1.CV <- t_hat.y2.n1.1.ADI[k]+1.96*sqrt(V15.hat[k])
L.y2.n2.1.CV <- t_hat.y2.n2.1.ADI[k]-1.96*sqrt(V16.hat[k])
U.y2.n2.1.CV <- t_hat.y2.n2.1.ADI[k]+1.96*sqrt(V16.hat[k])
L.y3.n1.1.CV <- t_hat.y3.n1.1.ADI[k]-1.96*sqrt(V17.hat[k])
U.y3.n1.1.CV <- t_hat.y3.n1.1.ADI[k]+1.96*sqrt(V17.hat[k])
L.y3.n2.1.CV <- t_hat.y3.n2.1.ADI[k]-1.96*sqrt(V18.hat[k])
U.y3.n2.1.CV <- t_hat.y3.n2.1.ADI[k]+1.96*sqrt(V18.hat[k])
```

```
L.y1.n1.2.CV <- t_hat.y1.n1.2.ADI[k]-1.96*sqrt(V19.hat[k])
U.y1.n1.2.CV <- t_hat.y1.n1.2.ADI[k]+1.96*sqrt(V19.hat[k])
L.y1.n2.2.CV <- t_hat.y1.n2.2.ADI[k]-1.96*sqrt(V20.hat[k])
U.y1.n2.2.CV <- t_hat.y1.n2.2.ADI[k]+1.96*sqrt(V20.hat[k])
L.y2.n1.2.CV <- t_hat.y2.n1.2.ADI[k]-1.96*sqrt(V21.hat[k])
U.y2.n1.2.CV <- t_hat.y2.n1.2.ADI[k]+1.96*sqrt(V21.hat[k])
L.y2.n2.2.CV <- t_hat.y2.n2.2.ADI[k]-1.96*sqrt(V22.hat[k])
U.y2.n2.2.CV <- t_hat.y2.n2.2.ADI[k]+1.96*sqrt(V22.hat[k])
L.y3.n1.2.CV <- t_hat.y3.n1.2.ADI[k]-1.96*sqrt(V23.hat[k])
U.y3.n1.2.CV <- t_hat.y3.n1.2.ADI[k]+1.96*sqrt(V23.hat[k])
L.y3.n2.2.CV <- t_hat.y3.n2.2.ADI[k]-1.96*sqrt(V24.hat[k])
U.y3.n2.2.CV <- t_hat.y3.n2.2.ADI[k]+1.96*sqrt(V24.hat[k])
```

```
Coverage.y1.n1.1.CV.ADI[k] <- ifelse(t.1.1>L.y1.n1.1.CV & t.1.1<U.y1.n1.1.CV,1,0)
Coverage.y1.n2.1.CV.ADI[k] <- ifelse(t.1.1>L.y1.n2.1.CV & t.1.1<U.y1.n2.1.CV,1,0)
Coverage.y2.n1.1.CV.ADI[k] <- ifelse(t.2.1>L.y2.n1.1.CV & t.2.1<U.y2.n1.1.CV,1,0)
Coverage.y2.n2.1.CV.ADI[k] <- ifelse(t.2.1>L.y2.n2.1.CV & t.2.1<U.y2.n2.1.CV,1,0)
Coverage.y3.n1.1.CV.ADI[k] <- ifelse(t.3.1>L.y3.n1.1.CV & t.3.1<U.y3.n1.1.CV,1,0)
Coverage.y3.n2.1.CV.ADI[k] <- ifelse(t.3.1>L.y3.n2.1.CV & t.3.1<U.y3.n2.1.CV,1,0)
```

```
Coverage.y1.n1.2.CV.ADI[k] <- ifelse(t.1.2>L.y1.n1.2.CV & t.1.2<U.y1.n1.2.CV,1,0)
Coverage.y1.n2.2.CV.ADI[k] <- ifelse(t.1.2>L.y1.n2.2.CV & t.1.2<U.y1.n2.2.CV,1,0)
Coverage.y2.n1.2.CV.ADI[k] <- ifelse(t.2.2>L.y2.n1.2.CV & t.2.2<U.y2.n1.2.CV,1,0)
Coverage.y2.n2.2.CV.ADI[k] <- ifelse(t.2.2>L.y2.n2.2.CV & t.2.2<U.y2.n2.2.CV,1,0)
Coverage.y3.n1.2.CV.ADI[k] <- ifelse(t.3.2>L.y3.n1.2.CV & t.3.2<U.y3.n1.2.CV,1,0)
Coverage.y3.n2.2.CV.ADI[k] <- ifelse(t.3.2>L.y3.n2.2.CV & t.3.2<U.y3.n2.2.CV,1,0)
```

```
der.y1.n1.1.CV.ADI[k] <- ifelse(t.1.1 > U.y1.n1.1.CV,1,0)
izq.y1.n1.1.CV.ADI[k] <- ifelse(t.1.1 < L.y1.n1.1.CV,1,0)
der.y1.n2.1.CV.ADI[k] <- ifelse(t.1.1 > U.y1.n2.1.CV,1,0)
izq.y1.n2.1.CV.ADI[k] <- ifelse(t.1.1 < L.y1.n2.1.CV,1,0)
der.y2.n1.1.CV.ADI[k] <- ifelse(t.2.1 > U.y2.n1.1.CV,1,0)
izq.y2.n1.1.CV.ADI[k] <- ifelse(t.2.1 < L.y2.n1.1.CV,1,0)
der.y2.n2.1.CV.ADI[k] <- ifelse(t.2.1 > U.y2.n2.1.CV,1,0)
izq.y2.n2.1.CV.ADI[k] <- ifelse(t.2.1 < L.y2.n2.1.CV,1,0)
der.y3.n1.1.CV.ADI[k] <- ifelse(t.3.1 > U.y3.n1.1.CV,1,0)
izq.y3.n1.1.CV.ADI[k] <- ifelse(t.3.1 < L.y3.n1.1.CV,1,0)
der.y3.n2.1.CV.ADI[k] <- ifelse(t.3.1 > U.y3.n2.1.CV,1,0)
izq.y3.n2.1.CV.ADI[k] <- ifelse(t.3.1 < L.y3.n2.1.CV,1,0)
```

```
der.y1.n1.2.CV.ADI[k] <- ifelse(t.1.2 > U.y1.n1.2.CV,1,0)
izq.y1.n1.2.CV.ADI[k] <- ifelse(t.1.2 < L.y1.n1.2.CV,1,0)
der.y1.n2.2.CV.ADI[k] <- ifelse(t.1.2 > U.y1.n2.2.CV,1,0)
izq.y1.n2.2.CV.ADI[k] <- ifelse(t.1.2 < L.y1.n2.2.CV,1,0)
der.y2.n1.2.CV.ADI[k] <- ifelse(t.2.2 > U.y2.n1.2.CV,1,0)
izq.y2.n1.2.CV.ADI[k] <- ifelse(t.2.2 < L.y2.n1.2.CV,1,0)
der.y2.n2.2.CV.ADI[k] <- ifelse(t.2.2 > U.y2.n2.2.CV,1,0)
izq.y2.n2.2.CV.ADI[k] <- ifelse(t.2.2 < L.y2.n2.2.CV,1,0)
```

## Anexos

```
der.y3.n1.2.CV.ADI[k] <- ifelse(t.3.2 > U.y3.n1.2.CV,1,0)
izq.y3.n1.2.CV.ADI[k] <- ifelse(t.3.2 < L.y3.n1.2.CV,1,0)
der.y3.n2.2.CV.ADI[k] <- ifelse(t.3.2 > U.y3.n2.2.CV,1,0)
izq.y3.n2.2.CV.ADI[k] <- ifelse(t.3.2 < L.y3.n2.2.CV,1,0)

##### g-weights #####

L.y1.n1.1.g_W <- t_hat.y1.n1.1.ADI[k]-1.96*sqrt(V61.hat[k])
U.y1.n1.1.g_W <- t_hat.y1.n1.1.ADI[k]+1.96*sqrt(V61.hat[k])
L.y1.n2.1.g_W <- t_hat.y1.n2.1.ADI[k]-1.96*sqrt(V62.hat[k])
U.y1.n2.1.g_W <- t_hat.y1.n2.1.ADI[k]+1.96*sqrt(V62.hat[k])
L.y2.n1.1.g_W <- t_hat.y2.n1.1.ADI[k]-1.96*sqrt(V63.hat[k])
U.y2.n1.1.g_W <- t_hat.y2.n1.1.ADI[k]+1.96*sqrt(V63.hat[k])
L.y2.n2.1.g_W <- t_hat.y2.n2.1.ADI[k]-1.96*sqrt(V64.hat[k])
U.y2.n2.1.g_W <- t_hat.y2.n2.1.ADI[k]+1.96*sqrt(V64.hat[k])
L.y3.n1.1.g_W <- t_hat.y3.n1.1.ADI[k]-1.96*sqrt(V65.hat[k])
U.y3.n1.1.g_W <- t_hat.y3.n1.1.ADI[k]+1.96*sqrt(V65.hat[k])
L.y3.n2.1.g_W <- t_hat.y3.n2.1.ADI[k]-1.96*sqrt(V66.hat[k])
U.y3.n2.1.g_W <- t_hat.y3.n2.1.ADI[k]+1.96*sqrt(V66.hat[k])

L.y1.n1.2.g_W <- t_hat.y1.n1.2.ADI[k]-1.96*sqrt(V67.hat[k])
U.y1.n1.2.g_W <- t_hat.y1.n1.2.ADI[k]+1.96*sqrt(V67.hat[k])
L.y1.n2.2.g_W <- t_hat.y1.n2.2.ADI[k]-1.96*sqrt(V68.hat[k])
U.y1.n2.2.g_W <- t_hat.y1.n2.2.ADI[k]+1.96*sqrt(V68.hat[k])
L.y2.n1.2.g_W <- t_hat.y2.n1.2.ADI[k]-1.96*sqrt(V69.hat[k])
U.y2.n1.2.g_W <- t_hat.y2.n1.2.ADI[k]+1.96*sqrt(V69.hat[k])
L.y2.n2.2.g_W <- t_hat.y2.n2.2.ADI[k]-1.96*sqrt(V70.hat[k])
U.y2.n2.2.g_W <- t_hat.y2.n2.2.ADI[k]+1.96*sqrt(V70.hat[k])
L.y3.n1.2.g_W <- t_hat.y3.n1.2.ADI[k]-1.96*sqrt(V71.hat[k])
U.y3.n1.2.g_W <- t_hat.y3.n1.2.ADI[k]+1.96*sqrt(V71.hat[k])
L.y3.n2.2.g_W <- t_hat.y3.n2.2.ADI[k]-1.96*sqrt(V72.hat[k])
U.y3.n2.2.g_W <- t_hat.y3.n2.2.ADI[k]+1.96*sqrt(V72.hat[k])

Coverage.y1.n1.1.g_W.ADI[k] <- ifelse(t.1.1>L.y1.n1.1.g_W & t.1.1<U.y1.n1.1.g_W ,1,0)
Coverage.y1.n2.1.g_W.ADI[k] <- ifelse(t.1.1>L.y1.n2.1.g_W & t.1.1<U.y1.n2.1.g_W ,1,0)
Coverage.y2.n1.1.g_W.ADI[k] <- ifelse(t.2.1>L.y2.n1.1.g_W & t.2.1<U.y2.n1.1.g_W ,1,0)
Coverage.y2.n2.1.g_W.ADI[k] <- ifelse(t.2.1>L.y2.n2.1.g_W & t.2.1<U.y2.n2.1.g_W ,1,0)
Coverage.y3.n1.1.g_W.ADI[k] <- ifelse(t.3.1>L.y3.n1.1.g_W & t.3.1<U.y3.n1.1.g_W ,1,0)
Coverage.y3.n2.1.g_W.ADI[k] <- ifelse(t.3.1>L.y3.n2.1.g_W & t.3.1<U.y3.n2.1.g_W ,1,0)

Coverage.y1.n1.2.g_W.ADI[k] <- ifelse(t.1.2>L.y1.n1.2.g_W & t.1.2<U.y1.n1.2.g_W ,1,0)
Coverage.y1.n2.2.g_W.ADI[k] <- ifelse(t.1.2>L.y1.n2.2.g_W & t.1.2<U.y1.n2.2.g_W ,1,0)
Coverage.y2.n1.2.g_W.ADI[k] <- ifelse(t.2.2>L.y2.n1.2.g_W & t.2.2<U.y2.n1.2.g_W ,1,0)
Coverage.y2.n2.2.g_W.ADI[k] <- ifelse(t.2.2>L.y2.n2.2.g_W & t.2.2<U.y2.n2.2.g_W ,1,0)
Coverage.y3.n1.2.g_W.ADI[k] <- ifelse(t.3.2>L.y3.n1.2.g_W & t.3.2<U.y3.n1.2.g_W ,1,0)
Coverage.y3.n2.2.g_W.ADI[k] <- ifelse(t.3.2>L.y3.n2.2.g_W & t.3.2<U.y3.n2.2.g_W ,1,0)

der.y1.n1.1.g_W.ADI[k] <- ifelse(t.1.1 > U.y1.n1.1.g_W,1,0)
izq.y1.n1.1.g_W.ADI[k] <- ifelse(t.1.1 < L.y1.n1.1.g_W,1,0)
der.y1.n2.1.g_W.ADI[k] <- ifelse(t.1.1 > U.y1.n2.1.g_W,1,0)
izq.y1.n2.1.g_W.ADI[k] <- ifelse(t.1.1 < L.y1.n2.1.g_W,1,0)
der.y2.n1.1.g_W.ADI[k] <- ifelse(t.2.1 > U.y2.n1.1.g_W,1,0)
izq.y2.n1.1.g_W.ADI[k] <- ifelse(t.2.1 < L.y2.n1.1.g_W,1,0)
der.y2.n2.1.g_W.ADI[k] <- ifelse(t.2.1 > U.y2.n2.1.g_W,1,0)
izq.y2.n2.1.g_W.ADI[k] <- ifelse(t.2.1 < L.y2.n2.1.g_W,1,0)
der.y3.n1.1.g_W.ADI[k] <- ifelse(t.3.1 > U.y3.n1.1.g_W,1,0)
izq.y3.n1.1.g_W.ADI[k] <- ifelse(t.3.1 < L.y3.n1.1.g_W,1,0)
der.y3.n2.1.g_W.ADI[k] <- ifelse(t.3.1 > U.y3.n2.1.g_W,1,0)
izq.y3.n2.1.g_W.ADI[k] <- ifelse(t.3.1 < L.y3.n2.1.g_W,1,0)
```

## Anexos

---

```
der.y1.n1.2.g_W.ADI[k] <- ifelse(t.1.2 > U.y1.n1.2.g_W,1,0)
izq.y1.n1.2.g_W.ADI[k] <- ifelse(t.1.2 < L.y1.n1.2.g_W,1,0)
der.y1.n2.2.g_W.ADI[k] <- ifelse(t.1.2 > U.y1.n2.2.g_W,1,0)
izq.y1.n2.2.g_W.ADI[k] <- ifelse(t.1.2 < L.y1.n2.2.g_W,1,0)
der.y2.n1.2.g_W.ADI[k] <- ifelse(t.2.2 > U.y2.n1.2.g_W,1,0)
izq.y2.n1.2.g_W.ADI[k] <- ifelse(t.2.2 < L.y2.n1.2.g_W,1,0)
der.y2.n2.2.g_W.ADI[k] <- ifelse(t.2.2 > U.y2.n2.2.g_W,1,0)
izq.y2.n2.2.g_W.ADI[k] <- ifelse(t.2.2 < L.y2.n2.2.g_W,1,0)
der.y3.n1.2.g_W.ADI[k] <- ifelse(t.3.2 > U.y3.n1.2.g_W,1,0)
izq.y3.n1.2.g_W.ADI[k] <- ifelse(t.3.2 < L.y3.n1.2.g_W,1,0)
der.y3.n2.2.g_W.ADI[k] <- ifelse(t.3.2 > U.y3.n2.2.g_W,1,0)
izq.y3.n2.2.g_W.ADI[k] <- ifelse(t.3.2 < L.y3.n2.2.g_W,1,0)

##### Classic #####

L.y1.n1.1 <- t_hat.y1.n1.1.ADI[k]-1.96*sqrt(V49.hat[k])
U.y1.n1.1 <- t_hat.y1.n1.1.ADI[k]+1.96*sqrt(V49.hat[k])
L.y1.n2.1 <- t_hat.y1.n2.1.ADI[k]-1.96*sqrt(V50.hat[k])
U.y1.n2.1 <- t_hat.y1.n2.1.ADI[k]+1.96*sqrt(V50.hat[k])
L.y2.n1.1 <- t_hat.y2.n1.1.ADI[k]-1.96*sqrt(V51.hat[k])
U.y2.n1.1 <- t_hat.y2.n1.1.ADI[k]+1.96*sqrt(V51.hat[k])
L.y2.n2.1 <- t_hat.y2.n2.1.ADI[k]-1.96*sqrt(V52.hat[k])
U.y2.n2.1 <- t_hat.y2.n2.1.ADI[k]+1.96*sqrt(V52.hat[k])
L.y3.n1.1 <- t_hat.y3.n1.1.ADI[k]-1.96*sqrt(V53.hat[k])
U.y3.n1.1 <- t_hat.y3.n1.1.ADI[k]+1.96*sqrt(V53.hat[k])
L.y3.n2.1 <- t_hat.y3.n2.1.ADI[k]-1.96*sqrt(V54.hat[k])
U.y3.n2.1 <- t_hat.y3.n2.1.ADI[k]+1.96*sqrt(V54.hat[k])

L.y1.n1.2 <- t_hat.y1.n1.2.ADI[k]-1.96*sqrt(V55.hat[k])
U.y1.n1.2 <- t_hat.y1.n1.2.ADI[k]+1.96*sqrt(V55.hat[k])
L.y1.n2.2 <- t_hat.y1.n2.2.ADI[k]-1.96*sqrt(V56.hat[k])
U.y1.n2.2 <- t_hat.y1.n2.2.ADI[k]+1.96*sqrt(V56.hat[k])
L.y2.n1.2 <- t_hat.y2.n1.2.ADI[k]-1.96*sqrt(V57.hat[k])
U.y2.n1.2 <- t_hat.y2.n1.2.ADI[k]+1.96*sqrt(V57.hat[k])
L.y2.n2.2 <- t_hat.y2.n2.2.ADI[k]-1.96*sqrt(V58.hat[k])
U.y2.n2.2 <- t_hat.y2.n2.2.ADI[k]+1.96*sqrt(V58.hat[k])
L.y3.n1.2 <- t_hat.y3.n1.2.ADI[k]-1.96*sqrt(V59.hat[k])
U.y3.n1.2 <- t_hat.y3.n1.2.ADI[k]+1.96*sqrt(V59.hat[k])
L.y3.n2.2 <- t_hat.y3.n2.2.ADI[k]-1.96*sqrt(V60.hat[k])
U.y3.n2.2 <- t_hat.y3.n2.2.ADI[k]+1.96*sqrt(V60.hat[k])

Coverage.y1.n1.1.ADI[k] <- ifelse(t.1.1>L.y1.n1.1 & t.1.1<U.y1.n1.1,1,0)
Coverage.y1.n2.1.ADI[k] <- ifelse(t.1.1>L.y1.n2.1 & t.1.1<U.y1.n2.1,1,0)
Coverage.y2.n1.1.ADI[k] <- ifelse(t.2.1>L.y2.n1.1 & t.2.1<U.y2.n1.1,1,0)
Coverage.y2.n2.1.ADI[k] <- ifelse(t.2.1>L.y2.n2.1 & t.2.1<U.y2.n2.1,1,0)
Coverage.y3.n1.1.ADI[k] <- ifelse(t.3.1>L.y3.n1.1 & t.3.1<U.y3.n1.1,1,0)
Coverage.y3.n2.1.ADI[k] <- ifelse(t.3.1>L.y3.n2.1 & t.3.1<U.y3.n2.1,1,0)

Coverage.y1.n1.2.ADI[k] <- ifelse(t.1.2>L.y1.n1.2 & t.1.2<U.y1.n1.2,1,0)
Coverage.y1.n2.2.ADI[k] <- ifelse(t.1.2>L.y1.n2.2 & t.1.2<U.y1.n2.2,1,0)
Coverage.y2.n1.2.ADI[k] <- ifelse(t.2.2>L.y2.n1.2 & t.2.2<U.y2.n1.2,1,0)
Coverage.y2.n2.2.ADI[k] <- ifelse(t.2.2>L.y2.n2.2 & t.2.2<U.y2.n2.2,1,0)
Coverage.y3.n1.2.ADI[k] <- ifelse(t.3.2>L.y3.n1.2 & t.3.2<U.y3.n1.2,1,0)
Coverage.y3.n2.2.ADI[k] <- ifelse(t.3.2>L.y3.n2.2 & t.3.2<U.y3.n2.2,1,0)

der.y1.n1.1.ADI[k] <- ifelse(t.1.1 > U.y1.n1.1,1,0)
izq.y1.n1.1.ADI[k] <- ifelse(t.1.1 < L.y1.n1.1,1,0)
der.y1.n2.1.ADI[k] <- ifelse(t.1.1 > U.y1.n2.1,1,0)
izq.y1.n2.1.ADI[k] <- ifelse(t.1.1 < L.y1.n2.1,1,0)
der.y2.n1.1.ADI[k] <- ifelse(t.2.1 > U.y2.n1.1,1,0)
izq.y2.n1.1.ADI[k] <- ifelse(t.2.1 < L.y2.n1.1,1,0)
der.y2.n2.1.ADI[k] <- ifelse(t.2.1 > U.y2.n2.1,1,0)
izq.y2.n2.1.ADI[k] <- ifelse(t.2.1 < L.y2.n2.1,1,0)
der.y3.n1.1.ADI[k] <- ifelse(t.3.1 > U.y3.n1.1,1,0)
```

## Anexos

---

```
izq.y3.n1.1.ADI[k] <- ifelse(t.3.1 < L.y3.n1.1,1,0)
der.y3.n2.1.ADI[k] <- ifelse(t.3.1 > U.y3.n2.1,1,0)
izq.y3.n2.1.ADI[k] <- ifelse(t.3.1 < L.y3.n2.1,1,0)

der.y1.n1.2.ADI[k] <- ifelse(t.1.2 > U.y1.n1.2,1,0)
izq.y1.n1.2.ADI[k] <- ifelse(t.1.2 < L.y1.n1.2,1,0)
der.y1.n2.2.ADI[k] <- ifelse(t.1.2 > U.y1.n2.2,1,0)
izq.y1.n2.2.ADI[k] <- ifelse(t.1.2 < L.y1.n2.2,1,0)
der.y2.n1.2.ADI[k] <- ifelse(t.2.2 > U.y2.n1.2,1,0)
izq.y2.n1.2.ADI[k] <- ifelse(t.2.2 < L.y2.n1.2,1,0)
der.y2.n2.2.ADI[k] <- ifelse(t.2.2 > U.y2.n2.2,1,0)
izq.y2.n2.2.ADI[k] <- ifelse(t.2.2 < L.y2.n2.2,1,0)
der.y3.n1.2.ADI[k] <- ifelse(t.3.2 > U.y3.n1.2,1,0)
izq.y3.n1.2.ADI[k] <- ifelse(t.3.2 < L.y3.n1.2,1,0)
der.y3.n2.2.ADI[k] <- ifelse(t.3.2 > U.y3.n2.2,1,0)
izq.y3.n2.2.ADI[k] <- ifelse(t.3.2 < L.y3.n2.2,1,0)

##### g-weights and cross-validation #####

L.y1.n1.1.CV.g_W <- t_hat.y1.n1.1.ADI[k]-1.96*sqrt(V85.hat[k])
U.y1.n1.1.CV.g_W <- t_hat.y1.n1.1.ADI[k]+1.96*sqrt(V85.hat[k])
L.y1.n2.1.CV.g_W <- t_hat.y1.n2.1.ADI[k]-1.96*sqrt(V86.hat[k])
U.y1.n2.1.CV.g_W <- t_hat.y1.n2.1.ADI[k]+1.96*sqrt(V86.hat[k])
L.y2.n1.1.CV.g_W <- t_hat.y2.n1.1.ADI[k]-1.96*sqrt(V87.hat[k])
U.y2.n1.1.CV.g_W <- t_hat.y2.n1.1.ADI[k]+1.96*sqrt(V87.hat[k])
L.y2.n2.1.CV.g_W <- t_hat.y2.n2.1.ADI[k]-1.96*sqrt(V88.hat[k])
U.y2.n2.1.CV.g_W <- t_hat.y2.n2.1.ADI[k]+1.96*sqrt(V88.hat[k])
L.y3.n1.1.CV.g_W <- t_hat.y3.n1.1.ADI[k]-1.96*sqrt(V89.hat[k])
U.y3.n1.1.CV.g_W <- t_hat.y3.n1.1.ADI[k]+1.96*sqrt(V89.hat[k])
L.y3.n2.1.CV.g_W <- t_hat.y3.n2.1.ADI[k]-1.96*sqrt(V90.hat[k])
U.y3.n2.1.CV.g_W <- t_hat.y3.n2.1.ADI[k]+1.96*sqrt(V90.hat[k])

L.y1.n1.2.CV.g_W <- t_hat.y1.n1.2.ADI[k]-1.96*sqrt(V91.hat[k])
U.y1.n1.2.CV.g_W <- t_hat.y1.n1.2.ADI[k]+1.96*sqrt(V91.hat[k])
L.y1.n2.2.CV.g_W <- t_hat.y1.n2.2.ADI[k]-1.96*sqrt(V92.hat[k])
U.y1.n2.2.CV.g_W <- t_hat.y1.n2.2.ADI[k]+1.96*sqrt(V92.hat[k])
L.y2.n1.2.CV.g_W <- t_hat.y2.n1.2.ADI[k]-1.96*sqrt(V93.hat[k])
U.y2.n1.2.CV.g_W <- t_hat.y2.n1.2.ADI[k]+1.96*sqrt(V93.hat[k])
L.y2.n2.2.CV.g_W <- t_hat.y2.n2.2.ADI[k]-1.96*sqrt(V94.hat[k])
U.y2.n2.2.CV.g_W <- t_hat.y2.n2.2.ADI[k]+1.96*sqrt(V94.hat[k])
L.y3.n1.2.CV.g_W <- t_hat.y3.n1.2.ADI[k]-1.96*sqrt(V95.hat[k])
U.y3.n1.2.CV.g_W <- t_hat.y3.n1.2.ADI[k]+1.96*sqrt(V95.hat[k])
L.y3.n2.2.CV.g_W <- t_hat.y3.n2.2.ADI[k]-1.96*sqrt(V96.hat[k])
U.y3.n2.2.CV.g_W <- t_hat.y3.n2.2.ADI[k]+1.96*sqrt(V96.hat[k])

Coverage.y1.n1.1.CV.g_W.ADI[k] <- ifelse(t.1.1>L.y1.n1.1.CV.g_W & t.1.1<U.y1.n1.1.CV.g_W,1,0)
Coverage.y1.n2.1.CV.g_W.ADI[k] <- ifelse(t.1.1>L.y1.n2.1.CV.g_W & t.1.1<U.y1.n2.1.CV.g_W,1,0)
Coverage.y2.n1.1.CV.g_W.ADI[k] <- ifelse(t.2.1>L.y2.n1.1.CV.g_W & t.2.1<U.y2.n1.1.CV.g_W,1,0)
Coverage.y2.n2.1.CV.g_W.ADI[k] <- ifelse(t.2.1>L.y2.n2.1.CV.g_W & t.2.1<U.y2.n2.1.CV.g_W,1,0)
Coverage.y3.n1.1.CV.g_W.ADI[k] <- ifelse(t.3.1>L.y3.n1.1.CV.g_W & t.3.1<U.y3.n1.1.CV.g_W,1,0)
Coverage.y3.n2.1.CV.g_W.ADI[k] <- ifelse(t.3.1>L.y3.n2.1.CV.g_W & t.3.1<U.y3.n2.1.CV.g_W,1,0)

Coverage.y1.n1.2.CV.g_W.ADI[k] <- ifelse(t.1.2>L.y1.n1.2.CV.g_W & t.1.2<U.y1.n1.2.CV.g_W,1,0)
Coverage.y1.n2.2.CV.g_W.ADI[k] <- ifelse(t.1.2>L.y1.n2.2.CV.g_W & t.1.2<U.y1.n2.2.CV.g_W,1,0)
Coverage.y2.n1.2.CV.g_W.ADI[k] <- ifelse(t.2.2>L.y2.n1.2.CV.g_W & t.2.2<U.y2.n1.2.CV.g_W,1,0)
Coverage.y2.n2.2.CV.g_W.ADI[k] <- ifelse(t.2.2>L.y2.n2.2.CV.g_W & t.2.2<U.y2.n2.2.CV.g_W,1,0)
Coverage.y3.n1.2.CV.g_W.ADI[k] <- ifelse(t.3.2>L.y3.n1.2.CV.g_W & t.3.2<U.y3.n1.2.CV.g_W,1,0)
Coverage.y3.n2.2.CV.g_W.ADI[k] <- ifelse(t.3.2>L.y3.n2.2.CV.g_W & t.3.2<U.y3.n2.2.CV.g_W,1,0)

der.y1.n1.1.CV.g_W.ADI[k] <- ifelse(t.1.1 > U.y1.n1.1.CV.g_W,1,0)
izq.y1.n1.1.CV.g_W.ADI[k] <- ifelse(t.1.1 < L.y1.n1.1.CV.g_W,1,0)
der.y1.n2.1.CV.g_W.ADI[k] <- ifelse(t.1.1 > U.y1.n2.1.CV.g_W,1,0)
izq.y1.n2.1.CV.g_W.ADI[k] <- ifelse(t.1.1 < L.y1.n2.1.CV.g_W,1,0)
```

## Anexos

```
der.y2.n1.1.CV.g_W.ADI[k] <- ifelse(t.2.1 > U.y2.n1.1.CV.g_W,1,0)
izq.y2.n1.1.CV.g_W.ADI[k] <- ifelse(t.2.1 < L.y2.n1.1.CV.g_W,1,0)
der.y2.n2.1.CV.g_W.ADI[k] <- ifelse(t.2.1 > U.y2.n2.1.CV.g_W,1,0)
izq.y2.n2.1.CV.g_W.ADI[k] <- ifelse(t.2.1 < L.y2.n2.1.CV.g_W,1,0)
der.y3.n1.1.CV.g_W.ADI[k] <- ifelse(t.3.1 > U.y3.n1.1.CV.g_W,1,0)
izq.y3.n1.1.CV.g_W.ADI[k] <- ifelse(t.3.1 < L.y3.n1.1.CV.g_W,1,0)
der.y3.n2.1.CV.g_W.ADI[k] <- ifelse(t.3.1 > U.y3.n2.1.CV.g_W,1,0)
izq.y3.n2.1.CV.g_W.ADI[k] <- ifelse(t.3.1 < L.y3.n2.1.CV.g_W,1,0)

der.y1.n1.2.CV.g_W.ADI[k] <- ifelse(t.1.2 > U.y1.n1.2.CV.g_W,1,0)
izq.y1.n1.2.CV.g_W.ADI[k] <- ifelse(t.1.2 < L.y1.n1.2.CV.g_W,1,0)
der.y1.n2.2.CV.g_W.ADI[k] <- ifelse(t.1.2 > U.y1.n2.2.CV.g_W,1,0)
izq.y1.n2.2.CV.g_W.ADI[k] <- ifelse(t.1.2 < L.y1.n2.2.CV.g_W,1,0)
der.y2.n1.2.CV.g_W.ADI[k] <- ifelse(t.2.2 > U.y2.n1.2.CV.g_W,1,0)
izq.y2.n1.2.CV.g_W.ADI[k] <- ifelse(t.2.2 < L.y2.n1.2.CV.g_W,1,0)
der.y2.n2.2.CV.g_W.ADI[k] <- ifelse(t.2.2 > U.y2.n2.2.CV.g_W,1,0)
izq.y2.n2.2.CV.g_W.ADI[k] <- ifelse(t.2.2 < L.y2.n2.2.CV.g_W,1,0)
der.y3.n1.2.CV.g_W.ADI[k] <- ifelse(t.3.2 > U.y3.n1.2.CV.g_W,1,0)
izq.y3.n1.2.CV.g_W.ADI[k] <- ifelse(t.3.2 < L.y3.n1.2.CV.g_W,1,0)
der.y3.n2.2.CV.g_W.ADI[k] <- ifelse(t.3.2 > U.y3.n2.2.CV.g_W,1,0)
izq.y3.n2.2.CV.g_W.ADI[k] <- ifelse(t.3.2 < L.y3.n2.2.CV.g_W,1,0)

### LINEAR MODEL ###

##### g-weights #####

L.y1.n1.1.g_W <- t_hat.y1.n1.LIN.1[k]-1.96*sqrt(V109.hat[k])
U.y1.n1.1.g_W <- t_hat.y1.n1.LIN.1[k]+1.96*sqrt(V109.hat[k])
L.y1.n2.1.g_W <- t_hat.y1.n2.LIN.1[k]-1.96*sqrt(V110.hat[k])
U.y1.n2.1.g_W <- t_hat.y1.n2.LIN.1[k]+1.96*sqrt(V110.hat[k])
L.y2.n1.1.g_W <- t_hat.y2.n1.LIN.1[k]-1.96*sqrt(V111.hat[k])
U.y2.n1.1.g_W <- t_hat.y2.n1.LIN.1[k]+1.96*sqrt(V111.hat[k])
L.y2.n2.1.g_W <- t_hat.y2.n2.LIN.1[k]-1.96*sqrt(V112.hat[k])
U.y2.n2.1.g_W <- t_hat.y2.n2.LIN.1[k]+1.96*sqrt(V112.hat[k])
L.y3.n1.1.g_W <- t_hat.y3.n1.LIN.1[k]-1.96*sqrt(V113.hat[k])
U.y3.n1.1.g_W <- t_hat.y3.n1.LIN.1[k]+1.96*sqrt(V113.hat[k])
L.y3.n2.1.g_W <- t_hat.y3.n2.LIN.1[k]-1.96*sqrt(V114.hat[k])
U.y3.n2.1.g_W <- t_hat.y3.n2.LIN.1[k]+1.96*sqrt(V114.hat[k])

L.y1.n1.2.g_W <- t_hat.y1.n1.LIN.2[k]-1.96*sqrt(V115.hat[k])
U.y1.n1.2.g_W <- t_hat.y1.n1.LIN.2[k]+1.96*sqrt(V115.hat[k])
L.y1.n2.2.g_W <- t_hat.y1.n2.LIN.2[k]-1.96*sqrt(V116.hat[k])
U.y1.n2.2.g_W <- t_hat.y1.n2.LIN.2[k]+1.96*sqrt(V116.hat[k])
L.y2.n1.2.g_W <- t_hat.y2.n1.LIN.2[k]-1.96*sqrt(V117.hat[k])
U.y2.n1.2.g_W <- t_hat.y2.n1.LIN.2[k]+1.96*sqrt(V117.hat[k])
L.y2.n2.2.g_W <- t_hat.y2.n2.LIN.2[k]-1.96*sqrt(V118.hat[k])
U.y2.n2.2.g_W <- t_hat.y2.n2.LIN.2[k]+1.96*sqrt(V118.hat[k])
L.y3.n1.2.g_W <- t_hat.y3.n1.LIN.2[k]-1.96*sqrt(V119.hat[k])
U.y3.n1.2.g_W <- t_hat.y3.n1.LIN.2[k]+1.96*sqrt(V119.hat[k])
L.y3.n2.2.g_W <- t_hat.y3.n2.LIN.2[k]-1.96*sqrt(V120.hat[k])
U.y3.n2.2.g_W <- t_hat.y3.n2.LIN.2[k]+1.96*sqrt(V120.hat[k])

Coverage.y1.n1.1.g_W.LIN[k] <- ifelse(t.1.1>L.y1.n1.1.g_W & t.1.1<U.y1.n1.1.g_W ,1,0)
Coverage.y1.n2.1.g_W.LIN[k] <- ifelse(t.1.1>L.y1.n2.1.g_W & t.1.1<U.y1.n2.1.g_W ,1,0)
Coverage.y2.n1.1.g_W.LIN[k] <- ifelse(t.2.1>L.y2.n1.1.g_W & t.2.1<U.y2.n1.1.g_W ,1,0)
Coverage.y2.n2.1.g_W.LIN[k] <- ifelse(t.2.1>L.y2.n2.1.g_W & t.2.1<U.y2.n2.1.g_W ,1,0)
Coverage.y3.n1.1.g_W.LIN[k] <- ifelse(t.3.1>L.y3.n1.1.g_W & t.3.1<U.y3.n1.1.g_W ,1,0)
Coverage.y3.n2.1.g_W.LIN[k] <- ifelse(t.3.1>L.y3.n2.1.g_W & t.3.1<U.y3.n2.1.g_W ,1,0)

Coverage.y1.n1.2.g_W.LIN[k] <- ifelse(t.1.2>L.y1.n1.2.g_W & t.1.2<U.y1.n1.2.g_W ,1,0)
Coverage.y1.n2.2.g_W.LIN[k] <- ifelse(t.1.2>L.y1.n2.2.g_W & t.1.2<U.y1.n2.2.g_W ,1,0)
```

## Anexos

---

```
Coverage.y2.n1.2.g_W.LIN[k] <- ifelse(t.2.2>L.y2.n1.2.g_W & t.2.2<U.y2.n1.2.g_W ,1,0)
Coverage.y2.n2.2.g_W.LIN[k] <- ifelse(t.2.2>L.y2.n2.2.g_W & t.2.2<U.y2.n2.2.g_W ,1,0)
Coverage.y3.n1.2.g_W.LIN[k] <- ifelse(t.3.2>L.y3.n1.2.g_W & t.3.2<U.y3.n1.2.g_W ,1,0)
Coverage.y3.n2.2.g_W.LIN[k] <- ifelse(t.3.2>L.y3.n2.2.g_W & t.3.2<U.y3.n2.2.g_W ,1,0)
```

```
der.y1.n1.1.g_W.LIN[k] <- ifelse(t.1.1 > U.y1.n1.1.g_W,1,0)
izq.y1.n1.1.g_W.LIN[k] <- ifelse(t.1.1 < L.y1.n1.1.g_W,1,0)
der.y1.n2.1.g_W.LIN[k] <- ifelse(t.1.1 > U.y1.n2.1.g_W,1,0)
izq.y1.n2.1.g_W.LIN[k] <- ifelse(t.1.1 < L.y1.n2.1.g_W,1,0)
der.y2.n1.1.g_W.LIN[k] <- ifelse(t.2.1 > U.y2.n1.1.g_W,1,0)
izq.y2.n1.1.g_W.LIN[k] <- ifelse(t.2.1 < L.y2.n1.1.g_W,1,0)
der.y2.n2.1.g_W.LIN[k] <- ifelse(t.2.1 > U.y2.n2.1.g_W,1,0)
izq.y2.n2.1.g_W.LIN[k] <- ifelse(t.2.1 < L.y2.n2.1.g_W,1,0)
der.y3.n1.1.g_W.LIN[k] <- ifelse(t.3.1 > U.y3.n1.1.g_W,1,0)
izq.y3.n1.1.g_W.LIN[k] <- ifelse(t.3.1 < L.y3.n1.1.g_W,1,0)
der.y3.n2.1.g_W.LIN[k] <- ifelse(t.3.1 > U.y3.n2.1.g_W,1,0)
izq.y3.n2.1.g_W.LIN[k] <- ifelse(t.3.1 < L.y3.n2.1.g_W,1,0)
```

```
der.y1.n1.2.g_W.LIN[k] <- ifelse(t.1.2 > U.y1.n1.2.g_W,1,0)
izq.y1.n1.2.g_W.LIN[k] <- ifelse(t.1.2 < L.y1.n1.2.g_W,1,0)
der.y1.n2.2.g_W.LIN[k] <- ifelse(t.1.2 > U.y1.n2.2.g_W,1,0)
izq.y1.n2.2.g_W.LIN[k] <- ifelse(t.1.2 < L.y1.n2.2.g_W,1,0)
der.y2.n1.2.g_W.LIN[k] <- ifelse(t.2.2 > U.y2.n1.2.g_W,1,0)
izq.y2.n1.2.g_W.LIN[k] <- ifelse(t.2.2 < L.y2.n1.2.g_W,1,0)
der.y2.n2.2.g_W.LIN[k] <- ifelse(t.2.2 > U.y2.n2.2.g_W,1,0)
izq.y2.n2.2.g_W.LIN[k] <- ifelse(t.2.2 < L.y2.n2.2.g_W,1,0)
der.y3.n1.2.g_W.LIN[k] <- ifelse(t.3.2 > U.y3.n1.2.g_W,1,0)
izq.y3.n1.2.g_W.LIN[k] <- ifelse(t.3.2 < L.y3.n1.2.g_W,1,0)
der.y3.n2.2.g_W.LIN[k] <- ifelse(t.3.2 > U.y3.n2.2.g_W,1,0)
izq.y3.n2.2.g_W.LIN[k] <- ifelse(t.3.2 < L.y3.n2.2.g_W,1,0)
```

##### Classic #####

```
L.y1.n1.1 <- t_hat.y1.n1.LIN.1[k]-1.96*sqrt(V97.hat[k])
U.y1.n1.1 <- t_hat.y1.n1.LIN.1[k]+1.96*sqrt(V97.hat[k])
L.y1.n2.1 <- t_hat.y1.n2.LIN.1[k]-1.96*sqrt(V98.hat[k])
U.y1.n2.1 <- t_hat.y1.n2.LIN.1[k]+1.96*sqrt(V98.hat[k])
L.y2.n1.1 <- t_hat.y2.n1.LIN.1[k]-1.96*sqrt(V99.hat[k])
U.y2.n1.1 <- t_hat.y2.n1.LIN.1[k]+1.96*sqrt(V99.hat[k])
L.y2.n2.1 <- t_hat.y2.n2.LIN.1[k]-1.96*sqrt(V100.hat[k])
U.y2.n2.1 <- t_hat.y2.n2.LIN.1[k]+1.96*sqrt(V100.hat[k])
L.y3.n1.1 <- t_hat.y3.n1.LIN.1[k]-1.96*sqrt(V101.hat[k])
U.y3.n1.1 <- t_hat.y3.n1.LIN.1[k]+1.96*sqrt(V101.hat[k])
L.y3.n2.1 <- t_hat.y3.n2.LIN.1[k]-1.96*sqrt(V102.hat[k])
U.y3.n2.1 <- t_hat.y3.n2.LIN.1[k]+1.96*sqrt(V102.hat[k])
```

```
L.y1.n1.2 <- t_hat.y1.n1.LIN.2[k]-1.96*sqrt(V103.hat[k])
U.y1.n1.2 <- t_hat.y1.n1.LIN.2[k]+1.96*sqrt(V103.hat[k])
L.y1.n2.2 <- t_hat.y1.n2.LIN.2[k]-1.96*sqrt(V104.hat[k])
U.y1.n2.2 <- t_hat.y1.n2.LIN.2[k]+1.96*sqrt(V104.hat[k])
L.y2.n1.2 <- t_hat.y2.n1.LIN.2[k]-1.96*sqrt(V105.hat[k])
U.y2.n1.2 <- t_hat.y2.n1.LIN.2[k]+1.96*sqrt(V105.hat[k])
L.y2.n2.2 <- t_hat.y2.n2.LIN.2[k]-1.96*sqrt(V106.hat[k])
U.y2.n2.2 <- t_hat.y2.n2.LIN.2[k]+1.96*sqrt(V106.hat[k])
L.y3.n1.2 <- t_hat.y3.n1.LIN.2[k]-1.96*sqrt(V107.hat[k])
U.y3.n1.2 <- t_hat.y3.n1.LIN.2[k]+1.96*sqrt(V107.hat[k])
L.y3.n2.2 <- t_hat.y3.n2.LIN.2[k]-1.96*sqrt(V108.hat[k])
U.y3.n2.2 <- t_hat.y3.n2.LIN.2[k]+1.96*sqrt(V108.hat[k])
```

```
Coverage.y1.n1.1.LIN[k] <- ifelse(t.1.1>L.y1.n1.1 & t.1.1<U.y1.n1.1,1,0)
Coverage.y1.n2.1.LIN[k] <- ifelse(t.1.1>L.y1.n2.1 & t.1.1<U.y1.n2.1,1,0)
Coverage.y2.n1.1.LIN[k] <- ifelse(t.2.1>L.y2.n1.1 & t.2.1<U.y2.n1.1,1,0)
Coverage.y2.n2.1.LIN[k] <- ifelse(t.2.1>L.y2.n2.1 & t.2.1<U.y2.n2.1,1,0)
```

## Anexos

---

```
Coverage.y3.n1.1.LIN[k] <- ifelse(t.3.1>L.y3.n1.1 & t.3.1<U.y3.n1.1,1,0)
Coverage.y3.n2.1.LIN[k] <- ifelse(t.3.1>L.y3.n2.1 & t.3.1<U.y3.n2.1,1,0)
```

```
Coverage.y1.n1.2.LIN[k] <- ifelse(t.1.2>L.y1.n1.2 & t.1.2<U.y1.n1.2,1,0)
Coverage.y1.n2.2.LIN[k] <- ifelse(t.1.2>L.y1.n2.2 & t.1.2<U.y1.n2.2,1,0)
Coverage.y2.n1.2.LIN[k] <- ifelse(t.2.2>L.y2.n1.2 & t.2.2<U.y2.n1.2,1,0)
Coverage.y2.n2.2.LIN[k] <- ifelse(t.2.2>L.y2.n2.2 & t.2.2<U.y2.n2.2,1,0)
Coverage.y3.n1.2.LIN[k] <- ifelse(t.3.2>L.y3.n1.2 & t.3.2<U.y3.n1.2,1,0)
Coverage.y3.n2.2.LIN[k] <- ifelse(t.3.2>L.y3.n2.2 & t.3.2<U.y3.n2.2,1,0)
```

```
der.y1.n1.1.LIN[k] <- ifelse(t.1.1 > U.y1.n1.1,1,0)
izq.y1.n1.1.LIN[k] <- ifelse(t.1.1 < L.y1.n1.1,1,0)
der.y1.n2.1.LIN[k] <- ifelse(t.1.1 > U.y1.n2.1,1,0)
izq.y1.n2.1.LIN[k] <- ifelse(t.1.1 < L.y1.n2.1,1,0)
der.y2.n1.1.LIN[k] <- ifelse(t.2.1 > U.y2.n1.1,1,0)
izq.y2.n1.1.LIN[k] <- ifelse(t.2.1 < L.y2.n1.1,1,0)
der.y2.n2.1.LIN[k] <- ifelse(t.2.1 > U.y2.n2.1,1,0)
izq.y2.n2.1.LIN[k] <- ifelse(t.2.1 < L.y2.n2.1,1,0)
der.y3.n1.1.LIN[k] <- ifelse(t.3.1 > U.y3.n1.1,1,0)
izq.y3.n1.1.LIN[k] <- ifelse(t.3.1 < L.y3.n1.1,1,0)
der.y3.n2.1.LIN[k] <- ifelse(t.3.1 > U.y3.n2.1,1,0)
izq.y3.n2.1.LIN[k] <- ifelse(t.3.1 < L.y3.n2.1,1,0)
```

```
der.y1.n1.2.LIN[k] <- ifelse(t.1.2 > U.y1.n1.2,1,0)
izq.y1.n1.2.LIN[k] <- ifelse(t.1.2 < L.y1.n1.2,1,0)
der.y1.n2.2.LIN[k] <- ifelse(t.1.2 > U.y1.n2.2,1,0)
izq.y1.n2.2.LIN[k] <- ifelse(t.1.2 < L.y1.n2.2,1,0)
der.y2.n1.2.LIN[k] <- ifelse(t.2.2 > U.y2.n1.2,1,0)
izq.y2.n1.2.LIN[k] <- ifelse(t.2.2 < L.y2.n1.2,1,0)
der.y2.n2.2.LIN[k] <- ifelse(t.2.2 > U.y2.n2.2,1,0)
izq.y2.n2.2.LIN[k] <- ifelse(t.2.2 < L.y2.n2.2,1,0)
der.y3.n1.2.LIN[k] <- ifelse(t.3.2 > U.y3.n1.2,1,0)
izq.y3.n1.2.LIN[k] <- ifelse(t.3.2 < L.y3.n1.2,1,0)
der.y3.n2.2.LIN[k] <- ifelse(t.3.2 > U.y3.n2.2,1,0)
izq.y3.n2.2.LIN[k] <- ifelse(t.3.2 < L.y3.n2.2,1,0)
```

```
}
### END ###
```

```
#### Dr. Luis Fernando Contreras Cruz, lufecon@gmail.com ####
```

## Anexo D: Demostración del Teorema 1 del Capítulo 6

Por simplicidad, definamos  $\hat{m}_i = \hat{m}_{VC}(\mathbf{x}_i)$  y  $m_i = m_{VC}(\mathbf{x}_i)$ . Entonces

$$\begin{aligned}
\frac{\sqrt{n_N}}{N} (\hat{t}_{y,VC} - t_y) &= \frac{\sqrt{n_N}}{N} \left( \sum_{i \in U_N} \hat{m}_i + \sum_{i \in s} \frac{y_i - \hat{m}_i}{\pi_i} - t_y \right) \\
&= \frac{\sqrt{n_N}}{N} \left[ \sum_{i \in U_N} \hat{m}_i + \sum_{i \in s} \frac{y_i - \hat{m}_i}{\pi_i} - t_y + \sum_{i \in U_N} m_i - \sum_{i \in U_N} m_i + \sum_{i \in s} \frac{m_i}{\pi_i} - \sum_{i \in s} \frac{m_i}{\pi_i} \right] \\
&= \frac{\sqrt{n_N}}{N} \left[ \sum_{i \in U_N} m_i + \sum_{i \in s} \frac{y_i - m_i}{\pi_i} - t_y + \sum_{i \in U_N} \hat{m}_i - \sum_{i \in U_N} m_i - \sum_{i \in s} \frac{\hat{m}_i}{\pi_i} + \sum_{i \in s} \frac{m_i}{\pi_i} \right] \\
&= \frac{\sqrt{n_N}}{N} \left[ (\hat{t}_{y,diff} - t_y) + \sum_{i \in U_N} c_i^T \hat{B}_{VC} - \sum_{i \in U_N} c_i^T B_{VC} - \sum_{i \in s} \frac{c_i^T \hat{B}_{VC}}{\pi_i} + \sum_{i \in s} \frac{c_i^T B_{VC}}{\pi_i} \right] \\
&= \frac{\sqrt{n_N}}{N} \left[ (\hat{t}_{y,diff} - t_y) + \sum_{i \in U_N} c_i^T \hat{B}_{VC} - \sum_{i \in U_N} c_i^T B_{VC} - \sum_{i \in U_N} c_i^T \hat{B}_{VC} \frac{I_i}{\pi_i} + \sum_{i \in U_N} c_i^T B_{VC} \frac{I_i}{\pi_i} \right] \\
&= \frac{\sqrt{n_N}}{N} \left[ (\hat{t}_{y,diff} - t_y) + \sum_{i \in U_N} c_i^T (\hat{B}_{VC} - B_{VC}) - \sum_{i \in U_N} c_i^T \frac{I_i}{\pi_i} (\hat{B}_{VC} - B_{VC}) \right] \\
&= \frac{\sqrt{n_N}}{N} \left[ (\hat{t}_{y,diff} - t_y) + \sum_{i \in U_N} c_i^T \left( 1 - \frac{I_i}{\pi_i} \right) (\hat{B}_{VC} - B_{VC}) \right] \\
&= \frac{\sqrt{n_N}}{N} (\hat{t}_{y,diff} - t_y) + \frac{\sqrt{n_N}}{N} \sum_{i \in U_N} c_i^T \left( 1 - \frac{I_i}{\pi_i} \right) (\hat{B}_{VC} - B_{VC}) \\
&= \frac{\sqrt{n_N}}{N} (\hat{t}_{y,diff} - t_y) + O_p \left( \frac{1}{\sqrt{n_N}} \right) \cdot o_p(1) \\
&= \frac{\sqrt{n_N}}{N} (\hat{t}_{y,diff} - t_y) + o_p \left( \frac{1}{\sqrt{n_N}} \right)
\end{aligned}$$



Ahora analizemos el siguiente término:

$$\begin{aligned}
 \frac{\sqrt{n_N}}{N} (\hat{t}_{y,diff} - t_y) &= \frac{\sqrt{n_N}}{N} \left[ \sum_{i \in U_N} m_i + \sum_{i \in s} \frac{y_i - m_i}{\pi_i} - \sum_{k \in U_N} y_k \right] \\
 &= \frac{\sqrt{n_N}}{N} \left[ \sum_{i \in U_N} m_i + \sum_{i \in U_N} y_i \frac{I_i}{\pi_i} - \sum_{i \in U_N} m_i \frac{I_i}{\pi_i} - \sum_{k \in U_N} y_k \right] \\
 &= \frac{\sqrt{n_N}}{N} \left[ \sum_{i \in U_N} y_i \left( \frac{I_i}{\pi_i} - 1 \right) - \sum_{i \in U_N} m_i \left( \frac{I_i}{\pi_i} - 1 \right) \right] \\
 &= \frac{\sqrt{n_N}}{N} \sum_{i \in U_N} y_i \left( \frac{I_i}{\pi_i} - 1 \right) - \frac{\sqrt{n_N}}{N} \sum_{i \in U_N} c_i^T \left( \frac{I_i}{\pi_i} - 1 \right) B_{VC}
 \end{aligned}$$

Por los *supuestos 1, 2 y 3*:

$$\frac{\sqrt{n_N}}{N} (\hat{t}_{y,diff} - t_y) \xrightarrow{d} N(\mathbf{0}, \Sigma_{yy} + B^T \Sigma_{cc} B - 2\Sigma_{yc} B).$$

Entonces:

$$\frac{\sqrt{n_N}}{N} (\hat{t}_{y,diff} - t_y) = O_p(1).$$

Por otra parte:

$$\begin{aligned}
 \frac{\sqrt{n_N}}{N} (\hat{t}_{y,VC} - t_y) &= \frac{\sqrt{n_N}}{N} (\hat{t}_{y,diff} - t_y) + o_p\left(\frac{1}{\sqrt{n_N}}\right) \\
 &= O_p(1) + o_p\left(\frac{1}{\sqrt{n_N}}\right)
 \end{aligned}$$

Entonces:

$$\frac{\hat{t}_{y,VC} - t_y}{N} = O_p\left(\frac{1}{\sqrt{n_N}}\right) + o_p\left(\frac{1}{\sqrt{n_N}}\right) = O_p\left(\frac{1}{\sqrt{n_N}}\right).$$

Por otra parte:

$$Var \left[ \frac{\sqrt{n_N}}{N} (\hat{t}_{y,diff} - t_y) \right] = \frac{n_N}{N^2} Var(\hat{t}_{y,diff}) \rightarrow \tau^2,$$

cuando  $N \rightarrow \infty$  con  $\tau^2 = \Sigma_{yy} + B^T \Sigma_{cc} B - 2\Sigma_{yc} B$ .

Por lo tanto

$$\frac{\hat{t}_{y,VC} - t_y}{\sqrt{Var(\hat{t}_{y,diff})}} \xrightarrow{d} N(0, 1).$$

## Anexo E: Demostración del Teorema 2 del Capítulo 6

Por simplicidad, definamos  $\widehat{m}_i = \widehat{m}_{VC}(\mathbf{x}_i)$  y  $m_i = m_{VC}(\mathbf{x}_i)$ . Entonces

$$\begin{aligned}\widehat{Var}(\widehat{t}_{y,VC}) &= \sum_{i,j \in s} \sum_{\pi_{ij}} \frac{\Delta_{ij} (y_i - m_i) + (m_i - \widehat{m}_i) (y_j - m_j) + (m_j - \widehat{m}_j)}{\pi_i \pi_j} \\ &= \sum_{i,j \in s} \sum_{\pi_{ij}} \frac{\Delta_{ij} y_i - m_i y_j - m_j}{\pi_i \pi_j} + 2 \sum_{i,j \in s} \sum_{\pi_{ij}} \frac{\Delta_{ij} y_i - m_i (m_j - \widehat{m}_j)}{\pi_i \pi_j} + \sum_{i,j \in s} \sum_{\pi_{ij}} \frac{\Delta_{ij} (m_i - \widehat{m}_i) (m_j - \widehat{m}_j)}{\pi_i \pi_j}\end{aligned}$$

Entonces

$$\widehat{Var}(\widehat{t}_{y,VC}) = \widehat{Var}(\widehat{t}_{y,diff}) + Var_1 + Var_2.$$

Donde,

$$\begin{aligned}Var_1 &= 2 \sum_{i,j \in s} \sum_{\pi_{ij}} \frac{\Delta_{ij} y_i - m_i m_j - \widehat{m}_j}{\pi_i \pi_j}, \\ Var_2 &= \sum_{i,j \in s} \sum_{\pi_{ij}} \frac{\Delta_{ij} m_i - \widehat{m}_i m_j - \widehat{m}_j}{\pi_i \pi_j}.\end{aligned}$$

Por el supuesto 4,  $\widehat{Var}(\widehat{t}_{y,diff}) = Var(\widehat{t}_{y,diff}) + o_p\left(\frac{N^2}{n_N}\right)$ .

Por otra parte, tenemos que:

$$\begin{aligned}Var_1 &= 2 \sum_{i,j \in s} \sum_{\pi_{ij}} \frac{\Delta_{ij} y_i - c_i^T B_{VC} c_j^T B_{VC} - c_j^T \widehat{B}_{VC}}{\pi_i \pi_j} \\ &= 2 \sum_{i,j \in s} \sum_{\pi_{ij}} \frac{\Delta_{ij} c_i^T B_{VC} - y_i c_j^T}{\pi_i \pi_j} (\widehat{B}_{VC} - B_{VC}) \\ &= 2 \sum_{i,j \in s} \sum_{\pi_{ij}} \left\{ \frac{\Delta_{ij} B_{VC} c_i^T c_j}{\pi_i \pi_j} (\widehat{B}_{VC} - B_{VC}) - \frac{\Delta_{ij} y_i c_j^T}{\pi_i \pi_j} (\widehat{B}_{VC} - B_{VC}) \right\} \\ &= 2 \sum_{i,j \in s} \sum_{\pi_{ij}} \frac{\Delta_{ij} B_{VC} c_i^T c_j}{\pi_i \pi_j} (\widehat{B}_{VC} - B_{VC}) - 2 \sum_{i,j \in s} \sum_{\pi_{ij}} \frac{\Delta_{ij} y_i c_j^T}{\pi_i \pi_j} (\widehat{B}_{VC} - B_{VC}) \\ &= O_p\left(\frac{N^2}{n_N}\right) o_p(1) - O_p\left(\frac{N^2}{n_N}\right) o_p(1) \\ &= O_p\left(\frac{N^2}{n_N}\right).\end{aligned}$$

Por otra parte

$$\begin{aligned}
 Var_2 &= \sum_{i,j \in s} \sum_{i,j \in s} \frac{\Delta_{ij}}{\pi_{ij}} \frac{m_i - \hat{m}_i}{\pi_i} \frac{m_j - \hat{m}_j}{\pi_j} \\
 &= \sum_{i,j \in s} \sum_{i,j \in s} \frac{\Delta_{ij}}{\pi_{ij}} \frac{c_i^T B_{VC} - c_i^T \hat{B}_{VC}}{\pi_i} \frac{c_j^T B_{VC} - c_j^T \hat{B}_{VC}}{\pi_j} \\
 &= \left( \hat{B}_{VC} - B_{VC} \right)^T \sum_{i,j \in s} \sum_{i,j \in s} \frac{\Delta_{ij}}{\pi_{ij}} \frac{c_i c_j^T}{\pi_i \pi_j} \left( \hat{B}_{VC} - B_{VC} \right) \\
 &= o_p(1) O_p \left( \frac{N^2}{n_N} \right) o_p(1) \\
 &= o_p \left( \frac{N^2}{n_N} \right).
 \end{aligned}$$

Entonces  $\widehat{Var}(\hat{t}_{y,VC}) = Var(\hat{t}_{y,diff}) + O_p\left(\frac{N^2}{n_N}\right) + o_p\left(\frac{N^2}{n_N}\right)$ . Por lo tanto

$$\widehat{Var}(\hat{t}_{y,VC}) = Var(\hat{t}_{y,diff}) + o_p\left(\frac{N^2}{n_N}\right).$$

## Anexo F: Demostración del Corolario 1 del Capítulo 6

Notemos que:

$$\frac{\widehat{t}_{y,VC} - t_y}{\sqrt{\widehat{Var}(\widehat{t}_{y,VC})}} = \frac{\widehat{t}_{y,VC} - t_y}{\sqrt{Var(\widehat{t}_{y,diff})}} \sqrt{\frac{Var(\widehat{t}_{y,diff})}{\widehat{Var}(\widehat{t}_{y,VC})}},$$

entonces el primer término de la expresión anterior  $\xrightarrow{d} N(0,1)$  por el **Teorema 1**, el segundo término *converge en probabilidad a 1* por el **Teorema 2** y por el Teorema de Slutsky ([Serfling, 1980](#)) se tiene que:

$$\frac{\widehat{t}_{y,VC} - t_y}{\sqrt{\widehat{Var}(\widehat{t}_{y,VC})}} \xrightarrow{d} N(0,1).$$