

# PACKAGE “STRATIFICATION”

---

Estratificación Univariada  
para Variables Asimétricas cuando  $H$  es dado

# Base.Productores.RData

764 observations of 10 variables

Uk	RENSPA	PRODUCTOR	EMPRESA.INTEGRADORA	TAMANIO	LOCALIDAD	LAT	LONG	ZONA	VENTAS
1	07.014.0.55448/01	ALBERT DORA FILIPUZZI DE	SUPER	3000	COLONIA ELIA			1	2271
2	07.014.0.50932/03	GAILLARD MARTIN EDEL	SERVIAVE S.A.	4000	PRONUNCIAMIENTO	-32.36487	-58.36213	2	6084
3	07.014.0.51659/02	CARBALLO CLEMENTE A.	FADEL S.A.	4000	ARROYO MOLINO			6	2320
4	07.014.0.54308/05	BLANC GRACIELA	BONNIN HNOS	4000	PRONUNCIAMIENTO			2	158
5	07.014.0.57861/01	GURNEL EVELIO LORENZO	FADEL S.A.	4000	COL. 5TO. ENS. MAYO			2	257
6	07.014.0.52354/01	ROUGUIER ROBERTO	NOELMA S.A.	5000	COLONIA 3 DE FEBRERO			2	2053
7	07.014.0.57310/01	JANNON CARLOS	BONNIN HNOS	5000	1j DE MAYO			2	613
8	07.014.0.50918/03	GABIoud GUSTAVO GABRIEL	FADEL S.A.	5200	COL. 5TO. ENS. MAYO	-32.3425	-58.32194	2	5461
9	07.014.0.51132/03	CUMBETO DE BOURLOT NORA	FADEL S.A.	5500	CASEROS	-32.40889	-58.49861	3	3134
10	07.014.0.56017/03	JUAREZ GABRIEL ANGEL	SERVIAVE S.A.	5500	VILLA MANTERO			4	7102
11	07.014.0.56057/01	MOUT ODER AUGUSTO	SUPER	5500	COLONIA 3 DE FEBRERO			2	2602
12	07.014.0.57832/01	SUFFO PATRICIA	SUPER	5710	COLONIA ELIA			1	1853
13	07.014.0.57875/01	DELAJOYE MARIO JORGE	BONNIN HNOS	5800	COLONIA PERFECCION			6	858
14	07.014.0.56359/06	VIGANONI RAUL	TABORDA RODOLFO MAXIMILIANO	6000	SANTA ANA	-32.55778	-58.43028	1	5495
15	07.014.0.56855/01	GARNIER OSCAR	FADEL S.A.	6000	CASEROS			3	3417
16	07.014.0.57285/00	ORCELLET MABEL TERESA	NOELMA S.A.	6000	1j DE MAYO			2	7949
17	07.014.0.57862/01	GUIONET LEONEL ELEUTERIO	FADEL S.A.	6000	COLONIA SAN CIPRIANO			2	2258

# Variable Estratificadora Asimétrica

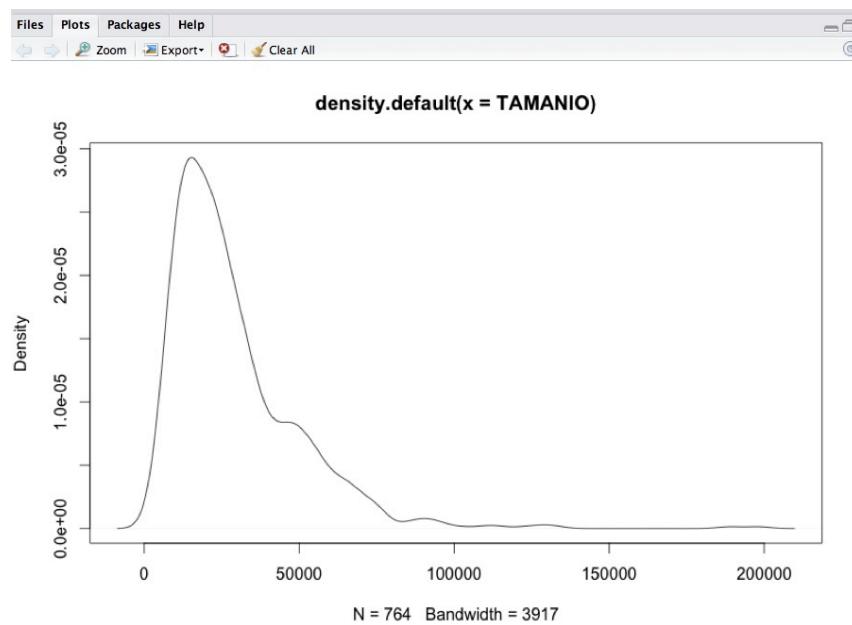
```
> library(dplyr)
> library(ggplot2)
> library(stratification)
> load("~/Downloads/Curso Cordoba 2015/Clases/Base.Productores 2015.RData")
> attach(Base.Productores)
> head(Base.Productores)
```

Uk	RENSPA	PRODUCTOR	EMPRESA.INTEGRADORA	TAMANIO	LOCALIDAD	LAT	LONG
1	07.014.0.55448/01	ALBERT DORA FILIPUZZI DE	SUPER	3000	COLONIA ELIA		
2	07.014.0.50932/03	GAILLARD MARTIN EDEL	SERVIACE S.A.	4000	PRONUNCIAMIENTO	-32.36487	-58.36213
3	07.014.0.51659/02	CARBALLO CLEMENTE A.	FADEL S.A.	4000	ARROYO MOLINO		
4	07.014.0.54308/05	BLANC GRACIELA	BONNIN HNOS	4000	PRONUNCIAMIENTO		
5	07.014.0.57861/01	GURNEL EVELIO LORENZO	FADEL S.A.	4000	COL. 5TO. ENS. MAYO		
6	07.014.0.52354/01	ROUGUIER ROBERTO	NOELMA S.A.	5000	COLONIA 3 DE FEBRERO		

ZONA VENTAS

1	1	2271
2	2	6084
3	6	2320
4	2	158
5	2	257
6	2	2053

```
> # Verificacion de Asimetria
> nrow(Base.Productores)
[1] 764
> summary(TAMANIO)
Min. 1st Qu. Median Mean 3rd Qu. Max.
3000 14700 23500 29130 36700 198100
> plot(density(TAMANIO))
```



# Métodos para Estratificar una Población

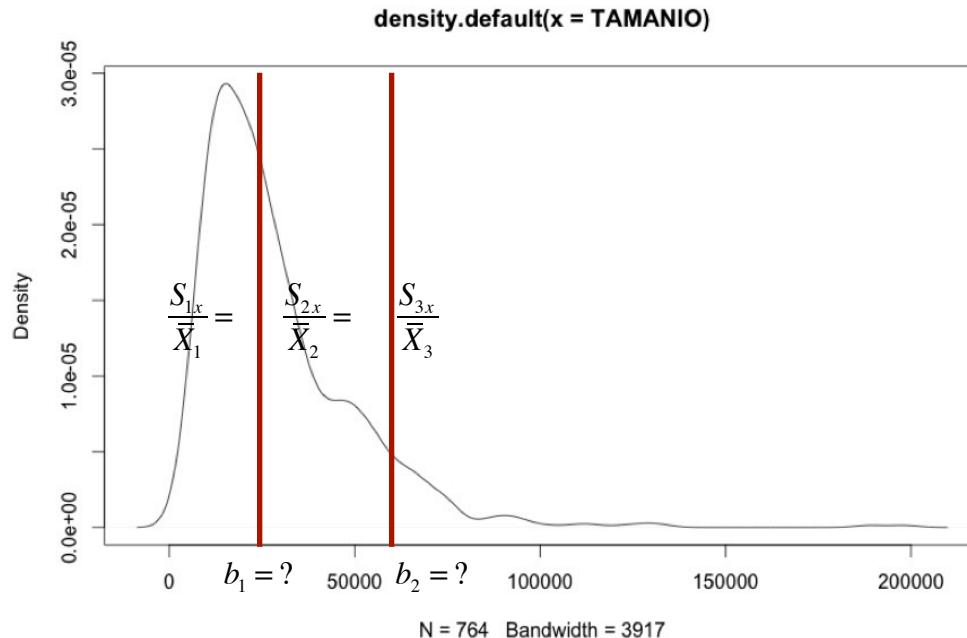
(sin restricciones de optimabilidad y fijando la cantidad de estratos)

## MÉTODO DE GUNNING & HORGAN

(Dist. Lognormal, Gamma, Weibull, Beta)

Dada una variable estratificadora  $X$  conocida para toda la población y un nro de estratos predeterminados  $H$  se quiere determinar  $b_h, h = 1, \dots, H$   $\min(X) = b_0 < b_1 < \dots < b_H = \max(X)$

de tal manera que los CV satisfagan  $\frac{S_{1x}}{\bar{X}_1} = \frac{S_{2x}}{\bar{X}_2} = \dots = \frac{S_{Hx}}{\bar{X}_H} \Rightarrow \min\left\{V_{HT}\left(\hat{T}_X\right)\right\}$



$H = 3$  dado  
 $b_0 = \min(X)$   
 $b_1 = ?$   
 $b_2 = ?$   
 $b_3 = \max(X)$

# Supuestos

Asumiendo a) una distribución asimétrica para el supuesto de CV iguales en los estratos  
y b) aprox. una distribución uniforme en cada estrato,

$$\bar{X}_h = \frac{b_h + b_{h-1}}{2}, S_{xh} = \frac{1}{\sqrt{12}}(b_h - b_{h-1}), CV_h = \frac{(b_h - b_{h-1})/\sqrt{12}}{(b_h + b_{h-1})/2}$$

$$\text{Si } CV_1 = \dots = CV_H \text{ entonces } \frac{b_{h+1} - b_h}{b_{h+1} + b_h} = \frac{b_h - b_{h-1}}{b_h + b_{h-1}} \Rightarrow b_h^2 = b_{h+1}b_{h-1}$$

(progresión geométrica)

$$b_h = ar^h \quad (h = 0, 1, \dots, H) \quad a = b_0 \quad ar^H = b_H \quad r = (b_H/b_0)^{1/H}$$

Se desean 3 estratos ( $H=3$ )  $X=\text{ha}$

se sabe que  $\min(X)=5$  y  $\max(X)=50000$

la progresión geométrica tiene por constante  $a$  y razón  $r$  :

$$a = \min(X) = 5 \quad y \quad r = (50000/5)^{1/3} \doteq 20 \quad \text{entonces}$$

$$b_h = 5(20)^h \quad h = 0, 1, 2, 3$$

$$\{5, 100, 2000\}$$

Estratos:  $[5, 100), [100, 2000), [2000, 50000)$

# Método de Gunning & Horgan

## Comando Strata.Geo()

*objeto = strata.geo(***X** = Variable estratificadora,  
**Ls** = nro estratos,  
**CV** =  $c_0$ , **n** =  $n_0$ , **alloc** =  $c(q_1, q_2, q_3)$ ,  
**rh** = vector)

*Advertencias :*

- ⊗  $CV = c_0$  o  $n = n_0$  (no los dos simultáneamente)
- ⊗ No se emplean para determinar los estratos.
- ⊗ Permiten calcular  $n$  para un  $CV = c_0$  o bien el CV para  $n = n_0$  con la estratificación alcanzada y la asignación definida en *alloc*.
- ⊗ *alloc* tampoco se emplea para determinar los estratos
- ⊗ *rh* vector con tasas de "respuesta" por estrato

# Parámetros “alloc” y “rh”

*Alloc(q<sub>1</sub>,q<sub>2</sub>,q<sub>3</sub>)*

$$V_{HT}(\hat{T}_{\pi X}) = \sum_{h=1}^H N_h^2 \left( \frac{1}{na_h} - \frac{1}{N_h} \right) S_{Xh}^2 \quad a_h = \frac{\alpha_h}{\sum_{h=1}^H \alpha_h} \quad \alpha_h = N_h^{2q_1} \bar{X}_h^{2q_2} S_{Xh}^{2q_3}$$

$$\begin{cases} \text{Neyman} & q_1 = 1/2, q_2 = 0, q_3 = 1/2 \\ \text{Proporcional} & q_1 = 1/2, q_2 = 0, q_3 = 1/2 \\ \text{Power} & q_1 = p/2, q_2 = p/2, q_3 = 0 \quad p > 0 \end{cases}$$

*rh=rep(1,L<sub>s</sub>) default*

*rh=0.85, rh=c(1,0.95,1,0.80)*

$$n = \frac{\sum_{h=1}^H N_h^2 S_{Xh}^2 / a_h r_h}{cv_0^2 T_{\pi X}^2 + \sum_{h=1}^{H-1} N_h S_{Xh}^2} \quad n_h = na_h \quad h = 1, \dots, H-1$$

## con CV=0.03

```
> #####
> # Estratificacion
> #####
> geo=strata.geo(x=TAMANIO,CV=0.03,Ls=3,alloc=c(0.5,0,0.5))
> geo
Given arguments:
x = TAMANIO
CV = 0.03, Ls = 3
allocation: q1 = 0.5, q2 = 0, q3 = 0.5
model = none

Strata information:
      |      type rh |      bh      E(Y)    Var(Y)   Nh   nh   fh
stratum 1 | take-some 1 | 12126.58  9109.65  4946030 133    5  0.04
stratum 2 | take-some 1 | 49017.96 25715.63  93378295 515   72  0.14
stratum 3 | take-some 1 | 198141.00 67215.45 555948798 116   40  0.34
Total                               764 117  0.15

Total sample size: 117
Anticipated population mean: 29125.82
Anticipated CV: 0.02964842
> geo$bh
[1] 12126.58 49017.96
```

# con n=40

```
> geo=strata.geo(x=TAMANIO,n=40,Ls=3,alloc=c(0.5,0,0.5))
```

```
> geo
```

Given arguments:

x = TAMANIO

n = 40, Ls = 3

allocation: q1 = 0.5, q2 = 0, q3 = 0.5

model = none

Strata information:

		type	rh	bh	E(Y)	Var(Y)	Nh	nh	fh
stratum 1		take-some	1	12126.58	9109.65	4946030	133	1	0.01
stratum 2		take-some	1	49017.96	25715.63	93378295	515	25	0.05
stratum 3		take-some	1	198141.00	67215.45	555948798	116	14	0.12
Total									

Total sample size: 40

Anticipated population mean: 29125.82

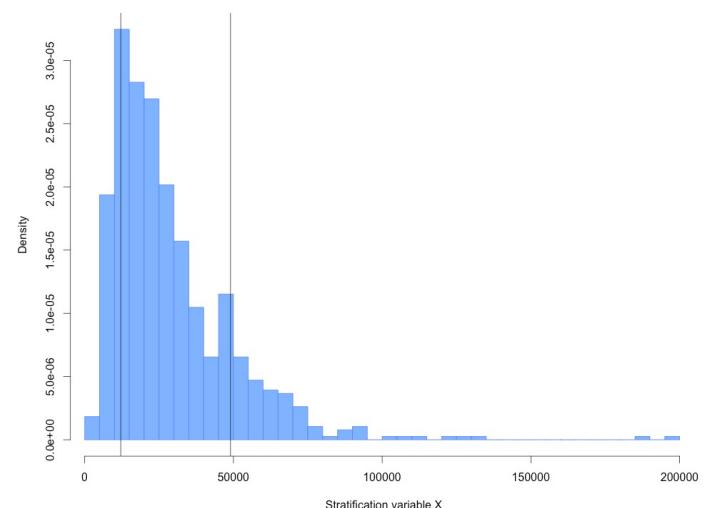
Anticipated CV: 0.05502568

```
> geo$bh
```

```
[1] 12126.58 49017.96
```

```
> plot(geo)
```

Graphical Representation of the Stratified Design geo			
	1	2	3
Nh	133	515	116
nh	1	25	14



# Búsqueda de un Estrato de Autorepresentados

$U = \{u_{(1)}, u_{(2)}, \dots, u_{(N)}\}$  ordenada por los valores de X



P-M="Pequeñas y Medianas"    G="Grandes"

$$\{u_{(1)}, \dots, u_{(N-k+1)}\}$$

$$\{u_{(N-k+1)}, \dots, u_{(N)}\}$$

- Sean :
- ⊗  $k = \#U_G$  a determinar
  - ⊗  $n(k)$  tamaño de la muestra (incluye a las  $k$  "más grandes")
  - ⊗  $n(k) - k$  total de la muestra en  $U - U_G$  a determinar

$$T_{\pi x} = \sum_{i=N-k+1}^N x_{(i)} + \sum_{i=1}^{N-k} x_{(i)} \Rightarrow \hat{T}_{\pi x} = \sum_{i=N-k+1}^N x_{(i)} + \frac{N-k}{n(k)-k} \sum_{j \in s} x_{(j)}$$

Componente no  
"aleatorizada"      Componente  
"aleatorizada"

con  $\pi_j = \frac{n(k)-k}{N-k}$   
bajo MSASR

# Planteo

Dada  $X$  asimétrica para estratificar y un  $CV(\hat{T}_{\pi_x}) = cv_0$  deseado para  $\hat{T}_{\pi_x}$   
se busca:

a)  $H=2$  estratos (Autorepresentados y No Autorepresentados)

$$b_0 = \min(X) \quad b_1 = ? \quad b_2 = \max(X)$$

b) un tamaño  $n$  tal que se satisfaga el requisito de precisión impuesto

$$CV^2(\hat{T}_{\pi_x}) = \frac{V_{MSA}(\hat{T}_{\pi_x})}{T_x^2} = cv_0^2 \quad \text{con} \quad V_{MSA}(\hat{T}_{\pi_x}) = N^2 \left( \frac{1}{n} - \frac{1}{N} \right) S_x^2$$

$$\text{despejando } n : \quad n = N - \frac{Ncv_0^2 T_x^2}{cv_0^2 T_x^2 + NS_x^2}$$

# Resolución Numérica

$$\hat{T}_{\pi x} = \sum_{i=N-k+1}^N x_{(i)} + \frac{N-k}{n(k)-k} \sum_{j \in s} x_{(j)}$$

$$V(\hat{T}_{\pi x}) = 0 + (N-k) \frac{N-n(k)}{n(k)-k} S_{x[N-k]}^2 \quad \text{y dado un } CV(\hat{T}_{\pi x}) = cv_0 \quad V(\hat{T}_{\pi x}) = cv_0^2 T_x^2$$

$$cv_0^2 T_x^2 = (N-k) \frac{N-n(k)}{n(k)-k} S_{x[N-k]}^2 \Rightarrow n(k) = N - \frac{(N-k)cv_0^2 T_x^2}{cv_0^2 T_x^2 + (N-k)S_{[N-k]}^2}$$

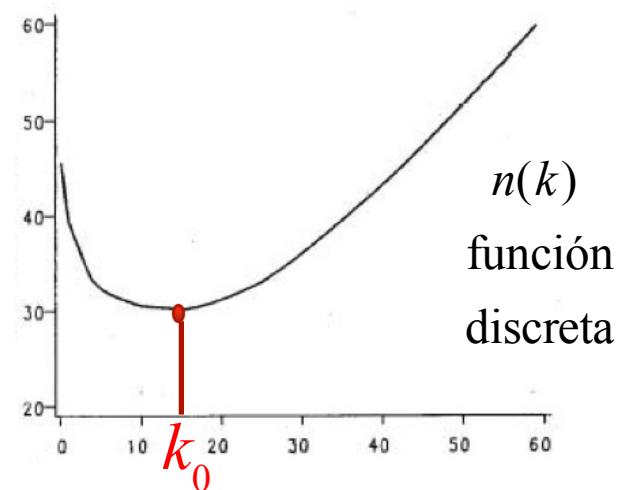
$$\min_{1 \leq k \leq N-1} \{n(k)\} \quad \text{Si} \quad k_0 = \min_{1 \leq k \leq N-1} \{n(k)\}$$

$$\text{Los "P-M"} \quad U_{[k_0-1]} = \{u_{(1)}, \dots, u_{(N-k_0+1)}\}$$

$$MSASR(N-k_0, n(k_0) - k_0)$$

$$\text{Los "G"} \quad U - U_{[k_0-1]} = \{u_{(N-k_0)}, \dots, u_{(N)}\}$$

$$u \in "G" \Rightarrow P_d(u) = 1$$



# Búqueda de más de 2 Bordes y cuando el nro de Estratos H es dado

Recordando que bajo un muestreo estratificado con H estratos :

$$V_{xHT} \left( \hat{T}_{HTx} \right) = \sum_{h=1}^H N_h^2 \left( \frac{1}{n_h} - \frac{1}{N_h} \right) S_{hx}^2$$

Si se conoce el nro de estratos (H) y se dispone de una variable estratificadora (X)  
¿cuál será la mejor estratificación que minimiza  $V_{HT} \left( \hat{T}_{HTx} \right)$ ?

Hay que responder a:

¿quiénes serán  $b_0, \dots, b_H$  ?

y      ¿quiénes los  $n_1, \dots, n_h$  ? tal que

$$n(b_1, \dots, b_{H-1}) = \sum_{h=1}^{H-1} n_h(b_h)$$

El problema así planteado no tiene respuesta inmediata dada la característica de la función  $V_{xHT} \left( b_1, \dots, b_{H-1}, n_1, \dots, n_H \right)$  le sobran incognitas!

# Acotando el Problema

El problema cambia si:

- 1) Se impone una cota,  $c_0$  para el CV de  $\hat{T}_{HTx}$
- 2) Se fija un tipo de asignación

$$1) \text{CV}^2_{HT_x}\left(\hat{T}_{HTx}\right) = \frac{\sum_{h=1}^H N_h^2(b_h) \left( \frac{1}{n_h(b_h)} - \frac{1}{N_h(b_h)} \right) S_{hx}^2(b_h)}{T_x^2} \leq c_0^2$$

$$2) n_h(b_h) = n(b_1, \dots, b_H) a_h \text{ tal que } \sum_{h=1}^H a_h = 1$$

$$a_h(b_h) = \gamma_h(b_h) / \sum_{h=1}^H \gamma_h(b_h) \rightarrow \gamma_h(b_h) = N_h^{2q_1}(b_h) \bar{X}_h^{2q_2}(b_h) S_{hx}^{2q_3}(b_h)$$

Proporcional     $q_1 = 1/2, \quad q_2 = 0, \quad q_3 = 0$

Neyman               $q_1 = 1/2, \quad q_2 = 0, \quad q_3 = 1/2$

Power                 $q_1 = p/2, \quad q_2 = p/2, \quad q_3 = 0 \quad p > 0$

# El Problema de Optimización

reemplazando 2) en 1) y despejando  $n(b_1, \dots, b_{H-1}) = \frac{N \sum_{h=1}^H W_h^2(b_h) S_{hx}^2(b_h) / a_h(b_h)}{N c_0^2 T_x^2 + \sum_{h=1}^H W_h(b_h) S_{hx}^2(b_h)}$

$$\min \{n(b_1, \dots, b_{H-1})\}$$

$$CV(\hat{T}_{xHT}) \leq cv_0$$

- ⊗ Los únicos valores que dependen de  $b_1, \dots, b_{H-1}$  son:  $N_h(W_h), \bar{X}_h$  y  $S_{hx}^2$
- ⊗ La minimización se realiza por métodos iterativos sobre un problema de  $H-1$  variables y puede llevar a mínimos locales y no globales.
- ⊗ Los enfoques que sobresalen son los que emplean:
  - a) derivadas parciales (Lavalle-Hidiroglou)
  - b) search o búsqueda "intensiva" (Kozak)

# Métodos que Emplean Derivadas

## (HIDIROGLOU-LAVALLE)

Determinar H estratos (uno AutoR) optimizando el tamaño de la muestra  $n$  para una precisión dada  $cv_o$  de  $\hat{T}_{\pi_y}$ .

$$n(b_1, \dots, b_{H-1}) = N_H + \frac{N \sum_{h=1}^{H-1} W_h^2(b_h) S_{Xh}^2(b_h) / a_h(b_h)}{N c v_0^2 T_{\pi X}^2 + \sum_{h=1}^{H-1} W_h(b_h) S_{Xh}^2(b_h)}$$

$\Leftarrow \begin{cases} \text{"Función Objetivo"} \\ \min \{ n(b_1, \dots, b_{H-1}) \} \\ \text{se la asume "continua"} \end{cases}$

Métodos Numéricos Iterativos para resolver el sistema de H-1 ecuaciones:

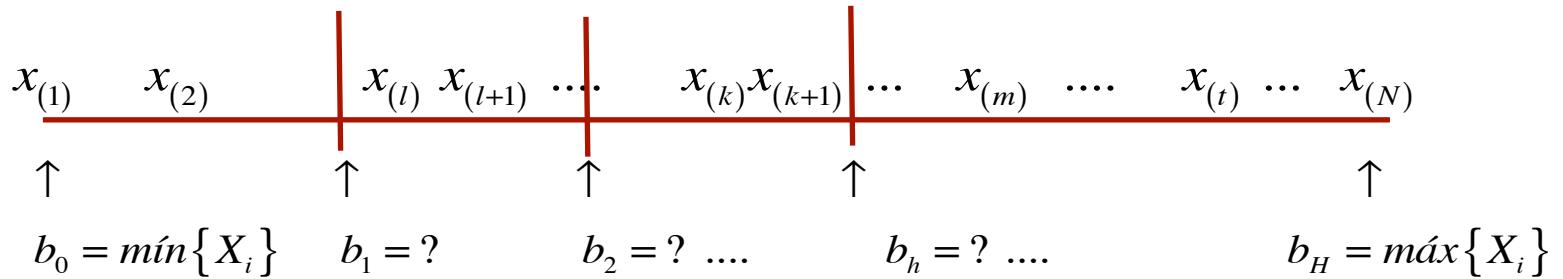
$$\frac{\partial n(b_1, \dots, b_{H-1})}{\partial b_h} = 0 \quad , h = 1, \dots, H-1$$

- ⊗ Pueden ser muy INESTABLES a causa del empleo de las derivadas parciales
- ⊗ Depende de los  $\{b_1^0, \dots, b_{H-1}^0\}$  iniciales y el método puede no converger
- ⊗ Los alcanzados por el método pueden ser un mínimo local de la función  $n(b_1, \dots, b_{H-1})$

# Métodos de Búsqueda Intensa



(ALGORITMO DE KOZAK)



Objetivo: "Elegir  $b_{H-1}$  valores de  $D = \{X^*_{(1)}, X^*_{(2)}, \dots, X^*_{(D)}\}$ , el conjunto de valores únicos de las X's, buscando minimizar la función objetivo  $n(b_1, \dots, b_{H-1})$ "

Si  $\binom{\# D - 1}{H - 1}$  es pequeño la búsqueda no es intensa

En cambio si  $N = 1000$  y se quiere  $H = 5 \Rightarrow \binom{1000}{5} > 8 * 10^{12}$   
pasar por todas las posibilidades o es imposible o es ineficiente!

# Algoritmo de Kozak

- 1) Proponer bordes iniciales  $b_1^0, b_2^0, \dots, b_H^0$
- 2) Calcular  $n(b_1^0, b_2^0, \dots, b_H^0)$
- 3) Elejir al azar un borde,  $b^*$
- 4) Seleccionar una modificación  $d$  de los  $2 * \underline{\max step}$  posibles  
del conjunto  $D \in \{-\max step, -\max step + 1, \dots, -1, 1, \dots, \max step - 1, \max step\}$
- 5) modificar el borde seleccionado moviéndolo  $d$  posiciones sobre el vector ordenado X,  $b^{new} = b^* + d(posiciones)$
- 6) Calcular  $n(b_1, b_2, \dots, b_H)$  con  $b^{new}$
- 7) a) Aceptar  $b^{new}$  si el tamaño de muestra  $n$  disminuye  
b) Rechazarlo y volver al paso 3)
- 8) Parar si: a)  $n$  se mantiene sin cambios durante  $\max still$  iteraciones  
b) o se llegó a un nro máximo de iteraciones,  $\max iter$

## Comando strata.LH()

`objeto = strata.LH(`*X*`= Variable estratificadora,`  
`CV = cv0,`  
`Ls = nro estratos,`  
`alloc = c(q1,q2,q3),`  
`takeall = 0,`  
`rh = 1,`  
`initbh = b0,`  
`model = c("none","loglinear","linear","random"),`  
`model.control = vector de parámetros del modelo,`  
`algo.control = list(rep = 5,maxiter = 10000,maxstep = 50,`  
`maxstill = 500))`

# Parámetros Propios del Algoritmo

*algo.control = list(rep = 5,maxiter = value,maxstep = value,  
minNh=value,maxstill = value, minsol = value, trymany = TRUE)*

Por defecto el algoritmo fija:

*rep* = 5 (permite entero  $\geq 1$ )

*maxiter* = 10000, (se puede disminuir o incrementar)

$$\text{maxstep} = \text{trunc} \left\{ \text{round} \left( \frac{N_{U-Ls}^u}{10} \right), 100 \right\}$$

*maxstill* = *max step* \* 10, (pero se permite  $50 \leq \text{maxstill} \leq 500$ )

*minsol* = 10000 (pero se permite  $2 \leq \text{minsol} \leq 2000000$ )

*minNh*=2 (permite entero  $> 1$ )

*trymany* = *TRUE* (dejar siempre esta opción en lo posible)

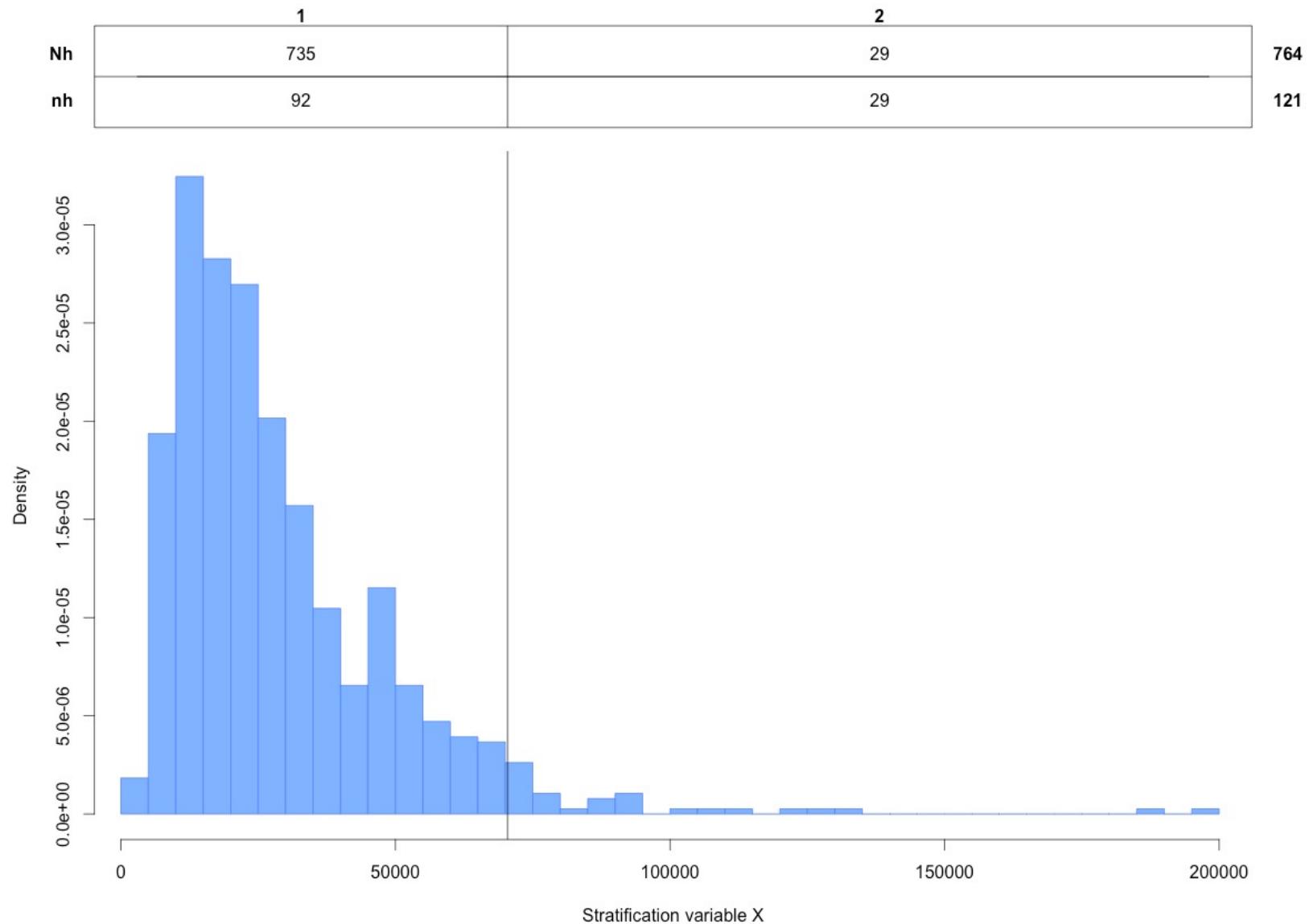
# Solución por strata.LH()

```
> #####
> # Kozak para determinar 1 Take-All
> #####
> Take1=strata.LH(x=TAMANIO,CV=0.05,Ls=2,takeall=1,algo="Kozak")
Warning message:
the number of possible solutions was smaller than 'minsol', therefore Kozak's algorithm !
, instead every possible strata boundaries were tried
> Take1
Given arguments:
x = TAMANIO
CV = 0.05, Ls = 2, takenone = 0, takeall = 1
allocation: q1 = 0.5, q2 = 0, q3 = 0.5
model = none
algo = Kozak: method = complete enumeration, minsol = 1000, idopti = nh, minNh = 2

Strata information:
      |      type rh |      bh      E(Y)      Var(Y)   Nh   nh   fh
stratum 1 | take-some 1 | 70433 26488.9 240692386 735  92 0.13
stratum 2 |  take-all  1 | 198141 95958.0 1008921960  29  29 1.00
Total                         764 121 0.16

Total sample size: 121
Anticipated population mean: 29125.82
Anticipated CV: 0.04997064
Note: CV=RRMSE (Relative Root Mean Squared Error) because takenone=0.
> Take1$bh
[1] 70433
> # Tamanio Muestral sin Autorepresentados para cv0=0.05
> n=(sd(TAMANIO)/(mean(TAMANIO)*0.05))**2
> n
[1] 210.6148
> plot(Take1)
```

### Graphical Representation of the Stratified Design Take1



## 4 Estratos y Asignación de Neyman

```
> #####
> #          Kozak No Model
> #####
> Kozak=strata.LH(x=TAMANIO,CV=0.02,Ls=4,alloc=c(0.5,0,0.5))
> Kozak
Given arguments:
x = TAMANIO
CV = 0.02, Ls = 4, takenone = 0, takeall = 0
allocation: q1 = 0.5, q2 = 0, q3 = 0.5
model = none
algo = Kozak: minsol = 1000, idopti = nh, minNh = 2, maxiter = 10000,
           maxstep = 50, maxstill = 500, rep = 5, trymany = TRUE

Strata information:
      |      type rh |      bh      E(Y)    Var(Y)   Nh nh   fh
stratum 1 | take-some 1 | 20550 13384.34 18228129 325 27 0.08
stratum 2 | take-some 1 | 38200 27945.95 22971714 255 24 0.09
stratum 3 | take-some 1 | 67090 50645.79 60139603 147 22 0.15
stratum 4 | take-some 1 | 198141 90028.81 918405330 37 22 0.59
Total                                     764 95 0.12

Total sample size: 95
Anticipated population mean: 29125.82
Anticipated CV: 0.01985317
Note: CV=RRMSE (Relative Root Mean Squared Error) because takenone=0.
```

## 4 Estratos, 1 Autorepresentado y con supuesto de No Respuesta Uniforme del 25%

```
> # 25% no-respuesta y con Autorepresentados
> Kozak1=strata.LH(x=TAMANIO,CV=0.02,Ls=4,alloc=c(0.5,0,0.5),takeall=1,
+ rh=0.75)
> Kozak1
Given arguments:
x = TAMANIO
CV = 0.02, Ls = 4, takenone = 0, takeall = 1
allocation: q1 = 0.5, q2 = 0, q3 = 0.5
model = none
algo = Kozak: minsol = 1000, idopti = nh, minNh = 2, maxiter = 10000,
           maxstep = 50, maxstill = 500, rep = 5, trymany = TRUE
```

Strata information:

	type	rh	bh	E(Y)	Var(Y)	Nh	nh	fh
stratum 1	take-some	0.75	20700.0	13406.48	18331436	326	35	0.11
stratum 2	take-some	0.75	38550.0	28056.71	23520933	256	31	0.12
stratum 3	take-some	0.75	70980.5	51862.67	73233843	154	33	0.21
stratum 4	take-all	0.75	198141.0	96865.93	1021049193	28	28	1.00
Total						764	127	0.17

Total sample size: 127

Anticipated population mean: 29125.82

Anticipated CV: 0.01994349

Note: CV=RRMSE (Relative Root Mean Squared Error) because takenone=0.

>

# Vectores del Objeto “Kozak1” [List]

```
> Kozak1$bh
[1] 20700.0 38550.0 70980.5
> Kozak1$meanh
[1] 13406.48 28056.71 51862.67 96865.93
> Kozak1$varh
[1] 18331436 23520933 73233843 1021049193
> Kozak1$nh
[1] 35 31 33 28
> Kozak1$Nh
[1] 326 256 154 28
> Kozak1$nsol
[1] 19608085
> Kozak1$iter.detail[1:10,]
      b1      b2      b3 opti.nh opti.nhnonint takeall step iter run
1 19261.67 32271.0 55037.33    145    143.0682      1     0     0     1
2 18053.00 32250.0 55154.00    144    142.5539      1   -13     2     1
3 18053.00 33067.5 55154.00    144    141.9126      1    10    11     1
4 18053.00 33067.5 55950.00    141    139.3720      1     5    14     1
5 18053.00 34022.5 55950.00    140    138.9618      1     8    15     1
6 18053.00 34022.5 56200.00    139    138.4327      1     1    25     1
7 18053.00 34022.5 73400.00    132    130.8000      1    41    30     1
8 18053.00 34022.5 70433.00    132    130.1391      1   -3    40     1
9 18053.00 41116.0 70433.00    131    130.0623      1    41    46     1
10 19216.50 41116.0 70433.00   130    128.8260      1    12    53     1
> |
```

# Parámetro “minsol”

> Kozak1\$nsol

[1] 19608085

- ⊗ Si  $N_u^x$  son el total de valores sin repetición que toma la variable X,
- ⊗  $L_s$  el nro de estratos deseados,
- ⊗ Existen  $\binom{N_u^x - 1}{L_s - 1}$  bordes posibles
- ⊗ Objetivo minimizar  $n(b_1, \dots, b_{L_s-1})$ .
- ⊗ Cómo el problema tiene un orden de infinitud muy grande dependiendo de  $N_u^x$  y  $L_s$ , el algoritmo emplea una busqueda aleatoria para resolverlo.
- ⊗ Si  $\binom{N_u^x - 1}{L_s - 1} \leq "minsol" = 1000$  el algoritmo pasa por todas las alternativas
- ⊗ Se puede fijar  $0 \leq "minsol" \leq 2000000$

## Sin *initbh*=()

Si no se especifica *initbh* el algoritmo trabaja con 3 juegos de bordes iniciales:

- 1) Dalenius & Hodges:  $Acum\sqrt{f}$  (*strata.cumrootf*)
- 2) Gunning & Horgan: *progresión geométrica* (*strat.geo*)
- 3) Robusto: propone como iniciales aquellos que en forma simultánea:

- $N_h \geq 2, n_h \geq 0,$
- el  $L_s$  (el AutoR) tenga la menor cantidad de unidades
- el resto de los estratos tengan aproximadamente el mismo tamaño,

$$\#U_h \cong \frac{N - N_{L_s}}{L_s - 1}, \text{ si no hay AutR } b_h = \min X + h * d, h=1, \dots, L_s - 1$$

$$d = \frac{\max X - \min X}{L_s}$$

- ⊗ Si *initbh* está presente el algoritmo no ejecuta los 3 juegos de bordes iniciales.

## Opción “trymany=TRUE”

El algoritmo siempre se replica 5 (rep=5) veces, es decir si no existe *initbh*, se van alcanzar  $5*3=15$  (3 juegos de bordes iniciales) resultados de bordes y por último elegirá aquel que alcanzo el valor más chico de la función objetivo:

$$n(b_1, \dots, b_{Ls-1}) = N_{Ls} + \frac{\sum_{h=1}^{Ls-1} N_h^2 S_{Xh}^2 / a_h r_h}{cv_0^2 T^2 \pi_X + \sum_{h=1}^{Ls-1} N_h S_{Xh}^2}$$

**Advertencia:** a) el algoritmo puede no converger  
b) si lo hace, puede que sea a un mínimo local

Es por esto que antes de decretar los bordes como "definitivos" hay que experimentar con bordes iniciales alternativos y modificar los parámetros del algoritmo para evaluar su comportamiento

# Detalle de las Iteraciones

```
> Kozak1$iter.detail[1:10,]
    b1      b2      b3 opti.nh opti.nhnonint takeall step iter run
1 19261.67 32271.0 55037.33    145     143.0682      1   0   0   1
2 18053.00 32250.0 55154.00    144     142.5539      1 -13   2   1
3 18053.00 33067.5 55154.00    144     141.9126      1  10  11   1
4 18053.00 33067.5 55950.00    141     139.3720      1   5  14   1
5 18053.00 34022.5 55950.00    140     138.9618      1   8  15   1
6 18053.00 34022.5 56200.00    139     138.4327      1   1  25   1
7 18053.00 34022.5 73400.00    132     130.8000      1  41  30   1
8 18053.00 34022.5 70433.00    132     130.1391      1  -3  40   1
9 18053.00 41116.0 70433.00    131     130.0623      1  41  46   1
10 19216.50 41116.0 70433.00   130     128.8260      1  12  53   1
...  ...  ...  ...  ...  ...  ...  ...  ...  ...
```

# Detalle de los Bordes Iniciales y Finales por REP

```
> Kozak1$run.detail
```

	b1	b2	b3	opti.nh	opti.nhnonint	takeall	niter	ibh.type	ib1	ib2	ib3	rep
1	20395	38700.0	70433.0	127	126.5259	1	267	cumrootf	19261.667	32271.00	55037.33	1
2	20550	39000.0	72312.5	127	126.4984	1	707	cumrootf	19261.667	32271.00	55037.33	2
3	19700	37869.5	70980.5	127	126.5899	1	226	cumrootf	19261.667	32271.00	55037.33	3
4	20550	38700.0	70980.5	128	126.4631	1	730	cumrootf	19261.667	32271.00	55037.33	4
5	20900	38700.0	69815.0	127	126.5792	1	215	cumrootf	19261.667	32271.00	55037.33	5
6	20395	38550.0	70980.5	127	126.4967	1	316	geo	8552.321	24380.73	69503.94	1
7	20700	38550.0	70980.5	127	126.4773	1	189	geo	8552.321	24380.73	69503.94	2
8	20550	39000.0	72312.5	127	126.4984	1	352	geo	8552.321	24380.73	69503.94	3
9	20700	38550.0	70980.5	127	126.4773	1	720	geo	8552.321	24380.73	69503.94	4
10	20700	38550.0	70980.5	127	126.4773	1	418	geo	8552.321	24380.73	69503.94	5
11	20550	39000.0	72312.5	127	126.4984	1	356	robust	20413.000	37118.00	160397.00	1
12	20550	39000.0	72312.5	127	126.4984	1	650	robust	20413.000	37118.00	160397.00	2
13	20550	39000.0	72312.5	127	126.4984	1	319	robust	20413.000	37118.00	160397.00	3
14	19700	38200.0	70433.0	127	126.5928	1	191	robust	20413.000	37118.00	160397.00	4
15	20550	39000.0	72312.5	127	126.4984	1	155	robust	20413.000	37118.00	160397.00	5

## Comando var.strata(*estratificación, variablename*)

```
> #####
> # Evaluacion de la Estratificacion
> # en otras Variables de Estudio
> #####
> var.strata(Kozak1,VENTAS)
Given arguments:
strata = Kozak1
y = VENTAS
rh.postcorr = FALSE

Strata information:
      |      type   rh   Nh   nh   fh |      E(Y)  Var(Y)
stratum 1 | take-some 0.75 326  35 0.11 | 4221.97 5940517
stratum 2 | take-some 0.75 256  31 0.12 | 3682.98 4845126
stratum 3 | take-some 0.75 154  33 0.21 | 4016.92 5266700
stratum 4 |  take-all 0.75  28  28 1.00 | 3842.36 5481136
Total          764 127 0.17

Total sample size: 127
Anticipated population mean: 3986.122
Anticipated CV: 0.06469515
Note: CV=RRMSE (Relative Root Mean Squared Error) because takenone=0.
```

## Dado n Minimizar el cv

El algoritmo de Kozak se puede emplear para resolver el problema dual, dado un  $n$  minimizar el  $cv$  del estimador de  $\hat{T}_{HTx}$  asumiendo un Tipo de Asignación, con  $n$  "fijo" minimizar:

$$cv(b_1, \dots, b_{H-1}) = \left\{ \frac{\sum_{h=1}^{H-1} N_h^2(b_h) S_{xh}^2(b_h) \left( \frac{1}{a_h(b_h)n} - \frac{1}{N_h(b_h)} \right)}{(n - N_H(b_h)) T_{\pi x}^2} \right\}$$

Reemplazar en  $strata.LH$  a  $CV = cv_o$  por  $n = n_o$

# Si Hay Recursos para n=75

```
> #####
> # Dado n=n0
> # minimizar CV
> #####
> strata.LH(x=TAMANIO,n=75,Ls=4,alloc=c(0.5,0,0.5),takeall=1,
+ rh=0.75)
Given arguments:
x = TAMANIO
n = 75, Ls = 4, takenone = 0, takeall = 1
allocation: q1 = 0.5, q2 = 0, q3 = 0.5
model = none
algo = Kozak: minsol = 1000, idopti = nh, minNh = 2, maxiter = 10000,
           maxstep = 50, maxstill = 500, rep = 5, trymany = TRUE

Strata information:
      |   type   rh |   bh     E(Y)    Var(Y)   Nh nh   fh
stratum 1 | take-some 0.75 | 20550  13384.34  18228129 325 19 0.06
stratum 2 | take-some 0.75 | 39350   28152.76  24712750 260 18 0.07
stratum 3 | take-some 0.75 | 81190   53776.28  100686755 163 22 0.13
stratum 4 |  take-all 0.75 | 198141 113560.19 1134659673 16 16 1.00
Total                               764 75 0.10

Total sample size: 75
Anticipated population mean: 29125.82
Anticipated CV: 0.02836104
Note: CV=RRMSE (Relative Root Mean Squared Error) because takenone=0.
```

# Modelos de Discrepancia entre X e Y

El algoritmo de Kozak es muy flexible con respecto a otros.

Permite introducir relaciones entre la variable  $X$  y la de estudio,  $Y$ .

A través de imponer un modelo  $\xi$  entre ellas permite anticipar los momentos de la distribución de  $Y$  en cada estrato, condicional a  $X$ . Asumiendo alguna distribución para  $X$  o bien para los errores  $\varepsilon$ .

$$\text{Esperanza} \quad \bar{Y}_h = E_{\xi}(Y / b_{h-1} \leq X < b_h)$$

$$\text{Varianza} \quad S_{y_h}^2 = E_{\xi}(Y^2 / b_{h-1} \leq X < b_h) - E_{\xi}(Y / b_{h-1} \leq X < b_h)^2$$

Hasta el momento el algoritmo permite tres modelos (apuntado a encuestas económicas):

- ⊗ Loglineal bajo el supuesto de mortalidad de las unidad
- ⊗ Lineal con herosedasticidad en los errores de crecimiento exponencial
- ⊗ Reemplazo aleatorio

# Modelos Aceptados

⊗ *modelos loglineales con mortalidad ("loglinear")*

$$Y = \begin{cases} e^{\alpha + \beta * \log(X) + \epsilon} & \text{con probabilidad } p_h \\ 0 & \text{con probabilidad } 1 - p_h \end{cases}$$

con  $\epsilon \sim N(0, \sigma^2)$

⊗ *modelos lineales con heterocedasticidad ("linear")*

$$Y = \beta * X + \epsilon$$

con  $\epsilon \sim N(0, \sigma^2 * X^\gamma)$

⊗ *modelos con reemplazo aleatorio ("random")*

$$Y = \begin{cases} X & \text{con probabilidad } 1 - \epsilon \\ X^{new} & \text{con probabilidad } \epsilon \end{cases}$$

## Parámetro `model.control=list()`

*beta* : un valor numérico para la pendiente en "*loglinear*" o "*linear*"  
(por defecto =1)

*sig2* : un valor numérico para la varianza en "*loglinear*" o "*linear*"  
(por defecto =0)

*ph* : un vector o valor numérico para cada estrato de la probabilidad de  
sobrevida en "*loglinear*"(por defecto =1)

*gamma* : un valor numérico para el exponente de X en los residuos del  
modelo "*linear*" (por defecto =0)

*epsilon* : un valor numérico para la probabilidad de que Y tome un valor X  
aleatorio de la población en el modelo "*random*" (por defecto =0)

# Modelo Loglineal

⊗ *modelos loglineales con mortalidad ("loglinear")*

$$Y = \begin{cases} e^{\alpha + \beta \log(X) + \epsilon} & \text{con probabilidad } p_h \\ 0 & \text{con probabilidad } 1 - p_h \end{cases}$$

$$Y = e^\alpha X^\beta e^{\epsilon}$$

$p_h$  es la probabilidad de sobrevida que crece a medida que  $X$  aumenta

Este modelo es muy empleado en muestras a unidades económicas en donde entre la etapa de diseño y la ejecución de la encuesta  $Y$  es 0 con probabilidad no nula.

Diferencias entre  $Y$  y  $X$  también se contempla en la perturbación  $\epsilon$  del modelo, en la escala logarítmica con  $\epsilon \sim N(0, \sigma^2)$

# Opción model="loglinear"

```
> # Model="loglinear"
> KozakLog=strata.LH(x=TAMANIO,CV=0.03,Ls=4,alloc=c(0.5,0,0.5),takeall=0,
+ rh=0.8,model="loglinear",
+ model.control=list(beta=1,sig2=0.05,ph=0.98))
> KozakLog
Given arguments:
x = TAMANIO
CV = 0.03, Ls = 4, takenone = 0, takeall = 0
allocation: q1 = 0.5, q2 = 0, q3 = 0.5
model = loglinear: beta = 1, sig2 = 0.05, ph = 0.98 0.98 0.98 0.98
algo = Kozak: minsol = 1000, idopti = nh, minNh = 2, maxiter = 10000,
           maxstep = 50, maxstill = 500, rep = 5, trymany = TRUE

Strata information:
      |      type   ph   rh |      bh      E(Y)      Var(Y)  Nh  nh   fh
stratum 1 | take-some 0.98 0.8 | 18873.0 12114.09  24648824 280  25 0.09
stratum 2 | take-some 0.98 0.8 | 37666.5 26108.84  77875887 298  47 0.16
stratum 3 | take-some 0.98 0.8 | 72312.5 50608.16 266638078 159  46 0.29
stratum 4 | take-some 0.98 0.8 | 198141.0 95852.02 1733454841  27  20 0.74
Total                               764 138 0.18

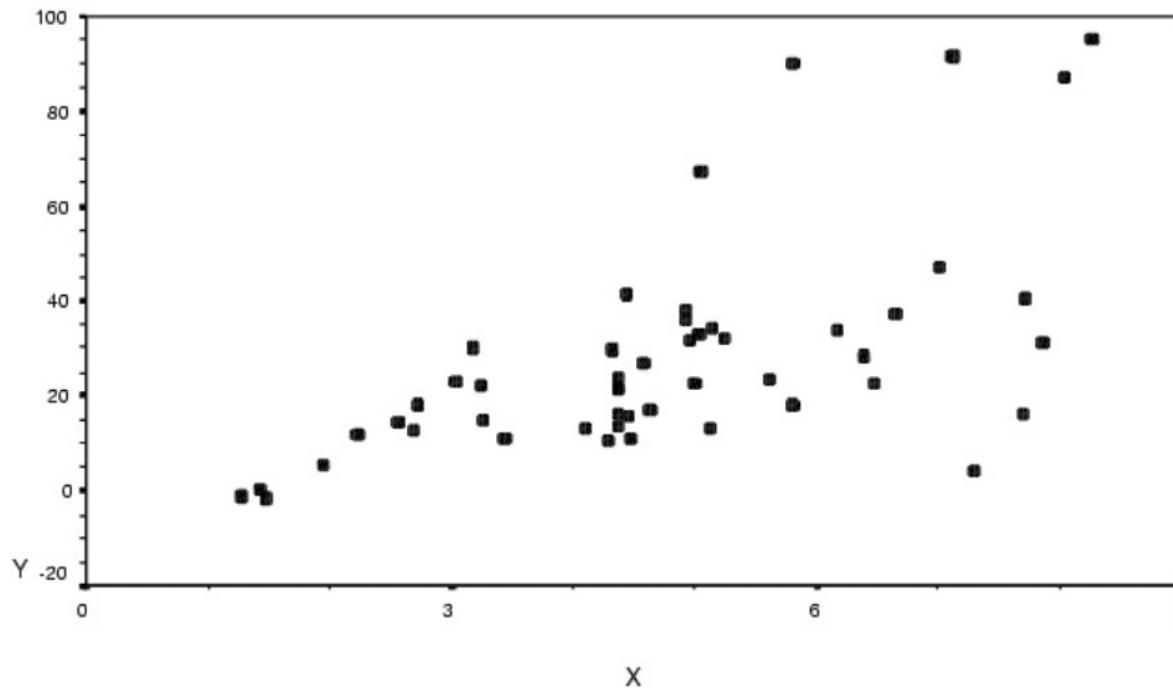
Total sample size: 138
Anticipated population mean: 28543.3
Anticipated CV: 0.02983858
Note: CV=RRMSE (Relative Root Mean Squared Error) because takenone=0.
> KozakLog$bh
[1] 18873.0 37666.5 72312.5
```

# Modelo Lineal

⊗ *modelos lineales con heterocedasticidad ("linear")*

$$Y = \text{beta} * X + \text{epsilon}$$

$$\text{con } \text{epsilon} \sim N(0, \text{sig2} * X^{\gamma})$$



## Opción model="linear"

```
> # Model="linear"
> KozakReg=strata.LH(x=TAMANIO,CV=0.03,Ls=4,alloc=c(0.5,0,0.5),takeall=1,
+ rh=0.8,model="linear",
+ model.control=list(beta=1, sig2=1.8,gamma=1.7))
> KozakReg
Given arguments:
x = TAMANIO
CV = 0.03, Ls = 4, takenone = 0, takeall = 1
allocation: q1 = 0.5, q2 = 0, q3 = 0.5
model = linear: beta = 1, sig2 = 1.8, gamma = 1.7
algo = Kozak: minsol = 1000, idopti = nh, minNh = 2, maxiter = 10000,
           maxstep = 50, maxstill = 500, rep = 5, trymany = TRUE

Strata information:
      |      type   rh |     bh      E(Y)    Var(Y)  Nh  nh   fh
stratum 1 | take-some 0.8 | 19150  12545.90  32004046 288  29  0.10
stratum 2 | take-some 0.8 | 38550  27006.15  90808588 294  49  0.17
stratum 3 | take-some 0.8 | 81190  53506.83  303572178 166  51  0.31
stratum 4 | take-all  0.8 | 198141 113560.19 1876895427  16   16  1.00
Total                               764 145  0.19

Total sample size: 145
Anticipated population mean: 29125.82
Anticipated CV: 0.0298394
Note: CV=RRMSE (Relative Root Mean Squared Error) because takenone=0.
> KozakReg$bh
[1] 19150 38550 81190
```

# Modelo de Reemplazo

- ⊗ *modelos con reemplazo aleatorio ("random")*

$$Y = \begin{cases} X & \text{con probabilidad } 1 - \text{epsilon} \\ X^{new} & \text{con probabilidad } \text{epsilon} \end{cases}$$

Este modelo permite que la variable  $Y$  tome un nuevo valor  $X^{new}$  con probabilidad  $\text{epsilon}$

$X^{new}$  es una variable aleatoria con función de densidad  $f(x)$  independiente  $X$

## Opción model="random"

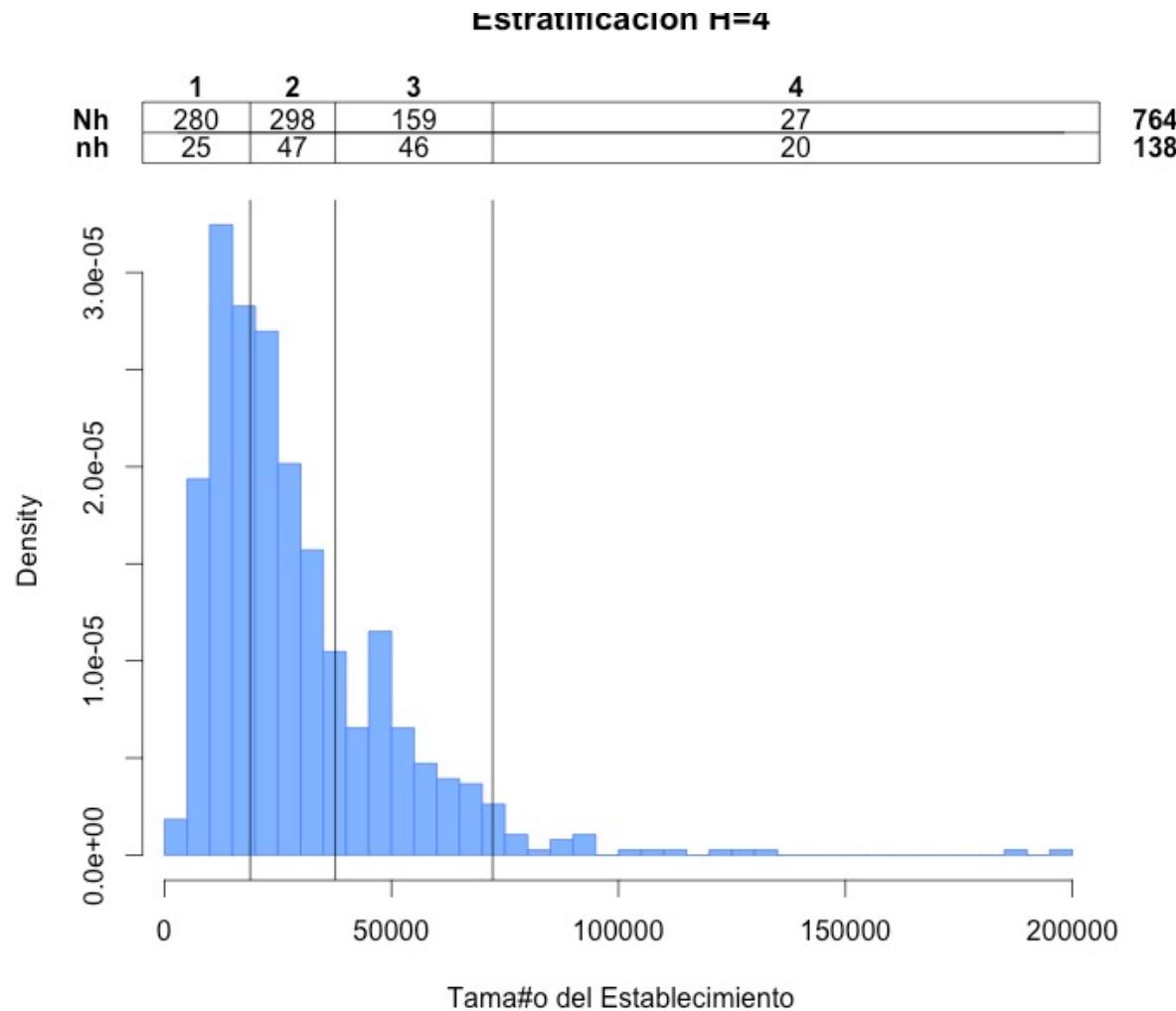
```
> # Model="random"
> KozakR=strata.LH(x=TAMANIO,CV=0.03,Ls=4,alloc=c(0.5,0,0.5),takeall=0,
+ rh=0.98,model="random",model.control=list(epsilon=0.03))
> KozakR
Given arguments:
x = TAMANIO
CV = 0.03, Ls = 4, takenone = 0, takeall = 0
allocation: q1 = 0.5, q2 = 0, q3 = 0.5
model = random: epsilon = 0.03
algo = Kozak: minsol = 1000, idopti = nh, minNh = 2, maxiter = 10000,
        maxstep = 50, maxstill = 500, rep = 5, trymany = TRUE

Strata information:
      |      type    rh |      bh      E(Y)      Var(Y)   Nh nh   fh
stratum 1 | take-some 0.98 | 20463.0 13835.28  38197260 324 24 0.07
stratum 2 | take-some 0.98 | 39000.0 28112.18  37143823 260 19 0.07
stratum 3 | take-some 0.98 | 72312.5 51471.49  100468747 153 19 0.12
stratum 4 | take-some 0.98 | 198141.0 95747.71 1153640916  27 11 0.41
Total                               764 73 0.10

Total sample size: 73
Anticipated population mean: 29125.82
Anticipated CV: 0.02984696
Note: CV=RRMSE (Relative Root Mean Squared Error) because takenone=0.
> KozakR$bh
[1] 20463.0 39000.0 72312.5
```

# Comando adicional plot()

```
> ### Comandos Adicionales #####
> plot(KozakLog,main="Estratificacion H=4",xlab="Tamaño del Establecimiento")
> |
```



# Agregar la Estratificación Alcanzada al Marco

```
> #####
> # Actualizacion de la Base
> #####
> Estrato=KozakLog$stratumID
> base=cbind(Base.Productores,Estrato)
> Ponderacion=KozakLog$Nh/KozakLog$nh
> Strato.id=unique(KozakLog$stratumID)
> resumen=cbind.data.frame(Strato.id,KozakLog$Nh,KozakLog$nh,Ponderacion)
> colnames(resumen)=c("Estrato","Nh","nh","Ponderacion")
> resumen
  Estrato Nh nh Ponderacion
1       1 280 25   11.200000
2       2 298 47    6.340426
3       3 159 46    3.456522
4       4  27 20    1.350000
> base=merge(base,resumen,by="Estrato")
```

## Selección de la Muestra Según los $n_h$ Óptimos

```
> #####
> library(sampling)
> seleccion=strata(base,stratanames="Estrato",size=KozakLog$nh,method="systematic",
+ pik=1/base$Ponderacion)
> muestral1=getdata(base,seleccion)
```

# TAMAÑO Y ASIGNACIÓN EN EL CASO MULTIVARIADO

- **Máxima:** “Una muestra es aceptable si los errores muestrales están por debajo de los límites predefinidos y los costos son sustentables”
- **Propuesta:** Sea “U” con “p” variables auxiliares (X) que se emplean para estratificar y restricciones de presición sobre “G” variables objetivo (Y) se busca en forma conjunta:
  - a) La mejor estratificación, o sea, la mejor partición de U
  - b) El menor tamaño de muestra y con una asignación de las unidades a los estratos que permitan satisfacer las restricciones de presición sobre los estimadores.
  - c) Al menor costo posible

## ALGORITMO DE BETHEL PARA (B) Y (C)

- Dada una estratificación sobre U, una función lineal de costos conocidos y un conjunto de restricciones en términos de cv dados sobre las G variables objetivo se busca minimizar,

$$\left\{ \begin{array}{l} \min \left\{ f(n_1, \dots, n_H) \right\} = \min \left\{ C_0 + \sum_{h=1}^H c_h n_h \right\} \\ \text{sujeta a } \left( CV_{yg} \right)^2 \leq cv_{yg}^2 \quad g = 1, \dots, G \end{array} \right.$$

$$\sum_{h=1}^H \frac{N_h^2 S_{hyg}^2}{\left( t_{yg}^2 cv_{yg}^2 + \sum_{h=1}^H N_h S_{hyg}^2 \right) n_h} \leq 1$$

Prog Lineal Convexa

$$\left\{ \begin{array}{l} \min f(n_1, \dots, n_H) \\ \sum_{h=1}^H \frac{a_{hg}}{n_h} \leq 1 \quad g = 1, \dots, G \\ 0 < \frac{1}{n_h} \quad h = 1, \dots, H \end{array} \right.$$