Muestreo: Primer Trabajo Práctico

Preparación

Cargamos las librerías que planeamos usar e importamos los marcos de datos necesarios.

```
library("tidyverse")

library("knitr")

library("stratification")

library("PracTools")

library("sampling")

library("lmtest")

library("here")

set_here()
```

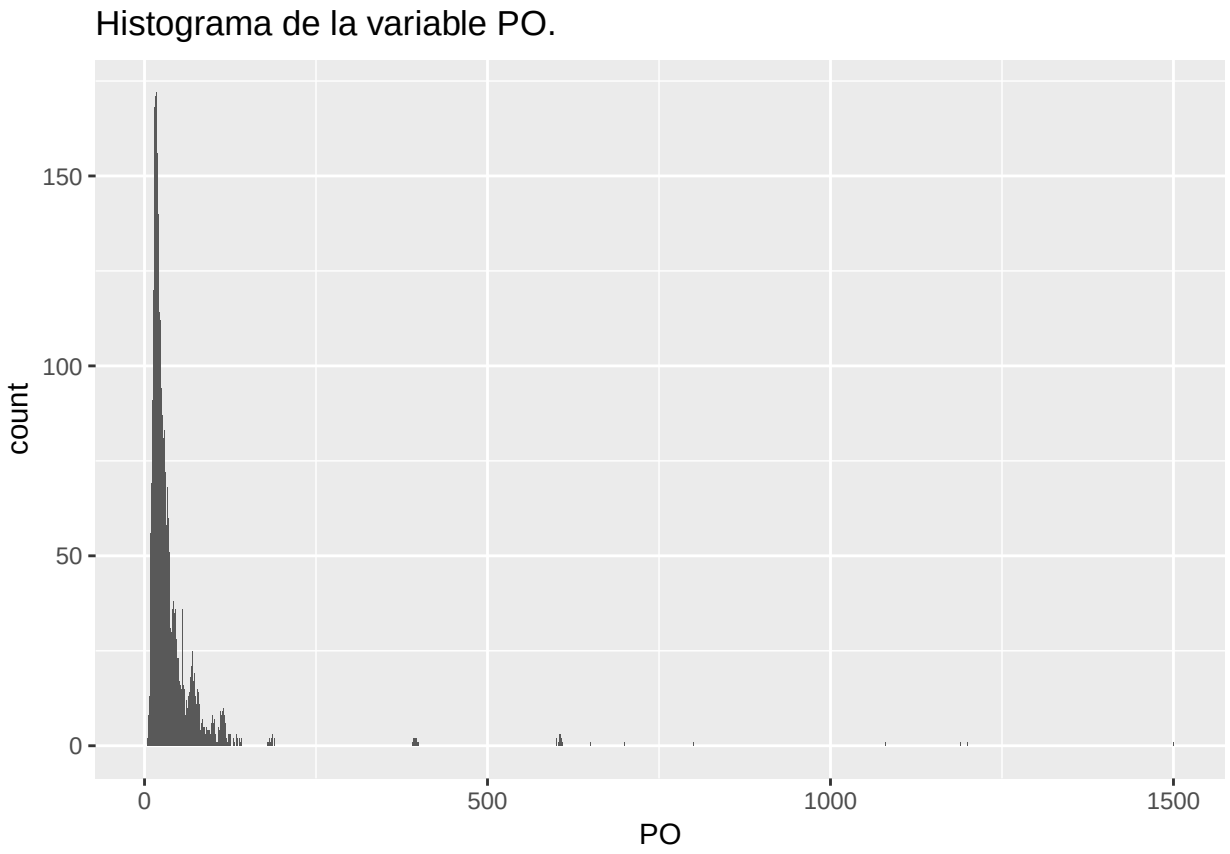
```
## File .here already exists in /home/octavio/Muestreo/Entrega del TP1
```

```
load("Marco.PO.RData")

load("Prueba.Piloto.RData")

marcoAuxiliar  <- as_tibble(Marco.PO)

piloto  <- as_tibble(Prueba.Piloto)

set.seed(42)
```

1 - Muestreo Aleatorio Simple

a Estudiar la simetría de la variable PO a través de gráficos y medidas.

```
marcoAuxiliar %>% ggplot()+
  aes(x = PO)+
  geom_bar() +
  ggtitle("Histograma de la variable PO.")
```



El primer gráfico ya revela que tenemos toda clase de inconvenientes. Hay asimetría y varios grupos de datos claramente separados.

Buscamos la media y el desvío estándar de esta variable, como ya sabemos que va a ser alta, buscamos además la mediana y el máximo para darnos una mejor idea de su comportamiento.

```
PO  <- marcoAuxiliar$PO

medidasPO  <- tibble(
  media = mean(PO),
  max = max(PO),
  min = min(PO),
  mediana = median(PO),
  varianza = var(PO),
  "rango inter-cuartil" = IQR(PO),
  DMV = mad(PO),
  CV = sqrt(varianza) / media
)

kable(medidasPO, caption = "Algunos Indicadores de la Variable PO.")
```

Table 1: Algunos Indicadores de la Variable PO.

media	max	min	mediana	varianza	rango inter-cuartil	DMV	CV
38.265562249	1500	4	24	4160.92016951	24	13.3434	1.68572447684

Observaciones

Quizás uno de los datos más llamativos sea la diferencia entre la media y la mediana, de un -61.734437751004%. El coeficiente de variación ampliamente superior a *100%* tampoco promete un camino fácil.

b Calcular el tamaño de muestra necesario para obtener los *CV* fijados y la respuesta considerada estableciendo un muestreo simple al azar.

Estos coeficientes de variación son respecto a las medias, por lo que vamos a tomar una fórmula que se basa en asumir que la variable bajo estudio es normal, establecer un intervalo de confianza y usarlo para despejar como variable el tamaño de la muestra. En este caso, consideramos que la variable auxiliar *PO* es válida para dar cuenta de la homogeneidad en el universo bajo estudio.

El paquete PracTools tiene ya incorporada esta fórmula.

```
nPorCVIdeales  <-  tibble(
  "5%"  = nCont(CV0=0.05,CVpop=medidasPO$CV),
  "3%"  = nCont(CV0=0.03,CVpop=medidasPO$CV),
  "1%"  = nCont(CV0=0.01,CVpop=medidasPO$CV)
)

kable(nPorCVIdeales, caption = "Tamaños para un MAS sin considerar la no respuesta.")
```

Table 2: Tamaños para un MAS sin considerar la no respuesta.

5%	3%	1%
1136.66680473	3157.40779091	28416.6701182

Como cabe esperar dado que ya sabemos que la variable auxiliar tiene un comportamiento muy poco idóneo, incluso en condiciones ideales (todavía no hicimos intervenir la no respuesta) se requiere muestrear aproximadamente un tercio de la población para obtener el coeficiente de variación más laxo considerado y el de 1% exige un tamaño de muestra superior a N siento teóricamente imposible. El CV de 3% está cerca del total, por lo que va a exigir un censo al añadir esta consideración.

```
nPorCV  <- tibble(
  "5%"  = ( 1 / 0.85 ) * nCont(CV0=0.05,CVpop=medidasPO$CV),
  "3%"  = ( 1 / 0.85 ) * nCont(CV0=0.03,CVpop=medidasPO$CV),
  "1%"  = ( 1 / 0.85 ) * nCont(CV0=0.01,CVpop=medidasPO$CV)
)

kable(nPorCV, caption = "Tamaños para un MAS, considerando la no respuesta.")
```

Table 3: Tamaños para un MAS, considerando la no respuesta.

5%	3%	1%
1337.25506439	3714.59740108	33431.3766097

c Especificar qué fórmula se usa para estimar el tamaño de muestra en el caso de dominios y calcular este suponiendo tres valores posibles para el CV_y en el caso de cada dominio particular y 3 dominios cada uno con una proporción distinta de unidades respecto al universo.

Realizando un desarrollo similar al indicado antes y considerando el trabajo en dominios, obtenemos el tamaño de muestra con este cálculo:

$n = \frac{1-P}{P} \cdot \frac{1}{CV_0{}^2}$

Este cálculo puede realizarse en forma automática empleando el paquete `PracTools` mediante el comando `nPropCont` .

```
cvs  <-  c( "5%"=0.05, "3%" =0.03, "1%"= 0.01)

proporciones  <-  c("3/4" = 0.75, "4/10" = 0.4, "1/10" = 0.1)

dominiosPorCVyP  <- tibble(
  CV = rep(cvs,3),
  proporciones = rep(proporciones, each=3)
)

dominiosPorCVyP$n <- map2_dbl(
  dominiosPorCVyP$CV,
  dominiosPorCVyP$proporciones,
  ~nProp(CV0=.x,
         pU=.y,
         N=3900)
  )

kable(dominiosPorCVyP, caption = "Tamaños de Muestra para los dominios según su representatividad y el C
```

Table 4: Tamaños de Muestra para los dominios según su representatividad y el CV deseado.

CV	proporciones	n
0.05	0.75	128.957592792
0.03	0.75	338.327275251
0.01	0.75	1797.483523068
0.05	0.40	520.115581240
0.03	0.40	1167.874468467
0.01	0.40	3095.401873115
0.05	0.10	1872.249633284
0.03	0.10	2805.957263112
0.01	0.10	3738.058978264

d A partir de este punto, se considera diseños estratificados. Estratificando según el tamaño según el criterio indicado, calcular el tamaño de muestra para la estimación de la media de la variable *PO* asumiendo adjudicación proporcional y de Neyman para los *CV* y la tasa de respuesta considerada.

```
asignarEstrato <- function(x) {
  ifelse(x<10,"Pequeña",
                ifelse(x<35,"Mediana","Grande")
  )
}

marcoAuxiliar %>% mutate(estrato = asignarEstrato(PO) ) -> marcoAuxiliar
```

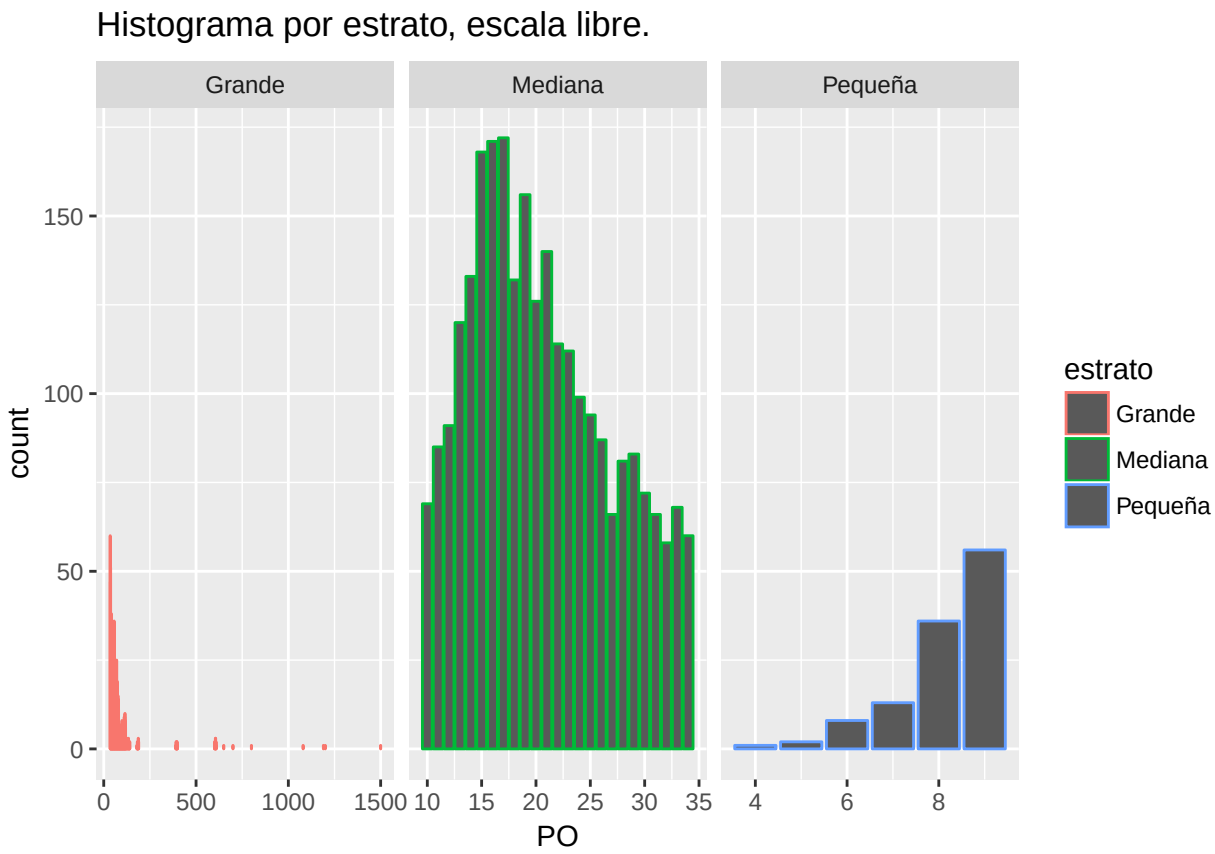
Observamos las 30 primeras filas para ver el formato de la tabla obtenida.

Table 5: Fracción del Marco Estratificado según el Criterio Dado.

Uk	PO	estrato
1	33	Mediana
2	26	Mediana
3	62	Grande
4	51	Grande
5	57	Grande
6	29	Mediana
7	77	Grande
8	56	Grande
9	13	Mediana
10	68	Grande
11	35	Grande
12	22	Mediana
13	62	Grande
14	44	Grande
15	56	Grande
16	607	Grande
17	79	Grande
18	75	Grande
19	31	Mediana
20	49	Grande
21	45	Grande
22	12	Mediana
23	86	Grande
24	31	Mediana
25	33	Mediana
26	19	Mediana
27	27	Mediana
28	26	Mediana
29	134	Grande
30	40	Grande

Además, podemos proponer un gráfico por estrato para ver si esta estratificación explica mejor los fenómenos observados:

```
#marcoAuxiliar %>% mutate(PO = as.factor(PO)) -> marcoAuxiliarFac

marcoAuxiliar %>% ggplot( aes(x=PO, color = estrato) ) +
     geom_bar() +
  facet_grid(.~estrato,scales="free") +
  ggtitle("Histograma por estrato, escala libre.")
```



por estrato-1.bb

El gráfico muestra que el estrato más complejo y el que aporta mayores dificultades es el de empresas categorizadas como grandes. Ellas muestran una gran dispersión. De hecho, parece observarse que la gran mayoría se concentra antes de los 250 empleados, con lo cual quizás sería conveniente una categoría más. Es posible que esta necesidad termine siendo capturada cuando establezcamos un estrato autorepresentado.

```
varianzaDelEstrato  <- function(df, colEstratos, estrato, colVariable) {
  indices  <- c( which(df[colEstratos]==estrato) )
  tablaEstrato  <- df[indices,]
  vectorDatos  <- tablaEstrato[colVariable]
  return( var(vectorDatos) )
}

totalDelEstrato  <- function(df, colEstratos, estrato) {
  indices  <- c( which(df[colEstratos]==estrato) )
  return( length(indices) )
}

varianzasPorEstrato  <- function(df,colEstratos,colVariable) {
  estratos  <- ( unique(df[colEstratos]) )[[1]]
  varianzas  <- map_dbl(estratos,
```

```
    ~varianzaDelEstrato(marcoAuxiliar,
                        colEstratos,
                        .x,
                        colVariable)
  )

  names(varianzas)  <- estratos
  return(varianzas)
}

totalesPorEstrato  <- function(df,colEstratos) {
  estratos  <- ( unique(df[colEstratos]) )[[1]]
  totales  <- map_dbl(estratos,~totalDelEstrato(df,colEstratos,.x))
  names(totales)  <- estratos
  return(totales)
}

vpE  <- varianzasPorEstrato(marcoAuxiliar,"estrato","PO")

tpE  <- totalesPorEstrato(marcoAuxiliar,"estrato")

asignarNeyman  <- function(df,colEstratos,colVariable,n) {
  varianzas  <- varianzasPorEstrato(df,colEstratos,colVariable)
  totales  <- totalesPorEstrato(marcoAuxiliar,"estrato")
  suma  <- sum(varianzas * totales)
  asignaciones  <- map2(varianzas,totales,~(.x * n * .y) / suma)
  return(asignaciones)
}

asignarProporcional  <- function(df,colEstratos,n) {
  N  <- length(df[[1]])
  totales  <- totalesPorEstrato(marcoAuxiliar,"estrato")
  return( map(totales,~(.x * n) / N)
  )


listaCVs  <- c("5%"=0.05,"3%"=0.03,"1%"=0.01)
}
```

Recordamos que la asignación de Neyman es de valor teórico ya que depende de conocer los datos que se desea obtener. Es por esta razón que tiene sentido tomar *PO*. Dado que la varianza es el criterio fundamental de esta asignación y es drásticamente mayor en el estrato de empresas grandes, la muestra es acaparada po ese estrato. Con la asignación proporcional, obtuvimos un resultado mucho más acorde a lo que se intuiría con un razonamiento ingenuo. Esta muestra va a representar mejor los estratos según su tamaño (después de todo, ese es su criterio rector) pero no va a explicar tan precisamente la varianza de la población en estudio. Este caso es un poco extremo y cabe preguntarse si en el caso de estar limitados a esta clase de muestreo no convendría la asignación proporcional dado que parece insensato ignorar a las empresas pequeñas y medianas a favor de conocer que pasa con las de mayor tamaño. En ambos casos es imposible plantear CV menores al 5%. En el caso de la asignación proporcional la muestra es una fracción muy considerable del total y en el caso de Neyman se presenta el desbalance extremo del que se habló antes.

```
listaCVs  <- c("5%"=0.05,"3%"=0.03,"1%"=0.01)

CV  <-  names(nPorCV)
```

```
nNeyman  <-  map_df(nPorCV,~asignarNeyman(marcoAuxiliar,"estrato","PO",.x) ) %>% mutate( CV = listaCVs )

nProp  <-  map_df(nPorCV,~asignarProporcional(marcoAuxiliar,"estrato",.x) ) %>% mutate( CV = listaCVs )
```

Table 6: Asignación de Neyman para la estratificación en 3 categorías.

Mediana	Grande	Pequeña	CV
10.6884679106	1326.55351265	0.013083829274	0.05
29.6901886406	3684.87086847	0.036343970206	0.03
267.2116977653	33163.83781619	0.327095731854	0.01

Table 7: Asignación Proporcional para la estratificación en 3 categorías.

Mediana	Grande	Pequeña	CV
880.426715333	417.892207621	38.936141433	0.05
2445.629764815	1160.811687836	108.155948425	0.03
22010.667883336	10447.305190527	973.403535824	0.01

e Métodos de Estratificación en Variales Asimétricas

Geométrico

Asignación de Neyman

```
allocNey  <- c(q1=0.5,q2=0,q3=0.5)

allocProp  <- c(q1=0.5,q2=0,q3=0)

listaCVs  <- c("5%"=0.05,"3%"=0.03,"1%"=0.01)

tresEstratos  <- c("P","M","G")

cuatroEstratos  <- c("1","2","3","4")

cincoEstratos  <- c("1","2","3","4","5")

asigGeoNey  <- function(cv,l) {strata.geo(x=marcoAuxiliar$PO,CV=cv,Ls = l, alloc = allocNey,rh = 0.85 )


listaCVs  <- c("5%"=0.05,"3%"=0.03,"1%"=0.01)

nGeoNey3  <- map_df(listaCVs,~(asigGeoNey(.x,3))[["nh"]] ) %>% mutate(estrato=tresEstratos)

nGeoNey4  <- map_df(listaCVs,~(asigGeoNey(.x,4))[["nh"]] ) %>% mutate(estrato=cuatroEstratos)

nGeoNey5  <- map_df(listaCVs,~(asigGeoNey(.x,5))[["nh"]] ) %>% mutate(estrato=cincoEstratos )

geometricoNeyman  <- list(nGeoNey3,nGeoNey4,nGeoNey5)
```

```r
asigProp  <- function(cv,l) {strata.geo(x=marcoAuxiliar$PO,CV=cv,Ls = l, alloc = allocProp ) }

nGeoProp3  <- map_df(listaCVs,~(asigProp(.x,3))[["nh"]] ) %>% mutate(estrato=tresEstratos)

nGeoProp4  <- map_df(listaCVs,~(asigProp(.x,4))[["nh"]] ) %>% mutate(estrato=cuatroEstratos)

nGeoProp5  <- map_df(listaCVs,~(asigProp(.x,5))[["nh"]] ) %>% mutate(estrato=cincoEstratos)

geometricoProporcional  <- list(nGeoProp3,nGeoProp4,nGeoProp5)

asigLHNey  <- function(cv,l,respuesta) {strata.LH(
                                x=marcoAuxiliar$PO,
                                CV=cv,
                                Ls = l,
                                alloc = allocNey,
                                rh = respuesta,
                                algo="Kozak",
                                takeall = 1,
                                )
}

nKozNey3  <- map_df(listaCVs,~(asigLHNey(.x,3,c(0.85,0.85,1)))[["nh"]] ) %>% mutate(estrato=tresEstratos

nKozNey4  <- map_df(listaCVs,~(asigLHNey(.x,4,c(0.85,0.85,0.85,1)))[["nh"]] ) %>% mutate(estrato=cuatroE

nKozNey5  <- map_df(listaCVs,~(asigLHNey(.x,5,c(0.85,0.85,0.85,0.85,1)))[["nh"]] ) %>% mutate(estrato=c

kozakNeyman  <- list(nKozNey3,nKozNey4,nKozNey5)

agregarTotales  <- function(asignacion) {
  cols <- ncol(asignacion)
  totales <- map(
    seq( 1,cols - 1),
    ~sum( asignacion[.x] )
  )
  totales  <- c(totales,"Total")
  asignacion[nrow(asignacion)+1,]  <- totales
  return( asignacion )
}

asigLHProp  <- function(cv,l,respuesta) {strata.LH(
                                        x=marcoAuxiliar$PO,
                                        CV=cv,
                                        Ls = l,
                                        alloc = allocProp,
                                        rh = respuesta,
                                        algo="Kozak",
                                        takeall = 1,
                                        )
}

nKozProp3  <- map_df(listaCVs,~(asigLHProp(.x,3,c(0.85,0.85,1)))[["nh"]] ) %>% mutate(estrato=tresEstra
```

Table 8: Asignación de Neyman con optimización geométrica.

5%	3%	1%	estrato
16	39	257	P
57	141	932	M
12	30	36	G
85	210	1225	Total

5%	3%	1%	estrato
4	9	68	1
46	115	865	2
8	18	135	3
11	27	36	4
69	169	1104	Total

5%	3%	1%	estrato
1	3	17	1
19	50	357	2
20	51	369	3
4	9	30	4
6	14	22	5
50	127	795	Total

Table 9: Asignación Proporcional con optimización geométrica.

5%	3%	1%	estrato
139	350	1430	P
97	242	991	M
3	6	23	G
239	598	2444	Total

5%	3%	1%	estrato
59	149	650	1
134	340	1487	2
13	33	144	3
2	5	21	4
208	527	2302	Total

5%	3%	1%	estrato
19	49	243	1
101	262	1300	2
35	89	441	3
2	4	16	4
1	3	12	5
158	407	2012	Total

```
nKozProp4  <- map_df(listaCVs,~(asigLHProp(.x,4,c(0.85,0.85,0.85,1)))[["nh"]] ) %>% mutate(estrato=cuat

nKozProp5  <- map_df(listaCVs,~(asigLHProp(.x,5,c(0.85,0.85,0.85,0.85,1)))[["nh"]] ) %>% mutate(estrato=


kozakProp  <- list(nKozProp3,nKozProp4,nKozProp5)

geometricoNeyman  <- map(geometricoNeyman,agregarTotales)

geometricoProporcional  <- map(geometricoProporcional,agregarTotales)

kozakNeyman <- map(kozakNeyman,agregarTotales)

kozakProp  <- map(kozakProp,agregarTotales)
```

Disponemos de listas con los resultados por Cv de todos los métodos de adjudicación para todas las cantidades de estratos propuestas. A ellas, añadimos una última fila de totales ya que dado que está fijo el coeficiente de variación el parámetro de eficiencia es el tamaño de muestra demandado.

```
kable(geometricoNeyman, caption = "Asignación de Neyman con optimización geométrica.")

kable(geometricoProporcional, caption = "Asignación Proporcional con optimización geométrica.")

kable(kozakNeyman, caption = "Asignación de Neyman con optimización de Kozak.")

kable(kozakProp, caption = "Asignación Proporcional con optimización de Kozak.")
```

Recordamos que la estimación empleando adjudicación de Neyman es optimista, en el sentido en que dado que no conocemos la varianza de las variables en estudio, sólo va a ser certera cuando la correlación que ellas presentan con la variable auxiliar sea perfecta. La asignación proporcional es en cambio una estimación conservadora, lograda empleando la mínima información que es razonable presuponer (el tamaño de los estratos y de la población). Así, generar ambas estimaciones del tamaño de muestra nos dota de un rango: podemos interpretarlo con enunciados del tipo "Lograr un coeficiente de variación del 5% con optimización

Table 10: Asignación de Neyman con optimización de Kozak.

5%	3%	1%	estrato
25	61	252	P
26	57	211	M
36	50	248	G
87	168	711	Total

5%	3%	1%	estrato
8	22	112	1
7	20	89	2
10	26	90	3
36	36	158	4
61	104	449	Total

5%	3%	1%	estrato
9	13	75	1
8	11	51	2
11	9	41	3
3	7	82	4
4	36	78	5
35	76	327	Total

Table 11: Asignación Proporcional con optimización de Kozak.

5%	3%	1%	estrato
54	118	382	P
11	24	104	M
36	50	264	G
101	192	750	Total

5%	3%	1%	estrato
25	56	153	1
7	16	78	2
2	4	34	3
36	50	212	4
70	126	477	Total

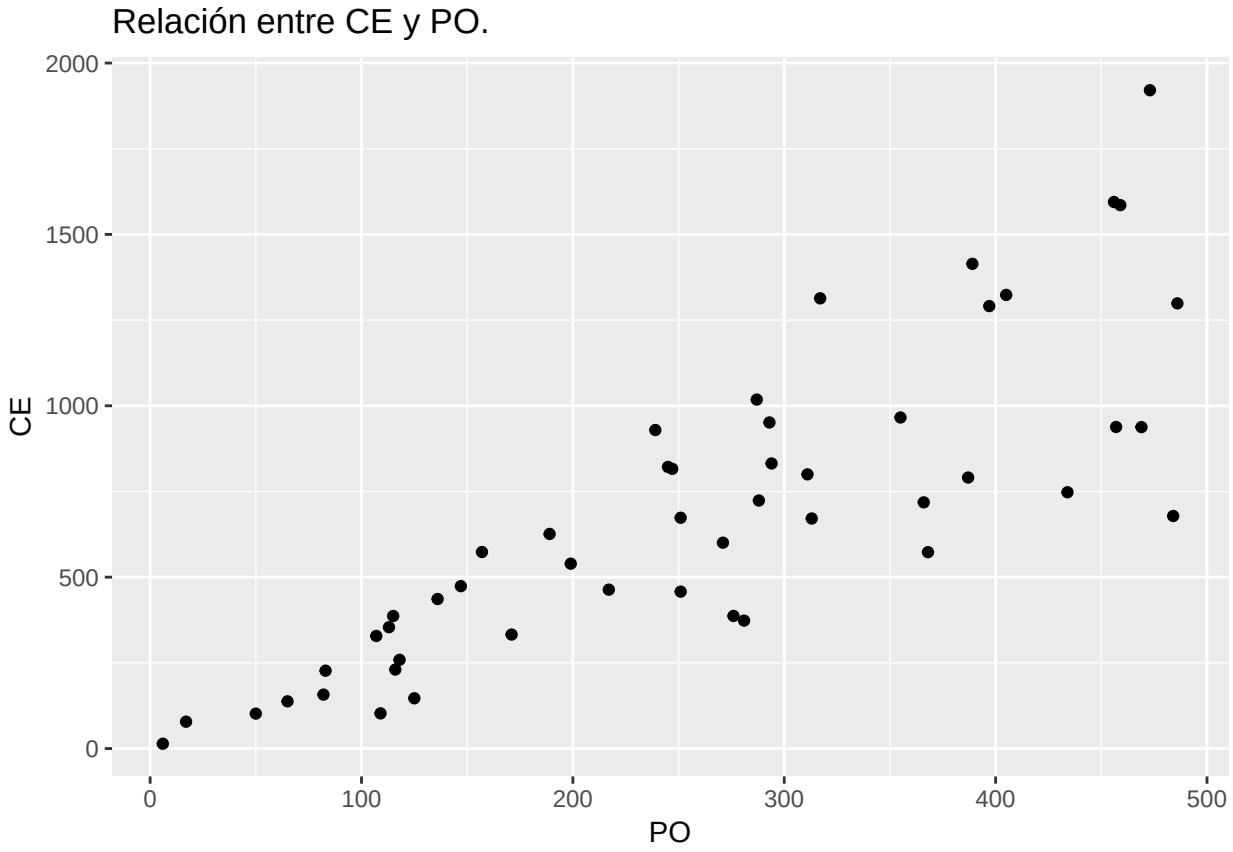
5%	3%	1%	estrato
11	21	164	1
6	11	83	2
3	5	35	3
1	2	12	4
36	50	67	5
57	89	361	Total

geométrica nos va a demandar entre 50 y 158 muestras si se emplea 5 estratos".

En los tamaños de las asignaciones es muy evidente la mejora al incorporar el método de optimización de Kozak así como al incrementar la cantidad de estratos (un fenómeno que ya se adivinaba con las primeras ojeadas al gráfico). La reducción en los n en todos los casos es muy evidente respecto a las aproximaciones menos sofisticadas que intentamos primero. Con la optimización de Kozak en particular pasamos a considerar que se puede lograr un CV del 1% con entre 361 y 367 muestras, cuando con los demás estábamos ante valores superiores a N.

f

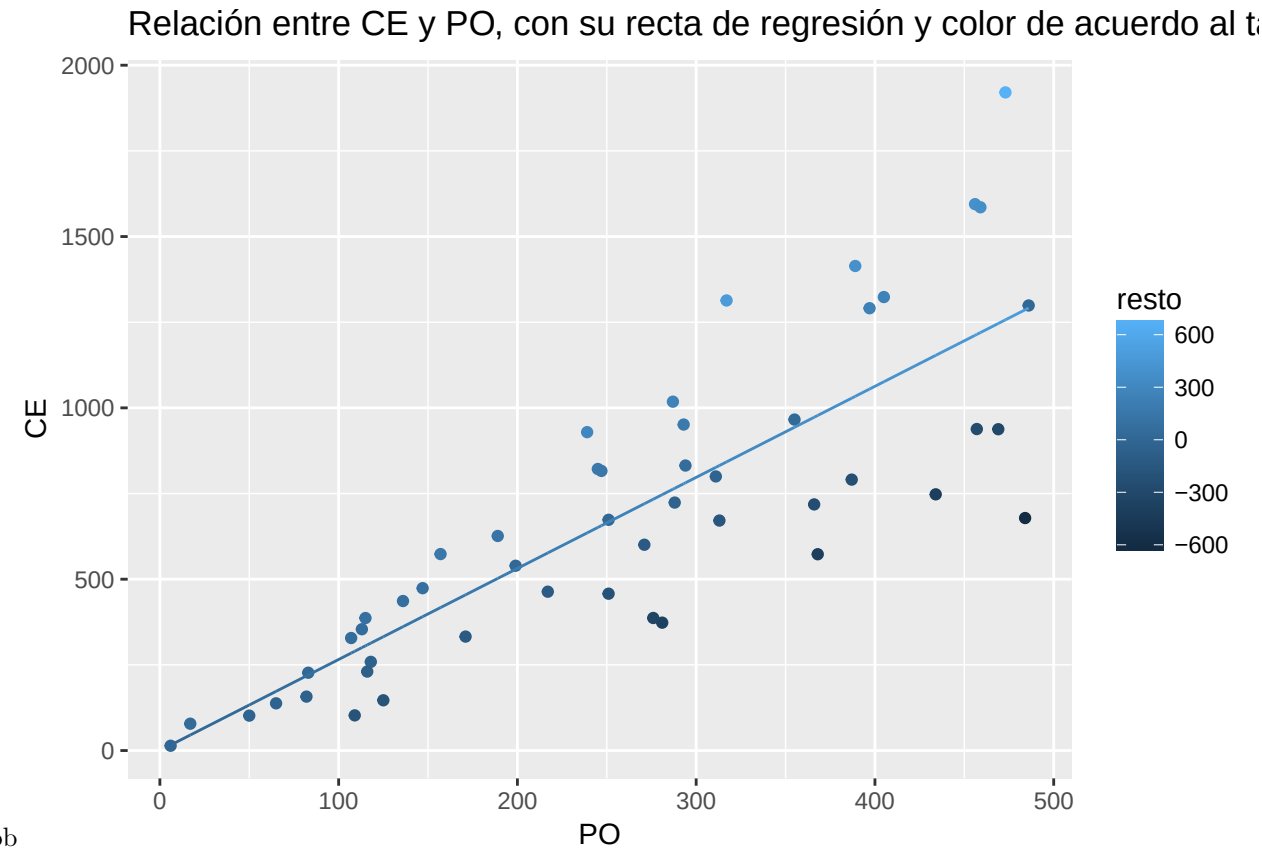
```
piloto %>%
  ggplot( aes( x=PO, y=CE ) ) +
  geom_point() +
  ggtitle("Relación entre CE y PO.")
```



Este primer gráfico parece adelantar que efectivamente la dispersión es dependiente de *PO* por su característica forma de haz de reflector.

Podemos agregar una recta de regresión al gráfico para evidenciar más el fenómeno antes de recurrir a la prueba de Breusch y Pagan propuesta.

```
modelo  <- lm(CE~PO+0,data=piloto)

betaObservado  <- modelo[["coefficients"]] %>% unname

recta  <- function(po) {
 return(betaObservado * po)
}

piloto %>% mutate(CErecta = recta(PO),resto = CE - CErecta)  -> marcoPiloto

marcoPiloto %>%
 ggplot( aes( x=PO, y=CE,color=resto) ) +
 geom_point() +
 geom_line(data = marcoPiloto, aes(x=PO,y=CErecta,color=PO) ) +
 ggtitle("Relación entre CE y PO, con su recta de regresión y color de acuerdo al tamaño del resto en ca
```



heterosc-1.bb

Prueba de Heteroscedasticidad

Podemos emplear la función `bptest` del paquete `lmtest`. Esta pide un modelo con ordenada al origen a pesar de que deseamos uno sin ella. Consideré que no es un problema dada la pequeña fracción que representa el valor hallado, respecto al máximo que alcanza la abcisa PO.

```
modelo2  <- lm(CE~PO,data=piloto)

prueba  <- bptest(modelo2)
```

Como se ve, el valor p es bajísimo y no se puede asumir homoscedasticidad. Por lo tanto vamos a aplicar el modelo propuesto a la estratificación.

Parametrización del Modelo Heteroscedástico

En *Valliant* se propone el astuto método de realizar una regresión entre los logaritmos de los cuadrados de los restos y los logaritmos de los valores de la variable auxiliar, ademas de detallarse comandos adecuados para realizar esta tarea en forma automática.

$e_i{}^2 = \sigma^2 \cdot PO_i^{\gamma}$

$ln\left(e_i{}^2\right) = ln\left(\sigma^2\right) + \gamma \cdot ln\left(PO_i\right)$

Queda así sugerido un método para encontrar los parámetros necesarios: vamos a realizar una regresión entre los logaritmos de los cuadrados de los restos de acuerdo al modelo lineal planteado.

$ln\left(e_i{}^2\right) = \beta_1 + \beta_2 \cdot ln\left(PO_i\right) \Rightarrow \gamma \approx \beta_2 \wedge e^{\beta_1} \approx \sigma^2$

```
marcoPiloto %>% mutate(logX = log(PO), logECuad=log(resto^2) ) -> marcoPiloto

modeloGamma  <- lm(logECuad~logX,data=marcoPiloto)

parametrosHetero  <- modeloGamma$coefficients %>% unname()

gamma2  <- parametrosHetero[2]

sig22  <- parametrosHetero[1] %>% exp()

parametrosPractools  <- list(beta = betaObservado, gamma = gamma2, varianza = sig22)

po  <- marcoPiloto$PO

res  <- marcoPiloto$resto

gamma1  <- (gammaFit(X=po,x=po,y=res,tol=0.001,maxiter=40))[["g.hat"]]
```

```
## Convergence attained in  3 steps.
## g.hat = 1.89783379352
```

```
beta1  <- mean( marcoPiloto$CE / marcoPiloto$PO )

sig21  <- var( marcoPiloto$CE / marcoPiloto$PO )

parametrosStrata  <- list(beta = betaObservado, gamma = gamma1, varianza = sig21)

parametros  <- tibble(
  variable = names(parametrosStrata),
  Strata = parametrosStrata,
  PracTools = parametrosPractools
  )
```

En las líneas precedentes probamos hallar gamma con la función específica para tal fin y por los métodos sugeridos en la ayuda del paquete strata (función strataLH), estos son las constantes que terminan en 1. Buscamos además las mismas constantes con el método explicado en *Valliant*. El resultado es apreciablemente dinstito para algunas de ellas.

g Recalcular los tamaños de muestra considerando el modelo hallado.

Esta mejora está incorporada al paquete, por lo que sólo hay que repetir el código anterior añadiendo la opción con los parámetros ya estimados.

```
controlModelo  <- list(beta = betaObservado, gamma = gamma1, sig2 = sig21)

modeloGeoNey  <- function(cv,l) {
  strata.geo(x=marcoAuxiliar$PO,
             CV=cv,Ls = l,
             alloc = allocNey,
             rh = 0.85,
             model="linear",
             model.control = controlModelo)
}
```

```r
mGeoNey3  <- map_df(listaCVs,~(modeloGeoNey(.x,3))[["nh"]] ) %>% mutate(estrato=tresEstratos)

mGeoNey4  <- map_df(listaCVs,~(modeloGeoNey(.x,4))[["nh"]] ) %>% mutate(estrato=cuatroEstratos)

mGeoNey5  <- map_df(listaCVs,~(modeloGeoNey(.x,5))[["nh"]] ) %>% mutate(estrato=cincoEstratos )

geometricoNeymanModelo  <- list(mGeoNey3,mGeoNey4,mGeoNey5)

modeloProp  <- function(cv,l) {
  strata.geo(x=marcoAuxiliar$PO,
             CV=cv,
             Ls = l,
             alloc = allocProp,
             model = "linear",
             model.control = controlModelo)
}

mGeoProp3  <- map_df(listaCVs,~(modeloProp(.x,3))[["nh"]] ) %>% mutate(estrato=tresEstratos)

mGeoProp4  <- map_df(listaCVs,~(modeloProp(.x,4))[["nh"]] ) %>% mutate(estrato=cuatroEstratos)

mGeoProp5  <- map_df(listaCVs,~(modeloProp(.x,5))[["nh"]] ) %>% mutate(estrato=cincoEstratos)

geometricoProporcionalModelo  <- list(mGeoProp3,mGeoProp4,mGeoProp5)

modeloLHNey <- function(cv,l,respuesta) {
  strata.LH(
            x=marcoAuxiliar$PO,
            CV=cv,
            Ls = l,
            alloc = allocNey,
            rh = respuesta,
            algo="Kozak",
            takeall = 1,
            model = "linear",
            model.control = controlModelo
                            )
}

mKozNey3  <- map_df(listaCVs,~(modeloLHNey(.x,3,c(0.85,0.85,1)))[["nh"]] ) %>% mutate(estrato=tresEstra

mKozNey4  <- map_df(listaCVs,~(modeloLHNey(.x,4,c(0.85,0.85,0.85,1)))[["nh"]] ) %>% mutate(estrato=cuat

mKozNey5  <- map_df(listaCVs,~(modeloLHNey(.x,5,c(0.85,0.85,0.85,0.85,1)))[["nh"]] ) %>% mutate(estrato


kozakNeymanModelo  <- list(mKozNey3,mKozNey4,mKozNey5)

modeloLHProp <- function(cv,l,respuesta) {
  strata.LH(
    x=marcoAuxiliar$PO,
```

Table 12: Asignación de Neyman con optimización geometrica y un modelo heteroscedástico.

5%	3%	1%	estrato
26	63	394	P
76	187	1175	M
16	36	36	G
118	286	1605	Total

5%	3%	1%	estrato
8	19	130	1
66	161	1155	2
13	32	226	3
15	36	36	4
102	248	1547	Total

5%	3%	1%	estrato
3	6	39	1
33	82	573	2
33	81	567	3
5	13	30	4
9	21	22	5
83	203	1231	Total

```
    CV=cv,
    Ls = l,
    alloc = allocProp,
    rh = respuesta,
    algo="Kozak",
    takeall = 1,
    model = "linear",
    model.control = controlModelo
    )
}

mKozProp3  <- map_df(listaCVs,~(modeloLHProp(.x,3,c(0.85,0.85,1)))[["nh"]] ) %>% mutate(estrato=tresEsti

mKozProp4  <- map_df(listaCVs,~(modeloLHProp(.x,4,c(0.85,0.85,0.85,1)))[["nh"]] ) %>% mutate(estrato=cua

mKozProp5  <- map_df(listaCVs,~(modeloLHProp(.x,5,c(0.85,0.85,0.85,0.85,1)))[["nh"]] ) %>% mutate(estrat


kozakPropModelo  <- list(mKozProp3,mKozProp4,mKozProp5)

geometricoNeymanModelo <- map(geometricoNeymanModelo,agregarTotales)

geometricoProporcionalModelo <- map(geometricoProporcionalModelo,agregarTotales)

kozakNeymanModelo <- map(kozakNeymanModelo,agregarTotales)

kozakPropModelo  <- map(kozakPropModelo,agregarTotales)
```

Imprimimos algunas tablas, como en el caso anterior:

```
kable(geometricoNeymanModelo, caption = "Asignación de Neyman con optimización geometrica y un modelo he

kable(geometricoProporcionalModelo, caption = "Asignación de Proporcional con optimización geometrica y

kable(kozakNeymanModelo, caption = "Asignación de Neyman con optimización de Kozak y un modelo heterosce

kable(kozakPropModelo, caption = "Asignación de Neyman con optimización de Kozak y un modelo heterosceda
```

Como era de esperar, los tamaños de muestra para los mismos CV con las mismas asignaciones se incrementan con el fin de hacer cargo a la mayor varianza esperada bajo el nuevo panorama.

Table 13: Asignación de Proporcional con optimización geometrica y un modelo heteroscedástico.

5%	3%	1%	estrato
183	446	1587	P
127	309	1100	M
3	7	25	G
313	762	2712	Total

5%	3%	1%	estrato
80	197	739	1
183	451	1690	2
18	44	164	3
3	7	24	4
284	699	2617	Total

5%	3%	1%	estrato
29	72	294	1
152	381	1571	2
52	130	533	3
2	5	19	4
2	4	14	5
237	592	2431	Total

Table 14: Asignación de Neyman con optimización de Kozak y un modelo heteroscedástico.

5%	3%	1%	estrato
40	95	240	P
37	89	276	M
36	50	485	G
113	234	1001	Total

5%	3%	1%	estrato
16	41	186	1
17	46	211	2
18	48	189	3
36	36	222	4
87	171	808	Total

5%	3%	1%	estrato
17	31	111	1
19	31	130	2
20	27	138	3
6	18	145	4
4	36	201	5
66	143	725	Total

Table 15: Asignación de Neyman con optimización de Kozak y un modelo heteroscedástico.

5%	3%	1%	estrato
75	184	361	P
15	38	154	M
46	57	541	G
136	279	1056	Total

5%	3%	1%	estrato
53	123	374	1
14	34	184	2
4	9	76	3
36	50	285	4
107	216	919	Total

5%	3%	1%	estrato
32	72	286	1
15	37	181	2
7	16	79	3
3	6	50	4
36	50	248	5
93	181	844	Total

h Elegir la opción más conveniente en términos de precisión en base a los puntos anteriores para generar una muestra de 200 elementos, incorporar los estratos al marco y presentarlo en un objeto. Presentar otro objeto con la muestra seleccionada.

Si bien es sencillo observar cuál es la asignación que ofrece la mejor relación entre precisión y tamaño de muestra, lo complejo en este caso es interpretar qué asignaciones son optimistas y cuáles son pesimistas y en qué grado lo son.

En principio, la asignación empleando Neyman sobre una variable auxiliar es una asignación optimista bajo el supuesto de que el comportamiento de esta y de la variable en estudio son similares. Al tomar en cuenta el modelo heteroscedástico, estamos logrando una estimación más realista, nos hacemos cargo de la diferencia en las varianzas y por tanto parece aceptable usar esta asignación. Además, efectivamente nos deja sobre un valor de n intermedio entre los propuestos por la asignación proporcional y la óptima de Neyman en el punto anterior. Cabe destacar que observando de todas formas las asignaciones proporcional y de Neyman, más allá de lo comentado, la diferencia no es de ninguna forma drástica. En todo caso cabría preguntarse cuán diferente puede ser el comportamiento de otras variables, para ver si el rendimiento mejorado que esperamos respecto a *CE* es suficiente para nuestros fines o si deberíamos incrementar algo más la muestra en atención a los avatares de otras variables.

En todos los casos el modelo de mejor rendimiento a la hora de reducir la muestra para cada nivel de precisión fue el de 5 estratos, así que es el que vamos a emplear.

- La asignación de muestras a emplear para la estificación será la de Neyman sobre una variable auxiliar a 5 estratos, ajustada para un modelo heteroscedástico. El estrato que representa los valores más extremos a partir de un corte determinado por el algoritmo es autorepresentado.

```
asignacionEmpleada <- function(n,l,respuesta) {
  strata.LH(
    x=marcoAuxiliar$PO,
    n=200,
    Ls = l,
    alloc = allocNey,
    rh = respuesta,
    algo="Kozak",
    takeall = 1,
    model = "linear",
    model.control = controlModelo
  )
}

asignacion  <- asignacionEmpleada(200,5,c(0.85,0.85,0.85,0.85,1))

asignacion
```

```
## Given arguments:
## x = marcoAuxiliar$PO
## n = 200, Ls = 5, takenone = 0, takeall = 1
## allocation: q1 = 0.5, q2 = 0, q3 = 0.5
## model = linear: beta = 2.65733210892, sig2 = 0.720178531839, gamma = 1.89783379352
## algo = Kozak: minsol = 1000, idopti = nh, minNh = 2, maxiter = 10000,
##               maxstep = 17, maxstill = 170, rep = 5, trymany = TRUE
##
## Strata information:
##           |      type   rh |     bh    E(Y)    Var(Y)   Nh  nh   fh
## stratum 1 | take-some 0.85 |   23.5   43.45    271.55 1905  44 0.02
```

```
## stratum 2 | take-some 0.85 |   45.5   85.56    801.71 1254  50 0.04
## stratum 3 | take-some 0.85 |   85.5  165.20   2711.28  593  43 0.07
## stratum 4 | take-some 0.85 |  289.5  302.84   9941.00  196  27 0.14
## stratum 5 |  take-all 1.00 | 1501.0 1603.63 625985.63   36  36 1.00
## Total                                                  3984 200 0.05
##
## Total sample size: 200
## Anticipated population mean: 101.68430723
## Anticipated CV: 0.0238892024887
## Note: CV=RRMSE (Relative Root Mean Squared Error) because takenone=0.
```

```
ext <- asignacion$bh

asignarEstrato  <- function(x) {
  if (x < ext[1] ) { return(1) }
  else if (x < ext[2] ) { return(2) }
  else if (x < ext[3] ) { return(3) }
  else if (x < ext[4] ) { return(4) }
  else { return(5) }
}

marcoAuxiliar %>% mutate( estrato = ( map_dbl(PO,asignarEstrato) %>% as.factor() ) ) %>% arrange(PO)-> m

muestra  <- strata(
  marcoEstratificado,
  stratanames = "estrato",
  size = asignacion$nh,
  method = "srswor",
  description = T
)
```

```
## Stratum 1
##
## Population total and number of selected units: 1905 44
## Stratum 2
##
## Population total and number of selected units: 1254 50
## Stratum 3
##
## Population total and number of selected units: 593 43
## Stratum 4
##
## Population total and number of selected units: 196 27
## Stratum 5
##
## Population total and number of selected units: 36 36
## Number of strata  5
## Total number of selected units 200
```

```
save(marcoEstratificado, file="MarcoEstratificado.Rdata")

save(muestra, file="MuestraSeleccionada.Rdata")
```