Técnicas de muestreo y remuestreo

En el curso, presentamos el concepto de técnicas de importación y exploración de datos. Ahora estás 'equipado' para cargar datos de diferentes fuentes y almacenarlos en un formato apropiado. En este documento, analizaremos importantes metodologías de muestreo de datos y su importancia en los algoritmos de aprendizaje automático.

El *muestreo* es un bloque importante en nuestro flujo de proceso de aprendizaje automático y cumple el doble propósito de ahorrar costos en la recopilación de datos y reducir el costo computacional sin comprometer la potencia del modelo de aprendizaje automático.

"*Una respuesta aproximada al problema correcto vale mucho más que una respuesta exacta a un problema aproximado*".

—John Tukey

La declaración de John Tukey encaja bien con el espíritu del muestreo. A medida que el avance tecnológico trajo consigo grandes capacidades de almacenamiento de datos, el costo incremental de aplicar técnicas de aprendizaje automático es enorme. El muestreo nos ayuda a equilibrar el costo de procesar grandes volúmenes de datos con una mejora marginal en los resultados.

Contrariamente a la creencia general de que el muestreo es útil solo para reducir un gran volumen de datos a un volumen manejable, el muestreo también es importante para mejorar las estadísticas obtenidas de muestras pequeñas.

En general, el aprendizaje automático trabaja con grandes volúmenes de datos, pero conceptos como el muestreo bootstrap también pueden ayudarte a obtener información de situaciones de muestras pequeñas.

Los objetivos de aprendizaje de este documento son los siguientes:

- Introducción al muestreo
- Terminología del muestreo
- Muestreo no probabilístico y muestreo probabilístico
- Implicaciones comerciales del muestreo
- Teoría estadística sobre las estadísticas de muestra
- Introducción al remuestreo
- Método de Monte Carlo: Muestreo de Aceptación-Rechazo
- Ilustración que ahorra tiempo computacional

Se ilustrarán diferentes técnicas de muestreo utilizando los datos de fraude con tarjetas de crédito, dataset adjunto al documento.

Introducción al muestreo

El muestreo es un proceso que selecciona unidades de una población de interés, de tal manera que la muestra pueda generalizarse a la población con confianza estadística. Por ejemplo, si un minorista en línea quisiera conocer el importe promedio del ticket de una compra en línea durante los últimos 12 meses, podría no querer promediar el importe del ticket a lo largo de la población (lo que puede representar millones de puntos de datos para grandes minoristas), pero sí podemos tomar una muestra representativa de compras de los últimos 12 meses y estimar el promedio de la muestra.

El promedio de la muestra puede entonces generalizarse a la población con cierta confianza estadística. La confianza estadística variará según la técnica de muestreo utilizada y el tamaño.

En general, las técnicas de muestreo se aplican a dos escenarios: para crear un conjunto de datos manejable para el modelado y para resumir las estadísticas de la población. Esta amplia categorización puede presentarse como objetivos del muestreo:

- Muestreo de modelos (*Model sampling*)
- Muestreo de encuestas (*Survey sampling*)

El *muestreo de modelos* se realiza cuando los datos poblacionales ya están recopilados y se desea reducir el tiempo y el costo computacional del análisis, además de mejorar la inferencia de los modelos. Otro enfoque consiste en crear un diseño muestral y luego encuestar únicamente a la población para recolectar la muestra y así ahorrar costes de recolección de datos.

La figura 1 muestra los dos objetivos empresariales del muestreo. El diseño y la evaluación de la encuesta de muestreo quedan fuera del alcance de este curso, por lo que nos centraremos únicamente en el muestreo de modelos.

Esta clasificación también es útil para identificar los objetivos finales del muestreo.
Esto facilita la elección de la metodología y la técnica exploratoria adecuadas para el muestreo. En el contexto del flujo de construcción de modelos de aprendizaje automático, nos centraremos en el muestreo de modelos. Se asume que los datos ya se han recopilado y

nuestro objetivo final es extraer información de ellos, en lugar de desarrollar una encuesta sistemática para recopilarla.

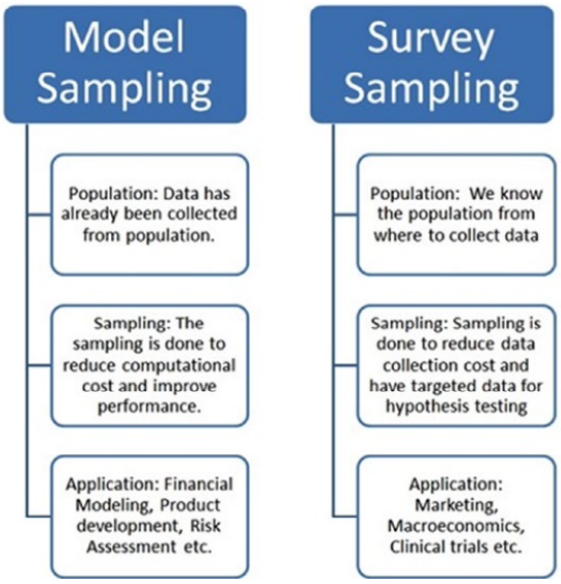

Figura 1: Objetivos del muestreo.

Terminología de muestreo

Antes de profundizar en el muestreo, definamos la terminología básica que utilizaremos a lo largo del documento. Los conceptos de estadística y probabilidad de tus cursos, resultarán útiles para comprender la terminología del muestreo. Esta sección describe la definición y la formulación matemática del muestreo.

Muestra

Una *muestra* es un conjunto de unidades o individuos seleccionados de una población original para proporcionar información útil sobre la población. Esta información puede ser la forma general de la distribución, estadísticas básicas, propiedades de los parámetros de distribución poblacional o información sobre momentos superiores. Además, la muestra puede utilizarse para estimar estadísticos de prueba para la comprobación de hipótesis. Una muestra representativa puede utilizarse para estimar propiedades de la población o para modelar parámetros poblacionales.

Por ejemplo, la Organización Nacional de Encuestas por Muestreo (NNSO, *National Sample Survey Organization*) recopila datos muestrales sobre desempleo contactando a un número limitado de hogares, y luego esta muestra se utiliza para proporcionar datos sobre el desempleo nacional.

Distribución de muestreo

La distribución de las medias de un tamaño particular de muestras se denomina *distribución muestral* de medias; de igual manera, la distribución de las varianzas muestrales correspondientes se denomina distribución muestral de las varianzas. Estas distribuciones son fundamentales para realizar cualquier tipo de prueba de hipótesis.

Media y varianza poblacionales

La *media poblacional* es el promedio aritmético de los datos poblacionales. Todos los puntos de datos contribuyen a la media poblacional con el mismo peso. De igual manera, la varianza poblacional es la varianza calculada utilizando todos los puntos de datos.

$$\text{Media poblacional: } \mu = \frac{\sum_{i=1}^{n} X_i}{n}$$

$$\text{Varianza poblacional: } \sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

Media y varianza muestrales

Cualquier subconjunto que se extraiga de la población constituye una muestra. La media y la varianza obtenidas de esa muestra se denominan *estadísticas muestrales*. El concepto de grados de libertad se utiliza cuando se utiliza una muestra para estimar parámetros de distribución; por lo tanto, para la varianza muestral, se observará que el denominador es diferente de la varianza poblacional.

$$\text{Media muestral: } \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\text{Varianza muestral: } s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

Media y varianza agrupadas

Para una muestra *k* de tamaño n_1, n_2, n_3, ..., n_k, tomada de la misma población, la media y la varianza poblacionales estimadas se definen como sigue.

$$\text{Media poblacional estimada: } \bar{x}_p = \frac{\sum_{i=1}^{n} \bar{x}_i}{\sum_{i=1}^{n}(n_i)} = \frac{(n_1)\bar{x}_1 + (n_2)\bar{x}_2 + \cdots + (n_k)\bar{x}_k}{n_1 + n_2 + \cdots + n_k}$$

$$\text{Varianza poblacional estimada: } s_p^2 = \frac{\sum_{i=1}^{n}(n_i - 1)s_i^2}{\sum_{i=1}^{n}(n_i - 1)} = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2 + \cdots + (n_k - 1)s_k^2}{n_1 + n_2 + \cdots + n_k - k}$$

En situaciones reales, normalmente podemos obtener múltiples muestras de la misma población en diferentes puntos del espacio/ubicación y del tiempo. Por ejemplo, supongamos que tenemos que estimar los ingresos promedio del dueño de una librería en una ciudad. Obtendremos muestras de los ingresos de los dueños de librerías de diferentes partes de la ciudad en diferentes momentos. Posteriormente, podemos combinar la media y la varianza individuales de diferentes muestras para obtener una estimación de la población utilizando la media y la varianza agrupadas.

Punto muestral

Un posible resultado en un experimento de muestreo se denomina *punto muestral*. En muchos tipos de muestreo, no todos los puntos de datos de la población son puntos muestrales.

Los puntos muestrales son importantes cuando el diseño muestral se vuelve complejo. El investigador podría querer excluir algunas observaciones del muestreo; alternativamente, el propio proceso de muestreo, por diseño, puede reducir la probabilidad de selección del dato no deseado.

Por ejemplo, supongamos que tienes datos de género con tres valores posibles: Masculino, Femenino y Desconocido. Podrías querer descartar todos los Desconocidos como un error; esto implica excluir la observación del muestreo. De lo contrario, si los datos son grandes y los Desconocidos representan una proporción muy pequeña, la probabilidad de muestrearlos es insignificante. En ambos casos, Desconocido no es un punto muestral.

Error de muestreo

La diferencia entre el valor real de una estadística poblacional y la estadística muestral es el *error de muestreo*. Este error se atribuye al hecho de que la estimación se obtuvo de la muestra.

Por ejemplo, supongamos que sabes, mediante datos del censo, que el ingreso mensual promedio de los residentes de Boston es de $3,000 (la media poblacional). Por lo tanto,

podemos decir que la media real de los ingresos es de $3,000. Supongamos que una empresa de investigación de mercado realizó una pequeña encuesta a residentes de Boston. Encontramos que el ingreso promedio de la muestra de esta pequeña encuesta es de $3,500. El error de muestreo es, por lo tanto, de $3,500 a $3,000, lo que equivale a $500. Nuestras estimaciones de la muestra sobreestiman el ingreso promedio, lo que también indica que la muestra no representa fielmente a la población.

Fracción de muestreo

La *fracción muestral* es la razón entre el tamaño de la muestra y el tamaño de la población, $f = \frac{n}{N}$.

Por ejemplo, si el tamaño total de la población es de 500,000 habitantes y se desea extraer una muestra de 2,000, la fracción muestral sería f = 2,000/50,000 = 0.04. En otras palabras, se muestrea el 4% de la población.

Sesgo de muestreo

El *sesgo muestral* se produce cuando las unidades muestrales de la población no son características de la misma (es decir, no reflejan) la población. El sesgo muestral hace que una muestra no sea representativa de la población.

Volviendo al ejemplo del error muestral, descubrimos que el ingreso promedio de la muestra es mucho mayor que el ingreso promedio del censo (promedio real). Esto significa que nuestro diseño muestral ha estado sesgado hacia los residentes de Boston con mayores ingresos. En ese caso, nuestra muestra no es una representación fiel de los residentes de Boston.

Muestreo sin reemplazo (SWOR, *Sampling without replacement*)

El muestreo con reemplazo se diferencia del SWOR en el hecho de que una unidad puede ser muestreada más de una vez en la misma muestra. El muestreo sin reemplazo requiere que se cumplan dos condiciones:

- Cada unidad/punto muestral tiene una probabilidad finita de selección distinta de cero.
- Una vez seleccionada una unidad, se elimina de la población.

En otras palabras, todas las unidades tienen una probabilidad finita de ser muestreadas estrictamente una sola vez.

Por ejemplo, si tenemos una bolsa con 10 bolas, marcadas con los números del 1 al 10, cada bola tiene una probabilidad de selección de 1/10 en un muestreo aleatorio sin reemplazo.

Supongamos que tenemos que elegir tres bolas de la bolsa; luego, después de cada selección, la probabilidad de selección aumenta a medida que disminuye el número de bolas restantes en la bolsa. Por lo tanto, para la primera bola, la probabilidad de ser seleccionada es de 1/10, para la segunda es de 1/9 y para la tercera es de 1/8.

Muestreo con reemplazo (SWR, *Sampling with replacement*)

El muestreo con reemplazo se diferencia del SWOR en que una unidad puede muestrearse más de una vez en la misma muestra. El muestreo con reemplazo requiere que se cumplan dos condiciones:

- Cada unidad/punto de muestra tiene una probabilidad finita de selección distinta de cero.
- Una unidad puede seleccionarse varias veces, ya que la población de muestreo siempre es la misma.

En el muestreo sin reemplazo, la unidad puede muestrearse más de una vez y cada vez tiene la misma probabilidad de ser muestreada. Este tipo de muestreo prácticamente amplía el tamaño de la población hasta el infinito, ya que se pueden crear tantas muestras de cualquier tamaño con este método. Volviendo a nuestro ejemplo anterior en SWOR, si tenemos que elegir tres bolas con SWR, cada bola tendrá exactamente la misma probabilidad finita de 1/10 para el muestreo.

Es importante tener en cuenta que el muestreo con reemplazo técnicamente hace que el tamaño de la población sea infinito. Ten cuidado al elegir el SWR, ya que, en la mayoría de los casos, cada observación es única y contarla varias veces crea sesgo en los datos. En esencia, esto significa que se permite la repetición de la observación. Por ejemplo, 100 personas con el mismo nombre, ingresos, edad y género en la muestra crearán sesgo en el conjunto de datos.

Fraude con Tarjetas de Crédito: Estadísticas Poblacionales

El conjunto de datos de fraude con tarjetas de crédito es un buen ejemplo de cómo crear un plan de muestreo para algoritmos de aprendizaje automático. El conjunto de datos es enorme, con 10 millones de filas y múltiples características. Esta sección te mostrará cómo calcular e

interpretar la medida de muestreo clave de la población para este conjunto de datos. Se mostrarán las siguientes medidas estadísticas:

- Media poblacional
- Varianza poblacional
- Media y varianza agrupadas

Para explicar estas medidas, elegimos la característica de saldo pendiente (*outstanding balance*) como la cantidad de interés.

Descripción de los datos

Un resumen para describir las variables en los datos de fraude con tarjetas de crédito:

- custID: Un identificador único para cada cliente
- gender: Sexo del cliente
- state: Estado de residencia del cliente en Estados Unidos
- cardholder: Número de tarjetas de crédito que posee el cliente
- balance: Saldo de la tarjeta de crédito
- numTrans: Número de transacciones hasta la fecha
- numIntlTrans: Número de transacciones internacionales hasta la fecha
- creditLine: Corporación de servicios financieros, como Visa, MasterCard o American Express
- scamRisk: Variable binaria; 1 significa que el cliente está siendo estafado; 0 significa lo contrario

```
> library(data.table)
> data <- fread("ccFraud.csv",header=T, verbose = FALSE, showProgress = FALSE)
> US_state <- fread("US_State_Code_Mapping.csv",header=T, showProgress = FALSE )
> data<-merge(data, US_state, by = 'state')
> Gender_map<-fread("Gender Map.csv",header=T)
> data<-merge(data, Gender_map, by = 'gender')
> Credit_line<-fread("credit line map.csv",header=T)
> data<-merge(data, Credit_line, by = 'creditLine')
> setnames(data,"custID","CustomerID")
> setnames(data,"code","Gender")
> setnames(data,"numTrans","DomesTransc")
> setnames(data,"numIntlTrans","IntTransc")
> setnames(data,"fraudRisk","FraudFlag")
> setnames(data,"cardholder","NumOfCards")
> setnames(data,"balance","OutsBal")
```

```
> setnames(data,"StateName","State")
> str(data)
```

```
Classes 'data.table' and 'data.frame':10000000 obs. of  14 variables:
 $ creditLine : int  1 1 1 1 1 1 1 1 1 1 ...
 $ gender     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ state      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ CustomerID : int  4446 59161 136032 223734 240467 248899 262655 324670 390138 482698 ...
 $ NumOfCards : int  1 1 1 1 1 1 1 1 1 1 ...
 $ OutsBal    : int  2000 0 2000 2000 2000 0 0 689 2000 0 ...
 $ DomesTransc: int  31 25 78 11 40 47 15 17 48 25 ...
 $ IntTransc  : int  9 0 3 0 0 0 0 9 0 35 ...
 $ FraudFlag  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ State      : chr  "Alabama" "Alabama" "Alabama" "Alabama" ...
 $ PostalCode : chr  "AL" "AL" "AL" "AL" ...
 $ Gender     : chr  "Male" "Male" "Male" "Male" ...
 $ CardType   : chr  "American Express" "American Express" "American Express" "American Express" ...
 $ CardName   : chr  "SimplyCash\256 Business Card from American Express" "SimplyCash\256 Business Card from American Express"
"SimplyCash\256 Business Card from American Express" "SimplyCash\256 Business Card from American Express" ...
 - attr(*, ".internal.selfref")=<externalptr>
 - attr(*, "sorted")= chr "creditLine"
```

Como se mencionó anteriormente, elegimos el saldo pendiente (outstanding balance) como la variable/característica de interés. En la salida de str() para la descripción de los datos, podemos ver que el saldo pendiente se almacena en una variable llamada OutsBal, que es de tipo entero. Al ser una variable continua, se pueden definir la media y la varianza para esta variable.

```
> data$creditLine <- NULL
> data$gender <- NULL
> data$state <- NULL
> data$PostalCode <- NULL
> # Ejecuta el siguiente código si deseas almacenar los datos transformados.
> write.csv(data,"Credit Card Fraud Dataset.csv",row.names = FALSE)
> # Describe los datos
> str(data)
```

Media Poblacional

La media es un estadístico más tradicional para medir la tendencia central de cualquier distribución de datos. El saldo pendiente medio de nuestros clientes en el conjunto de datos de fraude con tarjetas de crédito resulta ser de $4109.92. Esta es nuestra primera comprensión de la población. La media poblacional nos indica que, en promedio, los clientes tienen un saldo pendiente de $4109.92 en sus tarjetas.

```
> Population_Mean_P <-mean(data$OutsBal)
> cat("The average outstanding balance on cards is ",Population_Mean_P)
The average outstanding balance on cards is  4109.92
```

Varianza Poblacional

La varianza es una medida de dispersión para el conjunto de números dado. Cuanto menor sea la varianza, más cerca estarán los números de la media, y cuanto mayor sea la varianza, más lejos estarán los números de la media. Para el saldo pendiente, la varianza es de 15974788 y la desviación estándar es de 3996.8. La varianza por sí sola no es comparable entre diferentes poblaciones o muestras. Es necesario analizar la varianza junto con la media de la distribución. La desviación estándar es otra medida y equivale a la raíz cuadrada de la varianza.

```
> Population_Variance_P <-var(data$OutsBal)
> cat("The variance in the average outstanding balance is ",Population_Variance_P)
The variance in the average outstanding balance is  15974788
> cat("Standard deviation of outstanding balance is", sqrt(Population_Variance_P))
Standard deviation of outstanding balance is 3996.847
```

Media y varianza agrupadas

La media y la varianza agrupadas estiman la media y la varianza de la población cuando se extraen múltiples muestras de forma independiente. Para ilustrar la media y la varianza agrupadas en comparación con la media y la varianza reales de la población, primero crearemos cinco muestras aleatorias de 10,000, 20,000, 40,000, 80,000 y 100,000, y calcularemos su media y varianza.

Con estas muestras, estimaremos la media y la varianza de la población mediante la fórmula de media y varianza agrupadas. Los valores agrupados son útiles porque las estimaciones de una sola muestra pueden producir un gran error de muestreo, mientras que si extraemos muchas muestras de la misma población, el error de muestreo se reduce. La estimación colectiva se aproximará más a las estadísticas reales de la población.

Nota. Dado que la fracción de muestreo es baja para los distintos tamaños de muestra (para un tamaño de muestra de 100,000, f = 100000/10000000 = 1/100), es demasiado grande. La varianza no se verá afectada por la corrección de 1 grado de libertad, por lo que podemos usar con seguridad la función var() en R para la varianza de la muestra.

En el siguiente fragmento de código de R, creamos cinco muestras aleatorias utilizando la función sample(). sample() es una función integrada que se ha utilizado varias veces en los cursos de minería de datos. Cabe destacar que la función sample() funciona con valores de semilla aleatorios, por lo que si deseas crear código reproducible, utiliza la función set.seed(937) en R. Esto garantizará que cada vez que ejecutes el código, obtengas la misma muestra aleatoria.

```
> library(knitr)
> set.seed(937)
```

```
> i<-1
> n<-rbind(10000,20000,40000,80000,100000)
> Sampling_Fraction<-n/nrow(data)
> sample_mean<-numeric()
> sample_variance<-numeric()
> for(i in 1:5)
+ {
+   sample_100K <-data[sample(nrow(data),size=n[i], replace =FALSE, prob =NULL),]
+   sample_mean[i]<-round(mean(sample_100K$OutsBal),2)
+   sample_variance[i] <-round(var(sample_100K$OutsBal),2)
+ }
> Sample_statistics <-cbind (1:5,c('10K','20K','40K','80K','100K'),
+   sample_mean,sample_variance,round(sqrt(sample_variance),2),Sampling_Fraction)
> knitr::kable(Sample_statistics, col.names =c("S.No.", "Size","Sample_ Mean",
+ "Sample_Variance","Sample SD","Sample_Fraction"))
```

En la tabla 1, se presentan las propiedades básicas de las cinco muestras. La fracción muestral más alta corresponde al mayor tamaño de muestra. Cabe destacar que, a medida que aumenta el tamaño de la muestra, la varianza muestral disminuye.

Tabla 1: Estadísticas de la muestra.

S.No.	Size	Sample_Mean	Sample_Variance	Sample SD	Sample_Fraction
1	10K	4112.72	16325933.79	4040.54	0.001
2	20K	4087.86	15656578.13	3956.84	0.002
3	40K	4065.04	15864519.47	3983.03	0.004
4	80K	4099.68	15975735.51	3996.97	0.008
5	100K	4118.23	16053464.19	4006.68	0.01

Ahora, utilicemos la fórmula de media y varianza agrupadas para calcular la media poblacional de las cinco muestras que extrajimos de la población y luego compararlas con la media y la varianza poblacionales.

```
> i<-1
> Population_mean_Num<-0
> Population_mean_Den<-0
> for(i in 1:5)
+ {
+   Population_mean_Num =Population_mean_Num +sample_mean[i]*n[i]
+   Population_mean_Den =Population_mean_Den +n[i]
+ }
> Population_Mean_S<-Population_mean_Num/Population_mean_Den
> cat("The pooled mean ( estimate of population mean) is",Population_Mean_S)
```

```
The pooled mean ( estimate of population mean) is 4101.134
```

La media agrupada es de $4,101.134. Ahora aplicamos este mismo proceso para calcular la varianza agrupada de las muestras. Además, mostraremos la desviación estándar como una columna adicional para que la dispersión sea comparable con la media.

```
> i<-1
> Population_variance_Num<-0
> Population_variance_Den<-0
> for(i in 1:5)
+ {
+   Population_variance_Num =Population_variance_Num +(sample_variance[i])*(n[i] -1)
+   Population_variance_Den =Population_variance_Den +n[i] -1
+ }
> Population_Variance_S<-Population_variance_Num/Population_variance_Den
> Population_SD_S<-sqrt(Population_Variance_S)
> cat("The pooled variance (estimate of population variance) is", Population_Variance_S)
The pooled variance (estimate of population variance) is 15977508
> cat("The pooled standard deviation (estimate of population standard deviation) is",
sqrt(Population_Variance_S))
The pooled standard deviation (estimate of population standard deviation) is 3997.187
```

La desviación estándar agrupada es de $3,997.187. Ahora tenemos estadísticas agrupadas y estadísticas poblacionales. Aquí, creamos una comparación entre ambas y observamos la precisión con la que las estadísticas agrupadas estimaron las estadísticas poblacionales:

```
> SamplingError_percent_mean<-round((Population_Mean_P -sample_mean)/
Population_Mean_P,3)
> SamplingError_percent_variance<-round((Population_Variance_P -sample_variance)/
Population_Variance_P,3)
> Com_Table_1<-cbind(1:5,c('10K','20K','40K','80K','100K'),Sampling_Fraction,
+  SamplingError_percent_mean, SamplingError_percent_variance)
> knitr::kable(Com_Table_1, col.names =c("S.No.","Size","Sampling_Frac",
+   "Sampling_Error_Mean(%)","Sampling_Error_Variance(%)"))
```

La tabla 2 muestra la comparación de la media poblacional y la varianza con cada muestra individual. Cuanto mayor sea la muestra, más se acercará la estimación de la media a la estimación poblacional real.

Crea la misma vista para las estadísticas agrupadas con las estadísticas poblacionales. La diferencia se expresa como un porcentaje de las diferencias entre las estadísticas agrupadas/de la muestra y las estadísticas poblacionales reales.

Tabla 2: Estadísticas de la muestra versus la población.

S.No.	Size	Sampling_Frac	Sampling_Error_Mean(%)	Sampling_Error_Variance(%)
1	10K	0.001	-0.001	-0.022
2	20K	0.002	0.005	0.02
3	40K	0.004	0.011	0.007
4	80K	0.008	0.002	0
5	100K	0.01	-0.002	-0.005

```
> SamplingError_percent_mean<-(Population_Mean_P -Population_Mean_S)/Population_Mean_P
> SamplingError_percent_variance<-(Population_Variance_P -Population_Variance_S)/
Population_Variance_P
> Com_Table_2 <-cbind(Population_Mean_P,Population_Mean_S,
+   SamplingError_percent_mean)
> Com_Table_3 <-cbind(Population_Variance_P,Population_Variance_S,
+   SamplingError_percent_variance)
> knitr::kable(Com_Table_2)
> knitr::kable(Com_Table_3)
```

Tabla 3: Media poblacional y diferencia de medias muestrales.

Population_Mean_P	Population_Mean_S	SamplingError_percent_mean
4109.92	4101.134	0.0021378

Tabla 4: Varianza poblacional y varianza muestral.

Population_Variance_P	Population_Variance_S	SamplingError_percent_variance
15974788	15977508	-0.0001702

La media agrupada se acerca a la media real de la población. Esto demuestra que, dadas varias muestras, es más probable que las estadísticas agrupadas capturen valores estadísticos reales. Ahora ha visto cómo una muestra tan pequeña en comparación con la población proporciona estimaciones muy cercanas a la estimación poblacional. ¿Te ofrece este ejemplo una herramienta para gestionar big data utilizando pequeñas muestras?

Probablemente ya hayas empezado a pensar en el análisis costo-beneficio del muestreo. Esto es muy relevante para los algoritmos de aprendizaje automático que procesan millones de puntos de datos. Un mayor número de puntos de datos no significa necesariamente que todos contengan patrones, tendencias o información significativa. El muestreo intentará evitar errores y te ayudará a centrarte en conjuntos de datos relevantes para el aprendizaje automático.

Implicaciones empresariales del muestreo

El muestreo se aplica en múltiples etapas del desarrollo de modelos y la toma de decisiones. Los métodos de muestreo y su interpretación se rigen por las limitaciones empresariales y los métodos estadísticos elegidos para las pruebas de inferencia. Los científicos de datos establecen un delicado equilibrio entre las implicaciones empresariales y la validez y relevancia de los resultados estadísticos. En la mayoría de los casos, el problema lo plantea la empresa y los científicos de datos deben trabajar de forma específica para resolverlo.

Por ejemplo, supongamos que la empresa quiere saber por qué los clientes no vuelven a su sitio web. Este problema determinará las condiciones del muestreo. Para saber por qué los clientes no regresan, ¿Realmente necesitas una muestra representativa de toda su población? ¿O simplemente tomarás una muestra de los clientes que no regresaron? ¿O preferirías estudiar solo una muestra de los clientes que regresan y anular los resultados? ¿Por qué no crear una combinación personalizada de todos los clientes que regresan y los que no? Como puedes observar, muchas de estas preguntas, junto con las limitaciones prácticas de tiempo, costo, capacidad computacional, etc., serán factores decisivos para la recopilación de datos para este problema.

En general, los escenarios que se enumeran a continuación son características destacadas del muestreo y algunas deficiencias que deben tenerse en cuenta al utilizarlo en el flujo de construcción de modelos de aprendizaje automático.

Características del muestreo

- Carácter científico
- Optimiza las limitaciones de tiempo y espacio
- Método fiable de prueba de hipótesis
- Permite un análisis profundo al reducir costos
- En casos donde la población es muy grande y la infraestructura es una limitación, el muestreo es la única solución

Desventajas del muestreo

- El sesgo de muestreo puede causar inferencias erróneas
- El muestreo representativo no siempre es posible debido al tamaño, el tipo, los requisitos, etc.
- No es una ciencia exacta, sino una aproximación dentro de ciertos límites de confianza

- En las encuestas por muestreo, existen problemas de errores manuales, respuestas inadecuadas, ausencia de informantes, etc.

Muestreo probabilístico y no probabilístico

La metodología de muestreo depende en gran medida de lo que se desee hacer con la muestra. Ya sea para generar una hipótesis sobre los parámetros poblacionales o para contrastarla, clasificamos los métodos de muestreo en dos categorías principales: muestreo probabilístico y muestreo no probabilístico. La comparación en la figura 2 muestra las principales diferencias entre ambos.

Muestreo Probabilístico (Aleatorio)	Muestreo no probabilístico (no aleatorio)
Puede generalizarse a la población definida por el frame muestral.	No se puede generalizar más allá de la muestra.
Puede aplicar métodos estadísticos, pruebas de hipótesis y límites de confianza.	Investigación exploratoria, facilita la generación de hipótesis e inferencia analítica.
Puede estimar estadísticas/parámetros poblacionales a partir de la muestra.	Las estadísticas/parámetros poblacionales no son de interés.
Reduce el sesgo al variar el diseño muestral.	Sesgado, se desconoce la idoneidad de la muestra.
Selección aleatoria de la población.	Sin población definida; es más económico, fácil y práctico de realizar.

Figura 2: Muestreo probabilístico versus no probabilístico.

En el muestreo probabilístico, los métodos de muestreo extraen cada unidad con una probabilidad finita. El frame muestral que asigna la unidad poblacional a la unidad muestral se crea con base en la distribución de probabilidad de la variable aleatoria utilizada para el muestreo.

Estos tipos de métodos se utilizan comúnmente para el muestreo de modelos y ofrecen una alta confiabilidad para extraer inferencias poblacionales. Eliminan el sesgo en la estimación de parámetros y pueden generalizarse a la población.

A diferencia del muestreo no probabilístico, es necesario conocer de antemano la población de la que se realizará el muestreo. Esto hace que este método sea costoso y, en ocasiones, difícil de implementar. El muestreo no probabilístico se basa en el juicio subjetivo de expertos y en las necesidades del negocio.

Este método es popular cuando las necesidades del negocio no deben alinearse con los requisitos estadísticos o resulta difícil crear un frame de muestreo probabilístico. El método

de muestreo no probabilístico no asigna probabilidad a las unidades poblacionales, por lo que resulta poco fiable extraer inferencias de la muestra.

El muestreo no probabilístico presenta sesgo hacia las clases seleccionadas, ya que la muestra no es representativa de la población. Los métodos no probabilísticos son más populares en la investigación exploratoria de nuevos rasgos de la población que puedan evaluarse posteriormente con mayor rigor estadístico.

A diferencia de las técnicas probabilísticas, no es posible estimar parámetros poblacionales con precisión utilizando técnicas no probabilísticas.

Tipos de muestreo no probabilístico

En esta sección, abordaremos brevemente los tres tipos principales de métodos de muestreo no probabilístico. Dado que estas técnicas son más adecuadas para muestras de encuestas, no las analizaremos en detalle.

Muestreo por conveniencia

En el muestreo por conveniencia, el experto seleccionará los datos fácilmente disponibles. Esta técnica es la más económica y requiere menos tiempo. En nuestro caso, supongamos que los datos de Nueva York son accesibles, pero los de otros estados no lo son, por lo que seleccionamos datos de un solo estado para estudiar todo Estados Unidos.

La muestra no sería representativa de la población y estaría sesgada. Los datos tampoco se pueden generalizar a toda la población. Sin embargo, la muestra podría permitirnos formular hipótesis que posteriormente se puedan comprobar utilizando muestras aleatorias de todos los estados.

Muestreo propósito

Cuando el muestreo se basa en el juicio subjetivo del experto, se denomina muestreo propósito. En este método, el experto muestreará las unidades que le ayuden a establecer la hipótesis que intenta comprobar. En nuestro caso, si el investigador solo está interesado en analizar las tarjetas American Express, simplemente seleccionará algunas unidades del conjunto de registros de ese tipo de tarjeta.

Además, existen muchos tipos de métodos de muestreo intencional (*proposito*), como el muestreo de máxima varianza, el muestreo de casos extremos, el muestreo homogéneo, etc., pero no se abordan en este curso, ya que carecen de la representatividad de la población, necesaria para los métodos de aprendizaje automático imparciales.

Muestreo por cuotas

Como su nombre indica, el muestreo por cuotas se basa en una cuota prefijada para cada tipo de caso, generalmente la cuota decidida por un experto. Fijar una cuadrícula de cuotas para el muestreo garantiza una representación equitativa o proporcional de los sujetos muestreados. Esta técnica es popular en el diseño de campañas de marketing, las pruebas A/B y las pruebas de nuevas funciones.

En este documento, abordamos los métodos de muestreo con ejemplos extraídos de nuestros datos sobre fraude con tarjetas de crédito. Te animamos a explorar más el muestreo no probabilístico en el contexto del problema empresarial que tienes ante sí. Hay ocasiones en las que la experiencia puede más que las estadísticas, por lo que el muestreo no probabilístico es igualmente importante en muchos casos de uso.

Teoría estadística sobre distribuciones de muestreo

Las técnicas de muestreo se basan en teoremas bien establecidos y métodos estadísticos de eficacia comprobada. Para estudiar la distribución muestral, es necesario comprender dos teoremas importantes de la estadística:

- Ley de los Grandes Números
- Teorema del Límite Central

Esta sección explica estos dos teoremas con algunas simulaciones.

Ley de los Grandes Números: LLN (Law of large numbers)

En general, a medida que aumenta el tamaño de la muestra en una prueba, esperamos que los resultados sean más precisos, con menores desviaciones en los resultados esperados. La ley de los grandes números formaliza esto con la ayuda de la teoría de la probabilidad. La primera referencia notable a este concepto la dio el matemático italiano Gerolamo Cardano en el siglo XVI, cuando observó y afirmó que la estadística empírica se acerca a su valor real a medida que aumenta el número de ensayos.

Posteriormente, se realizó una gran cantidad de trabajo para obtener una forma diferente de la Ley de los Grandes Números. El ejemplo que vamos a analizar para el lanzamiento de una moneda fue demostrado inicialmente por Bernoulli y posteriormente proporcionó pruebas de sus observaciones. Aleksander Khinchin formuló la afirmación más popular de la Ley de los

Grandes Números, también llamada ley débil de los grandes números. Esta ley débil también se denomina ley de los promedios.

Ley débil de los grandes números

En el espacio de probabilidad, el promedio muestral converge a un valor esperado a medida que el tamaño de la muestra tiende al infinito. En otras palabras, a medida que aumenta el número de ensayos o el tamaño de la muestra, aumenta la probabilidad de acercarse al promedio real. Esta ley débil también se denomina Ley de Khinchin en reconocimiento a su contribución.

La ley débil de los grandes números establece que los promedios muestrales convergen en probabilidad hacia el valor esperado, $X_n \to \mu$ cuando $n \to \infty$.

Alternativamente, para cualquier número positivo $\in$.

$$\lim_{n\to\infty} \Pr(|\bar{X}_n - \mu| > \epsilon) = 0$$

Ley fuerte de los grandes números

Es importante comprender la sutil diferencia entre la ley débil y la ley fuerte de los grandes números. La ley fuerte de los grandes números establece que la media muestral convergerá a la media real con una probabilidad de 1, mientras que la ley débil solo establece que convergerán. Por lo tanto, la ley fuerte es más eficaz al estimar la media poblacional a partir de las medias muestrales.

La ley fuerte de los grandes números establece que la media muestral *converge casi con seguridad* al valor esperado $\bar{X}_n \xrightarrow{\text{a.s.}} \mu$ cuando $n \to \infty$.

Equivalente a

$$\Pr\left(\lim_{n\to\infty} \bar{X}_n = \mu\right) = 1$$

Nota. Existen múltiples representaciones y demostraciones de la ley de los grandes números. Se recomienda consultar cualquier texto de probabilidad de nivel de posgrado para obtener más información.

Sin entrar en los detalles estadísticos de este teorema, presentaremos un ejemplo para comprenderlo. Consideremos un ejemplo de lanzamiento de moneda, donde el resultado sigue una distribución binomial.

Supongamos que tienes una moneda sesgada y tienes que determinar cuál es la probabilidad de obtener "cara" en cualquier lanzamiento. Según LLN, si realiza el experimento de lanzamiento de moneda varias veces, podrás calcular la probabilidad real de obtener cara.

Ten en cuenta que, para una moneda no sesgada, puedes usar la teoría clásica de la probabilidad y obtener la probabilidad P(cara) = Número total de resultados favorables / Número total de resultados = 1/2. Sin embargo, para una moneda no sesgada, la probabilidad asociada a cada evento es desigual y, por lo tanto, no puedes usar el enfoque clásico. Realizaremos un experimento de lanzamiento de moneda para determinar la probabilidad de obtener cara.

Pasos de la simulación con código R

Paso 1: Supongamos un valor del parámetro de distribución binomial, p = 0.60 (por ejemplo), que estimaremos utilizando la Ley de los Grandes Números.

```
> # Set parameters for a binomial distribution Binomial(n, p)
> # n -> no. of toss
> # p -> probability of getting a head
> n <-100
> p <-0.6
```

En el fragmento de código anterior, establecimos la media real para nuestro experimento. Es decir, sabemos que nuestra población proviene de una distribución binomial con p = 0.6. El experimento nos ayudará a estimar este valor a medida que aumenta el número de experimentos.

Paso 2: Muestrear un punto de la distribución binomial (p).

```
> #Create a data frame with 100 values selected samples from Binomial(1,p)
> set.seed(917);
> dt <-data.table(binomial=rbinom(n, 1, p) ,count_of_heads=0, mean=0)
> # Setting the first observation in the data frame
> ifelse(dt$binomial[1] ==1, dt[1, 2:3] <-1, 0)
[1] 0
```

Utilizamos la función integrada rbinom() para muestrear una variable aleatoria de distribución binomial con el parámetro p = 0.6. Este valor de probabilidad se elige de forma que la moneda esté sesgada. Si la moneda no está sesgada, sabemos que la probabilidad de cara es 0.5.

Paso 3: Calcular la probabilidad de cara como el número de caras/número total de lanzamientos.

```
> # Ejecutemos un experimento una gran cantidad de veces (hasta n)
> # y veamos cómo el promedio de caras -> probabilidad de cara convergen a un valor
> for (i in 2 :n)
+ {
+   dt$count_of_heads[i] <-ifelse(dt$binomial[i] ==1, dt$count_of_heads[i]<-
dt$count_of_heads[i -1]+1, dt$count_of_heads[i -1])
+   dt$mean[i] <-dt$count_of_heads[i] /i
+ }
```

En cada paso, determinamos si el resultado es cara o cruz. Luego, contamos el número de caras y lo dividimos entre el número de ensayos para obtener una proporción estimada de caras.

Al realizar el mismo experimento un gran número de veces, LLN indica que convergerá a la probabilidad (expectativa o media) de obtener cara en un experimento. Por ejemplo, en el ensayo 30, contaremos cuántas caras hemos obtenido hasta el momento y dividiremos entre 30 para obtener el promedio de caras.

Paso 4: Grafica y observa cómo el promedio de la muestra converge a p = 0.60.

```
> # Plot the average no. of heads -> probability of heads at each experiment stage

> plot(dt$mean, type='l', main ="Simulation of average no. of heads",
+   xlab="Size of Sample", ylab="Sample mean of no. of Heads")

> abline(h = p, col="red")
```

La figura 3 muestra que, a medida que aumenta el número de experimentos, la probabilidad converge a la probabilidad real de cara (0.6). Se recomienda realizar el experimento un gran número de veces para observar la convergencia exacta.

Este teorema nos ayuda a estimar probabilidades desconocidas mediante el método experimental y a crear la distribución para las pruebas de inferencia.

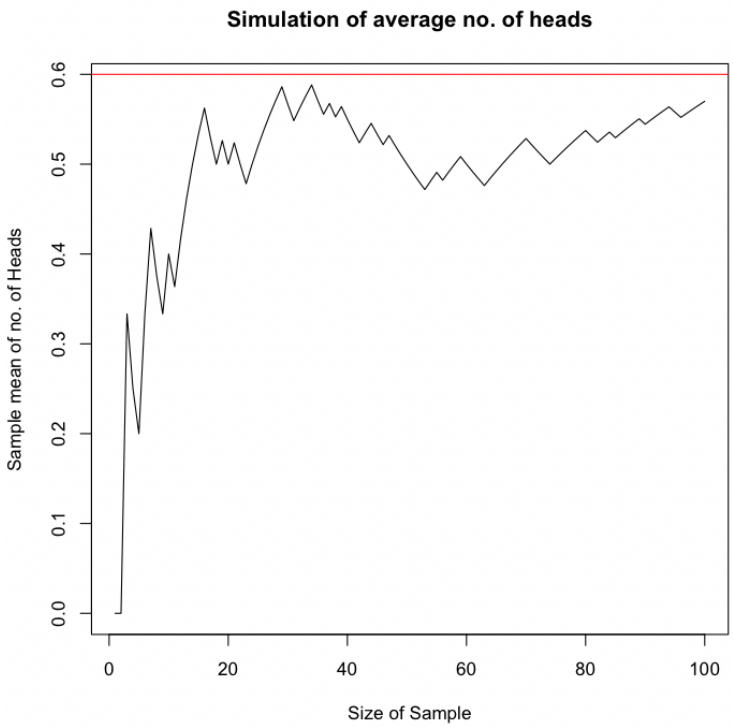

Figura 3: Simulación del experimento de lanzamiento de moneda.

Teorema del límite central

El Teorema del Límite Central es otro teorema muy importante en la teoría de la probabilidad, que permite la comprobación de hipótesis utilizando la distribución muestral. En pocas palabras, el Teorema del Límite Central (TLC) establece que los promedios muestrales de un gran número de iteraciones de variables aleatorias independientes, cada una con medias y varianzas bien definidas, se distribuyen aproximadamente de forma normal.

La primera explicación escrita de este concepto fue proporcionada por de Moivre en su obra de principios del siglo XVIII, cuando utilizó la distribución normal para aproximar el número de caras obtenidas al lanzar una moneda. Pierre-Simon Laplace publicó *Théorie analytique des probability* en 1812, donde amplió la idea de de Moivre al aproximar la distribución binomial con la distribución normal. La demostración precisa del TLC fue proporcionada por Aleksandr Lyapunov en 1901, quien lo definió en términos generales y demostró con precisión su funcionamiento matemático.

En probabilidad, este es uno de los teoremas más populares, junto con la Ley de los Grandes Números. En el contexto de estas notas, enunciaremos matemáticamente la versión más popular del Teorema del Límite Central (Teorema del Límite Central de Lindeberg-Levy).

Para una secuencia de variables aleatorias i.i.d {X1, X2, ...} con una esperanza y varianza bien definidas (E[Xi] = μ y Var[Xi] = $\sigma^2 < \infty$), a medida que n tiende a infinito $\sqrt{n}(S_n - \mu)$, converge en una distribución a una distribución normal N(0, σ^2),

$$\sqrt{n}\left(\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) - \mu\right) \xrightarrow{d} N(0, \sigma^2)$$

Existen otras versiones de este teorema, como el Teorema del Límite Central de Lyapunov, el Teorema del Límite Central de Lindeberg, el Teorema del Límite Central de Diferencia Martingala y muchas más. Es importante comprender cómo se relacionan la Ley de los Grandes Números y el Teorema del Límite Central en nuestro contexto de muestreo.

La Ley de los Grandes Números establece que la media muestral converge a la media poblacional a medida que aumenta el tamaño de la muestra, pero no aborda la distribución de las medias muestrales.

El Teorema del Límite Central nos proporciona información sobre la distribución alrededor de la media y afirma que converge a una distribución normal para un gran número de ensayos. Conocer la distribución nos permite realizar pruebas inferenciales, ya que podemos crear límites de confianza para una distribución normal.

De nuevo, presentaremos un ejemplo sencillo para explicar el teorema. Como ejemplo, comenzaremos el muestreo a partir de una distribución exponencial y mostraremos la distribución de la media muestral.

Pasos de la simulación con código R

Paso 1: Establecer un número de muestras (por ejemplo, r = 5000) para extraer de una población mixta.

```
> #Number of samples
> r<-5000
> #Size of each sample
> n<-10000
```

En el código anterior, r representaba el número de muestras a extraer y n el número de unidades en cada muestra. Según el método del TLC, a mayor número de muestras, mejor convergencia a una distribución normal.

Paso 2: Comienza el muestreo extrayendo una muestra de tamaño n (digamos n = 10,000 cada una). Extrae muestras de distribuciones normal, uniforme, de Cauchy, gamma y otras para comprobar el teorema en diferentes distribuciones. En este caso, consideramos una distribución exponencial con el parámetro $\lambda = 0.6$.

```
> # Generar una matriz de observaciones con n columnas y r filas.
> # Cada fila representa una muestra.
> lambda<-0.6
> Exponential_Samples =matrix(rexp(n*r,lambda),r)
```

Ahora, el frame de datos Exponential_Samples contiene la serie de muestras i.i.d. (*independientes e idénticamente distribuidas*) extraídas de una distribución exponencial con el parámetro $\lambda = 0.6$.

Paso 3: Calcula la suma, la media y la varianza de todas las muestras para cada muestra.

```
> all.sample.sums <-apply(Exponential_Samples,1,sum)
> all.sample.means <-apply(Exponential_Samples,1,mean)
> all.sample.vars <-apply(Exponential_Samples,1,var)
```

En el paso anterior, se calculó la suma, la media y la varianza de todas las muestras i.i.d. En el siguiente paso, observaremos la distribución de las sumas, las medias y las varianzas. Según el TLC, observaremos que la media sigue una distribución normal.

Paso 4: Graficar la suma, las medias y las varianzas combinadas.

```
> par(mfrow=c(2,2))
> hist(Exponential_Samples[1,],col="gray",main="Distribution of One Sample")
> hist(all.sample.sums,col="gray",main="Sampling Distribution of the Sum")
> hist(all.sample.means,col="gray",main="Sampling Distribution of the Mean")
> hist(all.sample.vars,col="gray",main="Sampling Distribution of the Variance")
```

La figura 4 muestra los gráficos de una muestra exponencial y la suma, la media y la desviación estándar de todas las muestras r.

La figura 4 muestra la distribución de los estadísticos de las muestras, es decir, la suma, la media y la varianza. El primer gráfico muestra el histograma de la primera muestra de la distribución exponencial. Se puede observar que la distribución de unidades en la muestra es exponencial. La inspección visual muestra que los estadísticos estimados a partir de muestras i.i.d. siguen una distribución cercana a la normal.

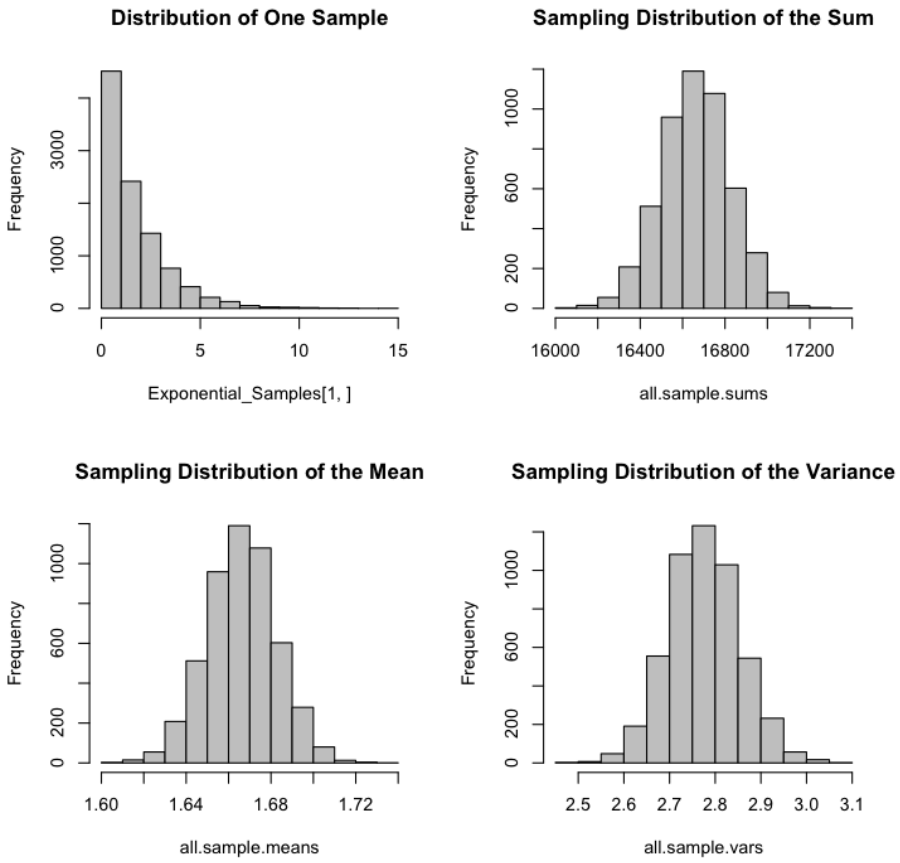

Figura 4: Gráficos de distribución muestral.

Paso 5: Repite este experimento con otras distribuciones y comprueba que los resultados sean consistentes con el TLC para todas las distribuciones. Por brevedad, no se muestran en estas notas.

Existen otras distribuciones estándar que pueden utilizarse para validar los resultados del Teorema del Límite Central. Nuestro ejemplo anterior analizó la distribución exponencial. Se recomienda utilizar la siguiente distribución para validar el Teorema del Límite Central.

```
> Normal_Samples =matrix(rnorm(n*r,param1,param2),r),
+ Uniform_Samples =matrix(runif(n*r,param1,param2),r),
+ Poisson_Samples =matrix(rpois(n*r,param1),r),
+ Cauchy_Samples =matrix(rcauchy(n*r,param1,param2),r),
+ Bionomial_Samples =matrix(rbinom(n*r,param1,param2),r),
+ Gamma_Samples =matrix(rgamma(n*r,param1,param2),r),
+ ChiSqr_Samples =matrix(rchisq(n*r,param1),r),
+ StudentT_Samples =matrix(rt(n*r,param1),r))
```

Es recomendable no basarse en la inspección visual y realizar pruebas formales para inferir las propiedades de la distribución.

El histograma y una prueba formal de normalidad son una buena manera de establecer, tanto visualmente como mediante pruebas paramétricas, que la distribución de medias se distribuye normalmente (como afirma el TLC).

A continuación, realizamos una prueba de Shapiro-Wilk para comprobar la normalidad de la distribución de medias. Otras pruebas de normalidad se analizan brevemente en los libros de probabilidad. Una de las pruebas de normalidad no paramétricas más populares es la prueba KS de una muestra.

```
#Do a formal test of normality on the distribution of sample means
> Mean_of_sample_means <-mean (all.sample.means)

> Variance_of_sample_means <-var(all.sample.means)
> # testing normality by Shapiro  wilk test

> shapiro.test(all.sample.means)

        Shapiro-Wilk normality test

data:  all.sample.means
W = 0.99965, p-value = 0.5545
```

Puede observarse que el valor p es significativo (> 0.05) en la prueba de Shapiro-Wilk, lo que significa que no podemos rechazar la hipótesis nula de que la distribución se distribuye normalmente. La distribución se distribuye normalmente, con una media = 1.66 y una varianza = 0.00027.

La inspección visual se puede realizar mediante histogramas. Además, para mayor claridad, superpongamos la función de densidad normal al histograma para confirmar si la distribución es normal.

```
> x <-all.sample.means
> h<-hist(x, breaks=20, col="red", xlab="Sample Means",
+  main="Histogram with Normal Curve")
> xfit<-seq(min(x),max(x),length=40)
> yfit<-dnorm(xfit,mean=Mean_of_sample_means,sd=sqrt(Variance_of_sample_means))
> yfit <-yfit*diff(h$mids[1:2])*length(x)
> lines(xfit, yfit, col="blue", lwd=2)
```

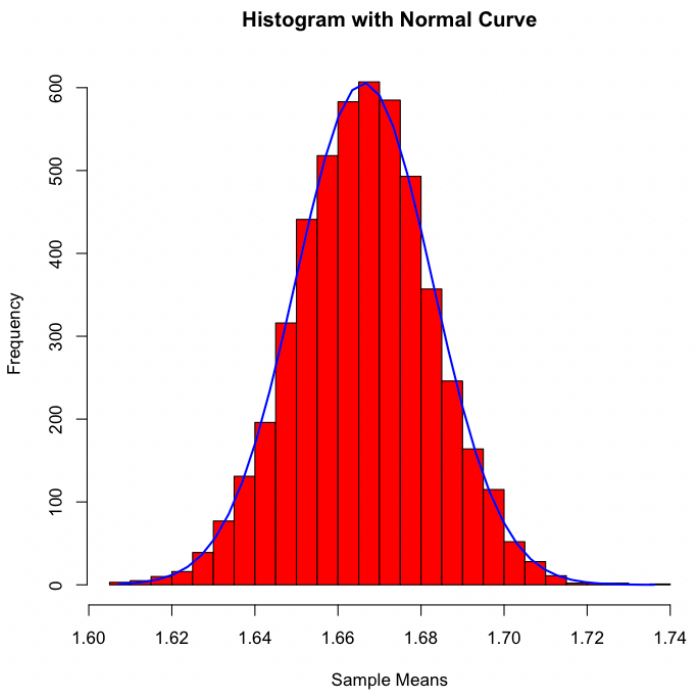

Figura 5: Distribución de medias muestrales con líneas de densidad normal.

Los puntos más importantes a recordar sobre la Ley de los Grandes Números y el TLC son:

- A medida que aumenta el tamaño de la muestra, se puede esperar una mejor estimación de los parámetros de la población/modelo. Dicho esto, un tamaño de muestra grande proporcionará estimaciones insesgadas y más precisas para las pruebas de hipótesis.
- El Teorema del Límite Central ayuda a obtener una distribución y, por lo tanto, permite obtener un intervalo de confianza en torno a los parámetros y aplicar pruebas de inferencia. Lo importante es que el TLC no asume ninguna distribución de la población de la que se extraen las muestras, lo que evita tener que hacer suposiciones sobre la distribución.

Técnicas de muestreo probabilístico

En esta sección, presentamos algunas de las técnicas de muestreo probabilístico más populares y mostramos cómo aplicarlas con R. Todas las técnicas de muestreo se explican utilizando nuestros datos de fraude con tarjetas de crédito. Como primer paso para explicar las técnicas individuales, crearemos las estadísticas y la distribución poblacional y luego compararemos las mismas propiedades muestrales con las propiedades poblacionales para determinar los resultados del muestreo.

Estadísticas poblacionales

Analizaremos algunas características básicas de los datos. Estas características se denominarán estadísticas/parámetros poblacionales. A continuación, mostraremos diferentes métodos de muestreo y compararemos los resultados con las estadísticas poblacionales. Se repite el proceso de cargar los datasets, se repiten nuevamente:

```
> library(data.table)
> data <- fread("ccFraud.csv",header=T, verbose = FALSE, showProgress = FALSE)
> US_state <- fread("US_State_Code_Mapping.csv",header=T, showProgress = FALSE )
> data<-merge(data, US_state, by = 'state')
> Gender_map<-fread("Gender Map.csv",header=T)
> data<-merge(data, Gender_map, by = 'gender')
> Credit_line<-fread("credit line map.csv",header=T)
> data<-merge(data, Credit_line, by = 'creditLine')
> setnames(data,"custID","CustomerID")
> setnames(data,"code","Gender")
> setnames(data,"numTrans","DomesTransc")
> setnames(data,"numIntlTrans","IntTransc")
> setnames(data,"fraudRisk","FraudFlag")
> setnames(data,"cardholder","NumOfCards")
> setnames(data,"balance","OutsBal")
> setnames(data,"StateName","State")
```

1. Dimensiones de los datos de población
str() muestra los nombres de las columnas, su tipo y algunos valores. Se puede observar que el conjunto de datos combina números enteros y caracteres.

```
> str(data)
```

```
Classes 'data.table' and 'data.frame':10000000 obs. of  14 variables:
 $ creditLine : int  1 1 1 1 1 1 1 1 1 1 ...
 $ gender     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ state      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ CustomerID : int  4446 59161 136032 223734 240467 248899 262655 324670 390138 482698 ...
 $ NumOfCards : int  1 1 1 1 1 1 1 1 1 1 ...
 $ OutsBal    : int  2000 0 2000 2000 2000 0 0 689 2000 0 ...
 $ DomesTransc: int  31 25 78 11 40 47 15 17 48 25 ...
 $ IntTransc  : int  9 0 3 0 0 0 0 9 0 35 ...
 $ FraudFlag  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ State      : chr  "Alabama" "Alabama" "Alabama" "Alabama" ...
 $ PostalCode : chr  "AL" "AL" "AL" "AL" ...
 $ Gender     : chr  "Male" "Male" "Male" "Male" ...
 $ CardType   : chr  "American Express" "American Express" "American Express" "American Express" ...
 $ CardName   : chr  "SimplyCash\256 Business Card from American Express" "SimplyCash\256 Business Card from American Express"
"SimplyCash\256 Business Card from American Express" "SimplyCash\256 Business Card from American Express" ...
 - attr(*, ".internal.selfref")=<externalptr>
 - attr(*, "sorted")= chr "creditLine"
```

2. Media de la población para las medidas

a. Saldo pendiente: En promedio, cada tarjeta tiene un saldo pendiente de $4109.92.

```
> mean_outstanding_balance <- mean(data$OutsBal)
> mean_outstanding_balance
[1] 4109.92
```

b. Número de transacciones internacionales: El promedio de transacciones internacionales es de 4.04.

```
> mean_international_trans <- mean(data$IntTransc)
> mean_international_trans
[1] 4.04719
```

c. Número de transacciones nacionales: El promedio de transacciones nacionales es muy alto en comparación con las internacionales; el número oscila entre 28.9 y 29 transacciones.

```
> mean_domestic_trans <- mean(data$DomesTransc)
> mean_domestic_trans
[1] 28.93519
```

3. Varianza de la población para las medidas

a. Saldo pendiente:

```
> Var_outstanding_balance <- var(data$OutsBal)
> Var_outstanding_balance
[1] 15974788
```

b. Número de transacciones internacionales:

```
> Var_international_trans <- var(data$IntTransc)
> Var_international_trans
[1] 74.01109
```

c. Número de transacciones nacionales:

```
> Var_domestic_trans <- var(data$DomesTransc)
> Var_domestic_trans
[1] 705.1033
```

4. Histograma

a. Saldo pendiente:

```
> hist(data$OutsBal, breaks=20, col="red", xlab="Outstanding Balance",
+   main="Distribution of Outstanding Balance")
```

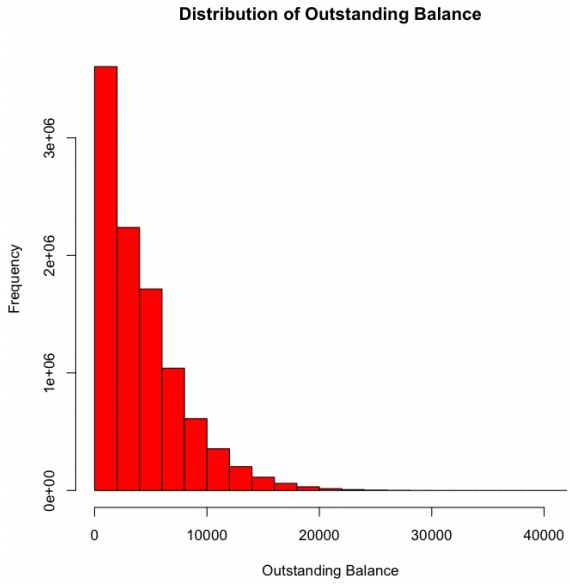

Figura 6: Histograma del saldo pendiente.

b. Número de transacciones internacionales:

```
> hist(data$IntTransc, breaks=20, col="blue", xlab="Number of International Transactions",
+   main="Distribution of International Transactions")
```

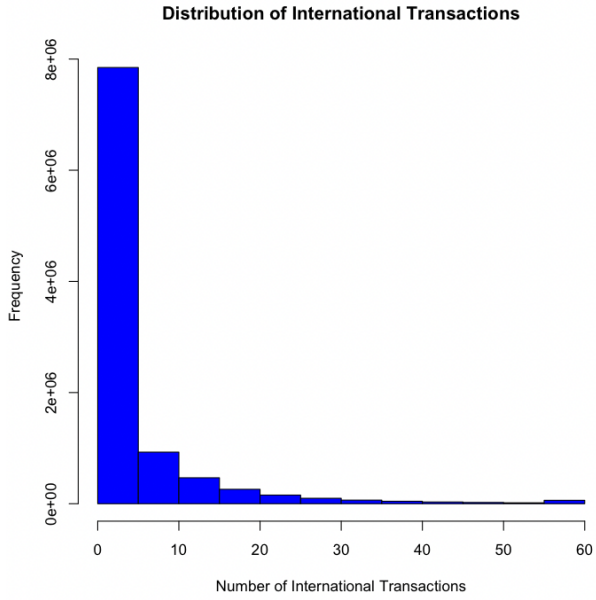

Figura 7: Histograma del número de transacciones internacionales.

c. Número de transacciones nacionales:

```
> hist(data$DomesTransc, breaks=20, col="green", xlab="Number of Domestic Transactions",
+   main="Distribution of Domestic Transactions")
```

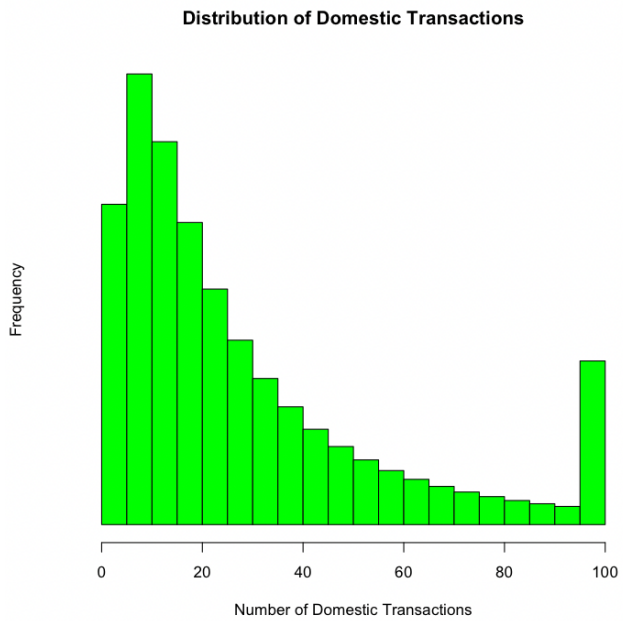

Figura 8: Histograma del número de transacciones nacionales.

La figura 8 muestra la media, la varianza y la distribución de algunas variables importantes de nuestro conjunto de datos sobre fraude con tarjetas de crédito. Estas estadísticas poblacionales se compararán con las estadísticas muestrales para determinar qué técnicas de muestreo proporcionan una muestra representativa.

Muestreo aleatorio simple

El *muestreo aleatorio simple* consiste en seleccionar una muestra de la población, donde cada unidad se selecciona aleatoriamente. Cada unidad tiene la misma probabilidad individual de ser elegida en cualquier etapa del proceso de muestreo, y el subconjunto de k individuos tiene la misma probabilidad de ser elegido para la muestra que cualquier otro subconjunto de k individuos.

El muestreo aleatorio simple es un tipo básico de muestreo, por lo que puede ser un componente de metodologías de muestreo más complejas. En los próximos temas, verás que el muestreo aleatorio simple es un componente importante de otros métodos de muestreo probabilístico, como el muestreo estratificado y el muestreo por cluster.

El muestreo aleatorio simple suele ser sin reemplazo; es decir, el diseño del proceso de muestreo garantiza que ninguna unidad pueda seleccionarse más de una vez. Sin embargo, el muestreo aleatorio simple puede realizarse con reemplazo, pero en ese caso las unidades de muestreo no serán independientes.

Si se extrae una muestra pequeña de una población grande, el muestreo sin reemplazo y el muestreo con reemplazo arrojarán aproximadamente los mismos resultados, ya que la probabilidad de que cada unidad sea seleccionada es muy pequeña. La tabla 5 compara las estadísticas del muestreo aleatorio simple con y sin reemplazo. Los valores son comparables, ya que el tamaño de la población es muy grande (aproximadamente 10 millones). Veremos este hecho en nuestro ejemplo.

Tabla 5: Tabla de comparación con el muestreo poblacional con y sin reemplazo.

CardType	OutstandingBalance_ Population	OutstandingBalance_ Random_WOR	OutstandingBalance_ Random_WR
American Express	3820.896	3835.064	3796.138
Discover	4962.420	4942.690	4889.926
MasterCard	3818.300	3822.632	3780.691
Visa	4584.042	4611.649	4553.196

Ventajas:

- Libre de errores de clasificación.
- No se requieren conocimientos avanzados de la población.
- Fácil interpretación de los datos muestrales.

Desventajas:

- Se requiere un frame muestral completo (población) para obtener una muestra representativa.
- La recuperación y el almacenamiento de datos aumentan el costo y el tiempo.
- El muestreo aleatorio simple conlleva el sesgo y los errores presentes en la población, y se requieren intervenciones adicionales para eliminarlos.

Función: Resumir (*Summarise*)

Resumir es una función de la biblioteca dplyr. Esta función ayuda a agregar los datos por dimensiones. Funciona de forma similar a una tabla dinámica en Excel.

- group_by: Este argumento toma la variable categórica mediante la cual se desean agregar las medidas.

- mean(OutsBal): Este argumento proporciona la función de agregación y el nombre del campo donde se realizará la agregación.

```
> #Datos de población: Distribución del saldo pendiente según el tipo de tarjeta
> library(dplyr)
> summarise(group_by(data,CardType),Population_OutstandingBalance=mean(OutsBal))
```

```
# A tibble: 4 x 2
  CardType         Population_OutstandingBalance
  <chr>                                    <dbl>
1 American Express                         3821.
2 Discover                                 4962.
3 MasterCard                               3818.
4 Visa                                     4584.
```

La llamada a la función *summarise* por CardType muestra el saldo pendiente promedio por tipo de tarjeta. Las tarjetas Discover tienen el saldo pendiente promedio más alto. A continuación, extraemos una muestra aleatoria de 100,000 registros mediante la función integrada sample() de la biblioteca base. Esta función crea un frame de muestreo seleccionando aleatoriamente índices de datos. Una vez obtenido el frame de muestreo, extraemos los registros correspondientes de los datos de la población.

Función: Sample

Observa algunos argumentos importantes de la función sample():

- nrow(data): Indica el tamaño de los datos. En este caso, es 10,000,000, por lo que creará un índice de 1 a 10,000,000 y luego seleccionará aleatoriamente el índice para el muestreo.
- size: Permite a los usuarios indicar cuántos puntos de datos se muestrearán de la población. En nuestro caso, hemos establecido n en 100,000.
- replace: Este argumento permite a los usuarios indicar si el muestreo debe realizarse sin reemplazo (FALSO) o con reemplazo (VERDADERO).
- prob: Este es el vector de probabilidades para obtener el frame muestral. Lo hemos establecido como NULL, para que todos tengan el mismo peso/probabilidad.

```
> set.seed(937)
> # Simple Random Sampling Without Replacement
> library("base")
> sample_SOR_100K <-data[sample(nrow(data),size=100000,
+   replace =FALSE, prob =NULL),]
```

Ahora, veamos de nuevo cómo se ve el balance promedio para la muestra aleatoria simple. Como puedes ver, se ha mantenido el orden de los balances promedio. Nótese que el

promedio es muy cercano al promedio poblacional, calculado en el paso anterior para la población.

```
> #Sample Data : Distribution of Outstanding Balance across Card Type
>           summarise(group_by(sample_SOR_100K,CardType),Sample_OutstandingBalance=mean
(OutsBal))
```

```
# A tibble: 4 x 2
  CardType         Sample_OutstandingBalance
  <chr>                                <dbl>
1 American Express                     3785.
2 Discover                             4952.
3 MasterCard                           3793.
4 Visa                                 4536.
```

Función: KS.test()

Esta es una de las pruebas no paramétricas para comparar las funciones de distribución empírica de los datos. Esta función ayuda a determinar si los puntos de datos provienen de la misma distribución. Puede realizarse como una prueba de una muestra (es decir, comparando la Función de Distribución Empírica (EDF, *Empirical Distribution Function*) de los datos con una PDF (*Probability Density Function*) de distribución predefinida (normal, Cauchy, etc.) o como una prueba de dos muestras (es decir, cuando queremos comprobar si la distribución de dos muestras es la misma).

Dado que una de las características importantes del muestreo es asegurar que la distribución de los datos no cambie después del muestreo (excepto cuando se realiza intencionalmente), utilizaremos pruebas de dos muestras para comprobar si la muestra representa fielmente a la población, comprobando si la población y la muestra pertenecen a la misma distribución.
Los argumentos importantes son las dos series de datos y la hipótesis a probar: dos colas, una cola. Para este ejemplo, elegimos la prueba de dos colas, más conservadora. Esta prueba significa que queremos asegurarnos de que se utilice la igualdad en la hipótesis nula.

```
> # Comprobar si los datos muestreados provienen de la población o no.
> # Esto garantiza que el muestreo no altere la distribución original.
> ks.test(data$OutsBal,sample_SOR_100K$OutsBal,alternative="two.sided")
```

```
    Asymptotic two-sample Kolmogorov-Smirnov test

data:  data$OutsBal and sample_SOR_100K$OutsBal
D = 0.0053575, p-value = 0.006802
alternative hypothesis: two-sided
```

Los resultados de la prueba KS indican claramente que la muestra y la población tienen la misma distribución. Por lo tanto, podemos afirmar que el muestreo no ha modificado la distribución. Por la naturaleza del muestreo, el muestreo aleatorio simple sin reemplazo conserva la distribución de los datos. Para una inspección visual, la figura 9 muestra los histogramas de la población y la muestra. Como puede observarse, la distribución es la misma para ambos.

```
> par(mfrow =c(1,2))
> hist(data$OutsBal, breaks=20, col="red", xlab="Outstanding Balance",
+   main="Population")
> hist(sample_SOR_100K$OutsBal, breaks=20, col="green", xlab="Outstanding Balance",
+   main="Random Sample Data (without replacement)")
```

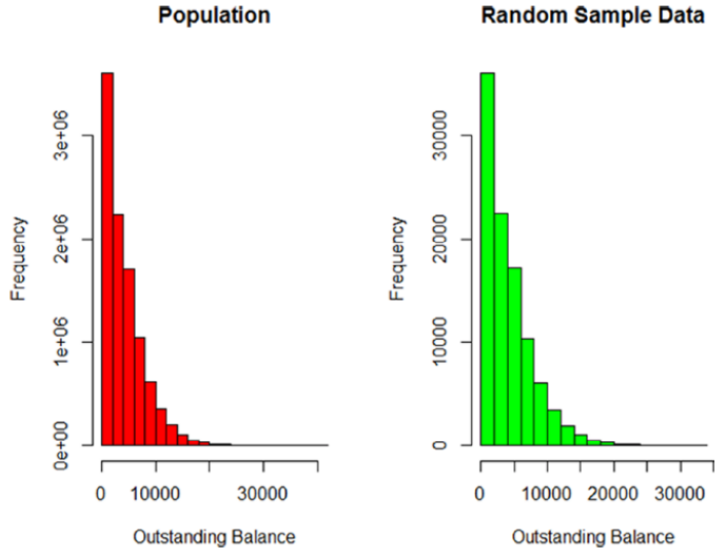

Figura 9: Distribución de la población frente a la muestra (sin reemplazo).

Ahora realizaremos una prueba formal sobre la media del saldo pendiente de la población y nuestra muestra aleatoria. Teóricamente, esperamos que la prueba t con medias de dos sea verdadera y, por lo tanto, podemos afirmar que la media de la muestra y la población son iguales con un 95 % de confianza.

```
> # Realicemos también una prueba t para la media de la población y la muestra.
> t.test(data$OutsBal, sample_SOR_100K$OutsBal)
    Welch Two Sample t-test

data:  data$OutsBal and sample_SOR_100K$OutsBal
t = 2.6436, df = 102019, p-value = 0.008205
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  8.661026 58.328252
sample estimates:
mean of x mean of y
 4109.920  4076.425
```

Estos resultados muestran que las medias de la población y la muestra son iguales, ya que el valor p de la prueba t es insignificante. No podemos rechazar la hipótesis nula de que las medias son iguales.

Aquí te mostraremos una prueba similar realizada para una muestra aleatoria simple con reemplazo. Como verás, no observamos ningún cambio significativo en los resultados en comparación con la muestra aleatoria simple, ya que el tamaño de la población es muy grande y el reemplazo no altera sustancialmente la probabilidad de registro del muestreo.

```
> set.seed(937)
> # Simple Random Sampling With Replacement
> sample_SR_100K <-data[sample(nrow(data),size=100000, replace =TRUE, prob =NULL),]
```

En este código, para el muestreo aleatorio simple con reemplazo, establecemos "replace" como TRUE en la llamada a la función de ejemplo.

El siguiente código muestra cómo realizamos la prueba KS sobre la distribución de la población y la muestra extraída con reemplazo. La prueba muestra que las distribuciones son iguales y que el valor p es insignificante, lo que no rechaza la hipótesis nula de distribución igual.

```
> ks.test(data$OutsBal,sample_SR_100K$OutsBal,alternative="two.sided")

    Asymptotic two-sample Kolmogorov-Smirnov test

data:  data$OutsBal and sample_SR_100K$OutsBal
D = 0.0052975, p-value = 0.00772
alternative hypothesis: two-sided
```

Creamos el histograma de la población y la muestra con reemplazo. Los gráficos son idénticos; al aplicar una prueba KS formal, observamos que tanto la muestra con reemplazo como las poblaciones tienen la misma distribución. Ten cuidado con esto cuando el tamaño de la población sea pequeño.

```
> par(mfrow =c(1,2))
> hist(data$OutsBal, breaks=20, col="red", xlab="Outstanding Balance",
+   main="Population ")
> hist(sample_SR_100K$OutsBal, breaks=20, col="green", xlab="Outstanding Balance",
+   main=" Random Sample Data ( WR)")
```

La distribución es similar para la población y la muestra aleatoria extraída con reemplazo. Resumimos el muestreo aleatorio simple comparando los resultados resumidos de la

población, la muestra aleatoria simple sin reemplazo y la muestra aleatoria simple con reemplazo.

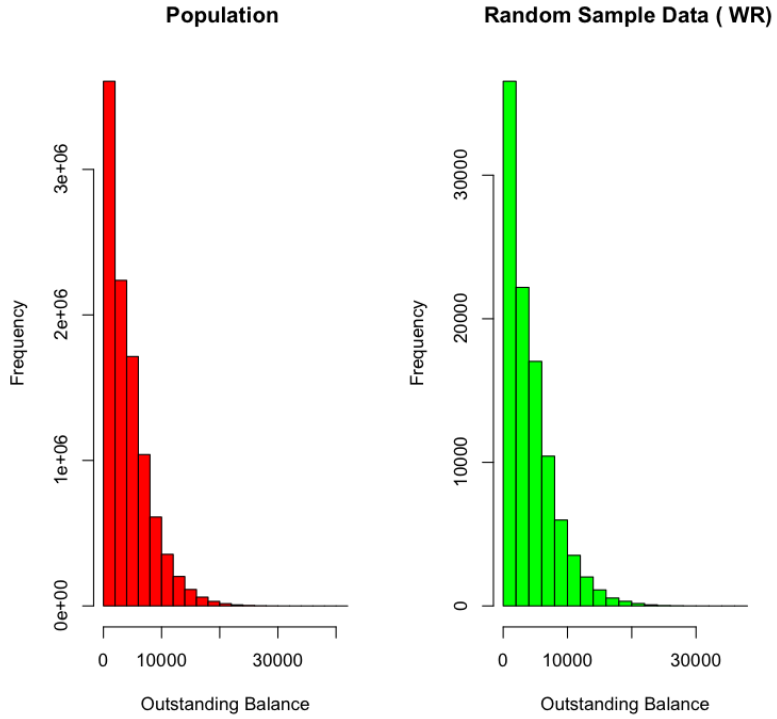

Figura 10: Distribución de la población frente a la muestra (con reemplazo).

La siguiente tabla muestra que, tanto con reemplazo como sin reemplazo, el muestreo aleatorio simple arrojó valores de media similares en todos los tipos de tarjeta, muy cercanos a la media real de la población.

```
> library(knitr)
> population_summary <-summarise(group_by(data,CardType),
+   OutstandingBalance_Population=mean(OutsBal))
> random_WOR_summary<-summarise(group_by(sample_SOR_100K,CardType),
+   OutstandingBalance_Random_WOR=mean(OutsBal))
> random_WR_summary<-summarise(group_by(sample_SR_100K,CardType),
+   OutstandingBalance_Random_WR=mean(OutsBal))
> compare_population_WOR<-merge(population_summary,random_WOR_summary,
+   by="CardType")
> compare_population_WR <-merge(population_summary,random_WR_summary,
+   by="CardType")
> summary_compare<-
cbind(compare_population_WOR,compare_population_WR$OutstandingBalance_Random_WR)
> colnames(summary_compare)[which(names(summary_compare) == "V2")] <-
"OutstandingBalance_Random_WR"
> knitr::kable(summary_compare, col.names = c("C_Type",
"OutBal_Population","OutBal_Random_WOR",
```

+ "OutBal_Random_WR"))

C_Type	OutBal_Population	OutBal_Random_WOR	OutBal_Random_WR
American Express	3820.896	3785.138	3787.115
Discover	4962.420	4951.778	4968.974
MasterCard	3818.300	3793.253	3791.226
Visa	4584.042	4536.006	4535.001

Puntos clave:

- El muestreo aleatorio simple proporciona muestras representativas de la población.
- El muestreo con y sin reemplazo puede arrojar resultados diferentes con distintos tamaños de muestra, por lo que se debe tener especial cuidado al elegir el método cuando el tamaño de la población es pequeño.
- El tamaño de muestra adecuado para cada problema varía según la confianza que se desee obtener con las pruebas, los objetivos comerciales, el análisis de costo-beneficio y otras razones. Obtendrás una buena comprensión de lo que ocurre con cada técnica de muestreo y podrás elegir la que mejor se adapte al problema en cuestión.

Muestreo aleatorio sistemático

El muestreo sistemático es un método estadístico en el que las unidades se seleccionan con un frame de muestreo ordenado sistemáticamente. La forma más popular de muestreo sistemático se basa en un frame de muestreo circular, donde se recorre la población de principio a fin y luego se continúa desde el principio de forma circular. En este enfoque, la probabilidad de que cada unidad sea seleccionada es la misma; por lo tanto, a veces se le denomina método de igual probabilidad. Sin embargo, se pueden crear otros frames sistemáticos según las necesidades del muestreo sistemático.

En esta sección, analizamos el enfoque circular más popular para el muestreo aleatorio sistemático. En este método, el muestreo comienza seleccionando aleatoriamente una unidad de la población y, a continuación, se selecciona cada elemento k. Al finalizar la lista, el muestreo comienza desde el principio. En este caso, el k se conoce como factor de omisión y se calcula de la siguiente manera:

$$k = \frac{N}{n}$$

donde N es el tamaño de la población y n es el tamaño de la muestra.

Este enfoque del muestreo sistemático lo hace funcionalmente similar al muestreo aleatorio simple, pero no es lo mismo, ya que no todas las muestras posibles de un tamaño determinado tienen la misma probabilidad de ser seleccionadas (por ejemplo, el valor de semilla garantizará que los elementos adyacentes nunca se seleccionen en el frame muestral). Sin embargo, este método es eficiente si la varianza dentro de la muestra sistemática es mayor que la varianza de la población.

Ventajas:

- Fácil de implementar
- Puede ser más eficiente

Desventajas:

- Se puede aplicar cuando la población es lógicamente homogénea
- Puede haber un patrón oculto en el frame muestral, lo que causa sesgos no deseados

Creemos un ejemplo de muestreo sistemático a partir de nuestros datos de fraude con tarjetas de crédito.

Paso 1: Identificar un subconjunto de la población que se pueda asumir como homogéneo. Una posible opción es subdividir la población por estado. En este ejemplo, utilizamos Rhode Island, el estado más pequeño de Estados Unidos, para asumir homogeneidad.

A modo de ejemplo, crearemos un conjunto homogéneo mediante la subdivisión de la población con la siguiente lógica de negocios. Subconjunto los datos y extraigamos los registros cuyas transacciones internacionales sean iguales a 0 y las transacciones nacionales sean menores o iguales a 3.

Suponiendo que el subconjunto anterior forma un conjunto de población homogénea, la suposición de la subdivisión también es parcialmente cierta, ya que es probable que los clientes que no usan tarjetas en el país no las usen en absoluto en el extranjero.

```
> Data_Subset <-subset(data, IntTransc==0&DomesTransc<=3)
> summarise(group_by(Data_Subset,CardType),OutstandingBalance=mean(OutsBal))
```

```
# A tibble: 4 x 2
  CardType         OutstandingBalance
  <chr>                         <dbl>
1 American Express              3828.
2 Discover                      4924.
3 MasterCard                    3807.
4 Visa                          4579.
```

Suponiendo que el subconjunto tiene conjuntos homogéneos de titulares de tarjetas por tipo de tarjeta, podemos proceder con el muestreo sistemático. Si el conjunto no es homogéneo, es muy probable que el muestreo sistemático genere una muestra sesgada y, por lo tanto, no proporcione una representación fiel de la población. Además, sabemos que los datos se almacenan en el frame de datos R mediante un índice interno (que será el mismo que nuestro ID de cliente), por lo que podemos confiar en el índice ordenado internamente para el muestreo sistemático.

Paso 2: Establecer un tamaño de muestra para muestrear de la población.

```
> #Size of population ( here the size of card holders from Data Subset)
> Population_Size_N<-length(Data_Subset$OutsBal)
> # Set a the size of sample to pull (should be less than N), n. We will
> # assume n=5000
> Sample_Size_n<-5000
```

Paso 3: Calcular el factor de salto utilizando esta fórmula: k= N/n.

El factor de salto proporcionará el factor de salto al crear el frame de muestreo sistemático. En esencia, con una semilla (o índice inicial) de c, se seleccionarán los elementos tras omitir k elementos en orden.

```
> #Calculate the skip factor
> k =ceiling(Population_Size_N/Sample_Size_n)
> #ceiling(x) rounds to the nearest integer thatâ<U+0080><U+0099>s larger than x.
> #This means ceiling (2.3) = 3
> cat("The skip factor for systematic sampling is ",k)
The skip factor for systematic sampling is  62
```

Paso 4: Establecer un valor de semilla aleatorio para el índice y luego crea un vector de secuencia con la semilla y el salto (frame de muestreo). Esto tomará un valor de semilla para el índice, por ejemplo, i, y luego creará un frame de muestreo como i, i+k, i+2k... uno hasta tener un total de n índices (tamaño de muestra) en el frame de muestreo.

```
> r =sample(1:k, 1)
> systematic_sample_index =seq(r, r +k*(Sample_Size_n-1), k)
```

Paso 5: Muestrear los registros de la población mediante el frame de muestreo. Una vez que tengamos listo nuestro frame de muestreo, no es más que una lista de índices, por lo que extraemos los registros de datos correspondientes al frame de muestreo.

```
> systematic_sample_5K<-Data_Subset[systematic_sample_index,]
```

Ahora comparemos la muestra sistemática con una muestra aleatoria simple del mismo tamaño de 5000. Como se explicó anteriormente, sabemos que el muestreo aleatorio simple es una representación fiel de la población, por lo que podemos usarlo como proxy de las propiedades de la población.

```
> set.seed(937)
> # Simple Random Sampling Without Replacement
> sample_Random_5K <-Data_Subset[sample(nrow(Data_Subset),size=5000,
+   replace =FALSE, prob =NULL),]
```

Aquí se presenta el resultado de la comparación resumida por tipo de tarjeta para los saldos pendientes. Esta comparación es importante para mostrar las diferencias en la media que se producirían si se hubiera elegido una muestra aleatoria simple en lugar de una muestra sistemática.

```
> sys_summary <-summarise(group_by(systematic_sample_5K,CardType),
+   OutstandingBalance_Sys=mean(OutsBal))
> random_summary<-summarise(group_by(sample_Random_5K,CardType),
+   OutstandingBalance_Random=mean(OutsBal))
> summary_mean_compare<-merge(sys_summary,random_summary, by="CardType")
> print(summary_mean_compare)

          CardType OutstandingBalance_Sys OutstandingBalance_Random
1 American Express               3864.164                  3928.651
2         Discover               4577.148                  4465.852
3       MasterCard               3757.742                  3825.158
4             Visa               4446.465                  4607.684
```

De nuevo, haremos hincapié en comparar la EDF muestral con la EDF poblacional para asegurar que el muestreo no haya distorsionado la distribución de los datos. Estos pasos se repetirán para todas las técnicas de muestreo, ya que esto garantiza la estabilidad del muestreo para fines de modelado.

```
> ks.test(Data_Subset$OutsBal,systematic_sample_5K$OutsBal,alternative="two.sided")

    Asymptotic two-sample Kolmogorov-Smirnov test

data:  Data_Subset$OutsBal and systematic_sample_5K$OutsBal
D = 0.012024, p-value = 0.4805
alternative hypothesis: two-sided
```

Los resultados de la prueba KS muestran que la distribución es la misma y, por lo tanto, la muestra es una representación de la población mediante la distribución. La figura 11 muestra

los histogramas para mostrar la distribución de un subconjunto de datos homogéneos y una muestra sistemática. Se puede observar que la distribución no ha cambiado drásticamente.

```
> par(mfrow =c(1,2))
> hist(Data_Subset$OutsBal, breaks=50, col="red", xlab="Outstanding Balance",
+   main="Homogenous Subset Data")
> hist(systematic_sample_5K$OutsBal, breaks=50, col="green", xlab="Outstanding Balance",
+   main="Systematic Sample")
```

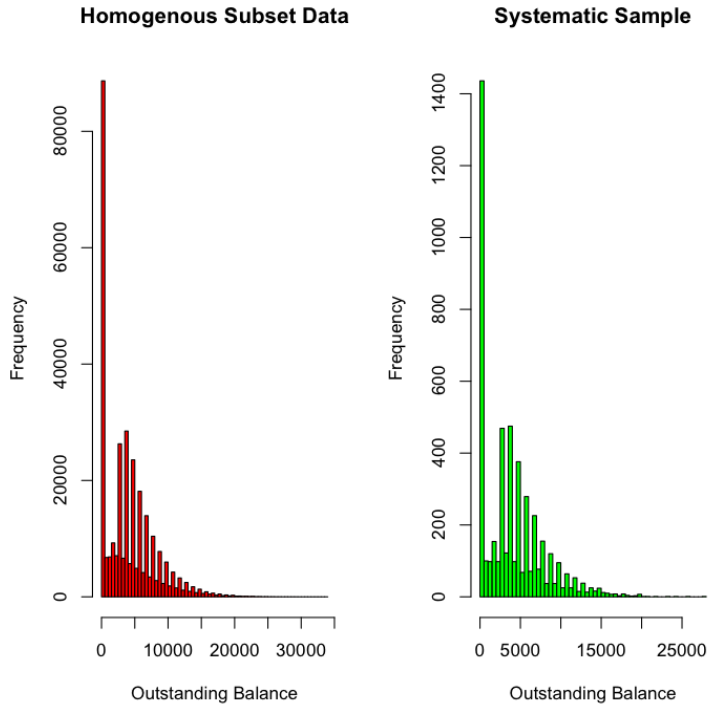

Figura 11: Distribución homogénea de la población y la muestra sistemática.

Puntos clave:

- El muestreo sistemático es equivalente al muestreo aleatorio simple si se realiza sobre un conjunto homogéneo de puntos de datos. Además, un tamaño poblacional grande suprime el sesgo asociado con el muestreo sistemático para fracciones de muestreo más pequeñas.

- La capacidad de negocio y computacional son criterios importantes para elegir una técnica de muestreo cuando el tamaño de la población es grande. En nuestro ejemplo, el muestreo sistemático proporciona una muestra representativa con un menor costo computacional. (No se requiere un generador de números aleatorios, por lo que no es necesario analizar la lista completa de registros).

Muestreo aleatorio estratificado

Cuando la población tiene subpoblaciones que varían, es importante que la técnica de muestreo considere las variaciones a nivel de subpoblación (estrato) y las muestree de forma independiente a nivel de estrato. La estratificación es el proceso de identificar grupos homogéneos caracterizándolos por alguna propiedad intrínseca. Por ejemplo, los clientes que viven en la misma ciudad pueden considerarse pertenecientes a ese estrato.

Los estratos deben ser mutuamente excluyentes y colectivamente exhaustivos; es decir, todas las unidades de la población deben asignarse a algún estrato y una unidad solo puede pertenecer a un estrato.

Una vez formado el estrato, se realiza un muestreo aleatorio simple o sistemático a nivel de estrato de forma independiente. Esto mejora la representatividad de la muestra y, en general, reduce el error de muestreo. Dividir la población en estratos también facilita el cálculo del promedio ponderado de la población, que presenta menor variabilidad que la población total combinada.

Existen dos métodos generalmente aceptados para identificar el tamaño de muestra estratificado:

- *Asignación proporcional*, que muestrea proporciones iguales de los datos de cada estrato. En este caso, se aplica la misma fracción de muestreo a todos los estratos de la población. Por ejemplo, supongamos que su población tiene cuatro tipos de tarjetas de crédito y asume que cada tipo de tarjeta de crédito forma un grupo homogéneo de clientes.
  Supongamos que el número de cada tipo de clientes en cada estrato es N1 + N2 + N3 + N4 = total; entonces, con la asignación proporcional, obtendrá una muestra con la misma proporción de cada estrato (n1/N1 = n2/N2 = n3/N3 = n4/N4 = fracción de muestreo).

- *Asignación óptima*, que muestrea proporciones de datos proporcionales a la desviación estándar de la distribución de la variable del estrato. Esto da como resultado muestras grandes de los estratos con mayor variabilidad, lo que significa que la varianza muestral se reduce.

Otra característica importante del muestreo estratificado es que garantiza que se muestree al menos una unidad de cada estrato, incluso si la probabilidad de que sea seleccionada es cero. Se recomienda limitar el número de estratos y asegurar que cada uno cuente con suficientes unidades para realizar el muestreo.

Ventajas:

- Mayor precisión que el muestreo aleatorio simple del mismo tamaño de muestra.
- Gracias a esta mayor precisión, es posible trabajar con muestras pequeñas y, por lo tanto, reducir costos.
- Evita muestras no representativas, ya que este método muestrea al menos una unidad de cada estrato.

Desventajas:

- No siempre es posible dividir la población en grupos disjuntos.
- La identificación de un estrato homogéneo antes del muestreo conlleva costos administrativos adicionales.
- Un estrato reducido puede limitar el tamaño representativo de la muestra.

Para construir un ejemplo de muestreo estratificado con datos de fraude con tarjetas de crédito, primero debemos verificar los estratos y luego proceder con el muestreo a partir de ellos. En nuestro ejemplo, crearemos un estrato basado en las variables CardType y State.
A continuación, explicamos paso a paso cómo realizar el muestreo estratificado.

Paso 1: Verificar las variables de estrato y su frecuencia en la población.

Supongamos que CardType y State son variables de estrato. En otras palabras, creemos que el tipo de tarjeta y el estado pueden utilizarse como criterios para estratificar a los clientes en categorías lógicas. Aquí se muestran las frecuencias según nuestras variables de estrato. Esperamos que el muestreo estratificado mantenga la misma proporción de registros en la muestra estratificada.

```
> #Frequency table for CardType in Population
> table(data$CardType)

American Express        Discover      MasterCard            Visa
         2474848          642531         4042704         2839917
> #Frequency table for State in Population
> table(data$State)
```

La tabla cruzada desglosa la población total según las variables de estrato: CardType y State. Cada estrato representa un conjunto de clientes con comportamientos similares, ya que pertenecen al mismo estrato.

El siguiente resultado se ha recortado para facilitar su lectura.

Alabama	American Samoa	Arizona	Arkansas	California	Colorado	Connecticut	Delaware
20137	162574	101740	202776	1216069	171774	121802	20603
Florida	Georgia	Guam	Hawaii	Idaho	Illinois	Indiana	Iowa
30333	608630	303984	50438	111775	60992	404720	203143
Kansas	Kentucky	Louisiana	Maine	Maryland	Massachusetts	Michigan	Minnesota
91127	142170	151715	201918	202444	40819	304553	182201
Mississippi	Missouri	Montana	Nebraska	Nevada	New Hampshire	New Jersey	New Mexico
203045	101829	30131	60617	303833	20215	40563	284428
New York	North Carolina	North Dakota	Ohio	Oklahoma	Oregon	Pennsylvania	Rhode Island
81332	91326	608575	364531	122191	121846	405892	30233
South Carolina	South Dakota	Tennessee	Texas	Utah	Vermont	Virginia	Washington
152253	20449	203827	812638	91375	252812	20017	202972
West Virginia	Wisconsin	Wyoming					
182557	61385	20691					

> #Cross table frequency for population data
> table(data$State,data$CardType)

	American Express	Discover	MasterCard	Visa
Alabama	4983	1353	8072	5729
American Samoa	40144	10602	65740	46088
Arizona	25010	6471	41111	29148
Arkansas	50158	12977	82042	57599
California	301183	78154	491187	345545
Colorado	42333	11194	69312	48935
Connecticut	30262	7942	49258	34340
Delaware	4990	1322	8427	5864
Florida	7613	1913	12143	8664
Georgia	150463	39005	246285	172877
Guam	74916	19427	123007	86634
Hawaii	12436	3326	20374	14302
Idaho	27758	7400	44973	31644
Illinois	15147	3913	24632	17300
Indiana	100274	26093	163891	114462
Iowa	50178	12829	81990	58146
Kansas	22333	5902	37075	25817
Kentucky	35375	9082	57307	40406
Louisiana	37656	9736	61252	43071
Maine	49701	13124	81903	57190
Maryland	50340	12827	81775	57502
Massachusetts	10113	2703	16619	11384
Michigan	75330	19611	123321	86291
Minnesota	44994	11746	73481	51980
Mississippi	50407	13065	81906	57667
Missouri	25221	6473	41067	29068
Montana	7377	1946	12143	8665
Nebraska	15099	3960	24384	17174
Nevada	75267	19771	122591	86204
New Hampshire	4963	1316	8238	5698
New Jersey	10000	2571	16344	11648
New Mexico	70648	18224	114794	80762
New York	20129	5269	33073	22861
North Carolina	22384	5789	37159	25994
North Dakota	150854	39105	246193	172423
Ohio	90258	23389	146971	103913
Oklahoma	30382	7751	49395	34663
Oregon	30025	7842	49381	34598
Pennsylvania	100226	26069	164394	115203
Rhode Island	7521	1924	12152	8636
South Carolina	37432	9663	61897	43261
South Dakota	5028	1318	8276	5827
Tennessee	50609	13133	82340	57745
Texas	201190	51858	328909	230681
Utah	22747	5992	36794	25842
Vermont	62606	16230	102225	71751
Virginia	5083	1295	7990	5649
Washington	50302	12876	81871	57923
West Virginia	45274	11771	73609	51903
Wisconsin	15072	3900	25023	17390
Wyoming	5054	1379	8408	5850

Paso 2: Muestreo aleatorio sin reemplazo de cada estrato. Considera muestrear el 10 % del tamaño del estrato.

Elegimos el método más popular de muestreo estratificado, el muestreo proporcional. Muestrearemos el 10 % de los registros de cada estrato.

Función: stratified()

La función stratified toma muestras de un data.frame o un data.table en el que una o más columnas pueden utilizarse como variables de "estratificación" o "agrupación". El resultado es un nuevo data.table con el número especificado de muestras de cada grupo. La sintaxis estándar de la función se muestra a continuación:

stratified(indt, group, size, select = NULL, replace = FALSE, keep.rownames = FALSE, bothSets = FALSE, ...)

- group: Este argumento permite a los usuarios definir las variables de estrato. Aquí hemos elegido CardType y State como variables de estrato. En total, tendremos 4 (tipos de tarjeta) x 52 (estados) estratos para muestrear.
- size: En general, el tamaño puede pasarse como un número (igual número de muestras de cada estrato) o como una fracción de muestreo. Usaremos la fracción de muestreo de 0.1. Para otras opciones, escriba ?stratified en la consola.
- replace: Esto permite elegir entre muestreo con o sin reemplazo. Lo hemos establecido como falso, lo que significa muestreo sin reemplazo.

Usaremos esta función para realizar un muestreo aleatorio estratificado.
También podemos realizar el muestreo estratificado utilizando nuestra función sample() estándar, siguiendo estos pasos:

1. Crear subconjuntos de datos por variables de estrato.
2. Calcular el tamaño de muestra para una fracción de muestreo de 0.1 para cada estrato.
3. Realizar un muestreo aleatorio simple de cada estrato para el tamaño de muestra calculado.

Los resultados anteriores y los de la función stratified() serán los mismos. Sin embargo, la función stratified() se ejecutará más rápido. Te recomendamos implementar este algoritmo y probar las demás funciones.

```
> set.seed(937)
> #Queremos asegurarnos de que nuestro muestreo conserve la misma proporción
>#  del tipo de tarjeta en la muestra.
```

```
> #Selecciona una muestra aleatoria sin reemplazo de cada grupo inicial que
> # consista en el 10% del tamaño total del estrato.
> install.packages("splitstackshape")
> library(splitstackshape)
> stratified_sample_10_percent<-stratified(data, group=c("CardType","State"),
+   size=0.1,replace=FALSE)
```

Paso 3: Comprueba si las proporciones de los puntos de datos en la muestra coinciden con la población.

Aquí se muestra el resultado de la muestra estratificada por tipo de tarjeta, estado y tabulación cruzada. Los valores muestran que el muestreo se realizó en todo el estrato con la misma proporción. Por ejemplo, si el número de registros de Alabama y American Express es 4980, en la muestra estratificada, el número de titulares de tarjetas de Alabama y American Express es 1/10 de la población, es decir, 498. Para todos los demás estratos, la proporción es la misma.

```
> #Frequency table for CardType in sample
> table(stratified_sample_10_percent$CardType)

American Express         Discover       MasterCard             Visa
          247483            64250           404268           283988
> #Frequency table for State in sample
> table(stratified_sample_10_percent$State)

   Alabama American Samoa          Arizona         Arkansas       California
      2013           16257            10174            20278           121606
   Georgia             Guam           Hawaii            Idaho         Illinois
     60862           30399             5044            11177             6099
Louisiana            Maine         Maryland    Massachusetts         Michigan
     15172           20191            20245             4081            30455
  Nebraska           Nevada    New Hampshire       New Jersey       New Mexico
      6061           30383             2022             4056            28442
  Oklahoma           Oregon     Pennsylvania     Rhode Island   South Carolina
     12219           12184            40589             3023            15225
   Vermont         Virginia       Washington    West Virginia        Wisconsin
     25281            2002            20297            18255             6138
      Colorado      Connecticut         Delaware          Florida
         17177            12180             2060             3032
       Indiana             Iowa           Kansas         Kentucky
         40471            20315             9113            14218
     Minnesota      Mississippi         Missouri          Montana
         18220            20305            10183             3013
  New York North Carolina   North Dakota             Ohio
          8133             9132            60856            36453
  South Dakota        Tennessee            Texas             Utah
          2046            20382            81264             9137
       Wyoming
          2069

> #Cross table frequency for sample data
> table(stratified_sample_10_percent$State,stratified_sample_10_percent$CardType)
```

Nota: Por brevedad se mostrará sólo una parte de la tabla.

Puedes observar que la proporción se ha mantenido igual. Aquí, comparamos las propiedades de la muestra y la población. La función summarise() muestra el promedio del saldo pendiente por estrato. Puede srealizar una prueba t por pares para comprobar que el muestreo no ha alterado las medias del saldo pendiente de cada estrato. Se recomienda realizar pruebas de medias mediante t.test(), como se muestra en la sección de muestreo aleatorio simple.

	American Express	Discover	MasterCard	Visa
Alabama	498	135	807	573
American Samoa	4014	1060	6574	4609
Arizona	2501	647	4111	2915
Arkansas	5016	1298	8204	5760
California	30118	7815	49119	34554
Colorado	4233	1119	6931	4894
Connecticut	3026	794	4926	3434
Delaware	499	132	843	586
Florida	761	191	1214	866
Georgia	15046	3900	24628	17288
Guam	7492	1943	12301	8663
Hawaii	1244	333	2037	1430
Idaho	2776	740	4497	3164
Illinois	1515	391	2463	1730
Indiana	10027	2609	16389	11446
Iowa	5018	1283	8199	5815
Kansas	2233	590	3708	2582
Kentucky	3538	908	5731	4041
Louisiana	3766	974	6125	4307

```
> # Average outstanding balance by stratum variables
> summary_population<-summarise(group_by(data,CardType,State),
+   OutstandingBalance_Stratum=mean(OutsBal))
`summarise()` has grouped output by 'CardType'. You can override using the `.groups` argument.
> #We can see below the want to make sure that our sampling retain the same proportion of the cardtype in the sample
> summary_sample<-summarise(group_by(stratified_sample_10_percent,CardType,
+   State),OutstandingBalance_Sample=mean(OutsBal))
`summarise()` has grouped output by 'CardType'. You can override using the `.groups` argument.
> #Mean Comparison by stratum
> summary_mean_compare<-merge(summary_population,summary_sample,
+   by=c("CardType","State"))
```

Nuevamente, realizaremos una prueba KS para comparar la distribución de la muestra estratificada. Podemos observar que la prueba KS muestra que ambas tienen la misma distribución.

```
> ks.test(data$OutsBal,stratified_sample_10_percent$OutsBal,alternative="two.sided")

    Asymptotic two-sample Kolmogorov-Smirnov test

data:  data$OutsBal and stratified_sample_10_percent$OutsBal
D = 0.00062693, p-value = 0.8672
alternative hypothesis: two-sided
```

La figura 12 muestra los histogramas que muestran la distribución del saldo pendiente para la muestra y la población. La comparación visual muestra claramente que la muestra es representativa de la población.

```
> par(mfrow =c(1,2))
> hist(data$OutsBal, breaks=50, col="red", xlab="Outstanding Balance",
+   main="Population ")
> hist(stratified_sample_10_percent$OutsBal, breaks=50, col="green",
+   xlab="Outstanding Balance", main="Stratified Sample")
```

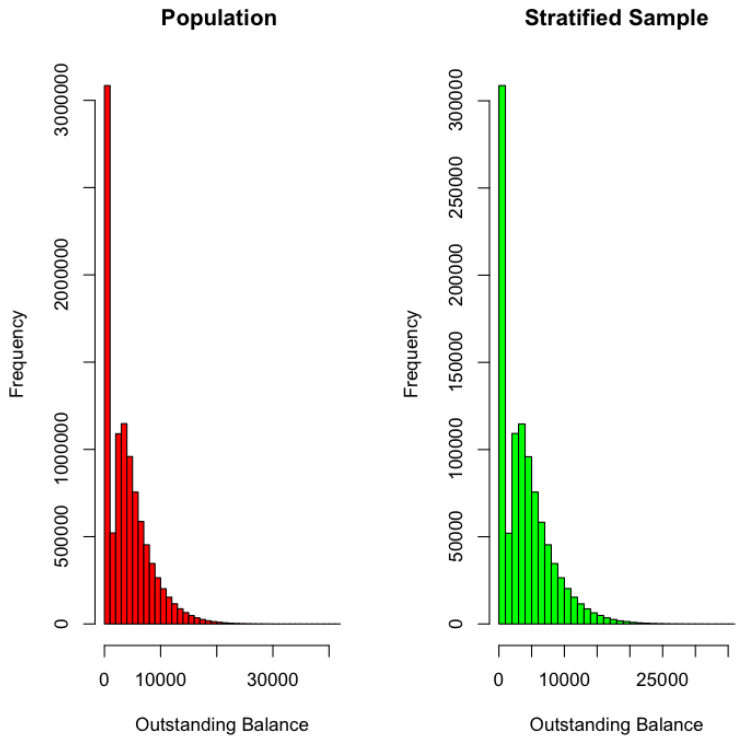

Figura 12: Distribución de la población y la muestra estratificada.

El gráfico de distribución de la figura 12 enfatiza los resultados de la prueba: tanto la población como la muestra aleatoria estratificada tienen la misma distribución. La muestra aleatoria estratificada es representativa de la población.

Puntos clave:

- El muestreo estratificado debe utilizarse cuando se desea asegurar que la proporción de puntos de datos se mantenga constante en la muestra. Esto no solo garantiza la representatividad, sino que también garantiza que todos los estratos tengan representación en la muestra.
- El muestreo estratificado también puede ayudarte a diseñar sistemáticamente la proporción de registros de cada estrato, lo que te permite diseñar un plan de muestreo estratificado para modificar la representación según las necesidades de la empresa. Por ejemplo, si estás modelando una función de respuesta binomial, y la tasa par o la proporción de 1 es muy pequeña en el conjunto de datos, Puedes realizar un muestreo

aleatorio estratificado del estrato (respuesta 0 o 1) e intentar muestrear de forma que la proporción de 1 aumente para facilitar el modelado.

Muestreo por conglomerados (cluster)

Muchas veces, las poblaciones contienen grupos heterogéneos que son estadísticamente evidentes. En esos casos, es importante identificar primero los grupos heterogéneos y luego planificar la estrategia de muestreo. Esta técnica es popular entre los diseñadores de marketing y campañas, ya que abordan las características de los grupos heterogéneos dentro de una población.

El muestreo por cluster puede realizarse de dos maneras:

- Muestreo monoetápico (*Single-stage sampling*): Todos los elementos dentro de los conglomerados seleccionados se incluyen en la muestra. Por ejemplo, si se desea estudiar una característica poblacional particular que predomina en un conglomerado, se podría identificar primero el conglomerado y su elemento y luego tomar todas las unidades de ese conglomerado.

- Muestreo bietápico (*Two-stage sampling*): Se selecciona aleatoriamente un subconjunto de elementos dentro de los conglomerados seleccionados para su inclusión en la muestra. Este método es similar al muestreo estratificado, pero difiere en que aquí los conglomerados son unidades madre, mientras que en el caso anterior eran estratos. Las variables de los estratos pueden dividirse en múltiples conglomerados en la escala de medida.

Para un tamaño de muestra fijo, el muestreo por conglomerados ofrece mejores resultados cuando la mayor parte de la variación en la población se concentra dentro de los grupos, no entre ellos. No siempre es sencillo elegir los métodos de muestreo. Muchas veces, el costo por punto de muestra es menor con el muestreo por conglomerados que con otros métodos de muestreo. Con este tipo de limitaciones de costo, el muestreo por conglomerados puede ser una buena opción.

Es importante destacar la diferencia entre estratos y conglomerados. Si bien ambos son subconjuntos superpuestos de la población, difieren en muchos aspectos.

- Si bien todos los estratos están representados en la muestra, en el conglomerado solo un subconjunto de conglomerados está en la muestra.
- El muestreo estratificado ofrece mejores resultados cuando las unidades dentro de los estratos son internamente homogéneas. Sin embargo, con el muestreo por

conglomerados, los mejores resultados se obtienen cuando los elementos dentro de los conglomerados son internamente heterogéneos.

Ventajas:

- Es más económico que otros métodos de recopilación de datos, ya que el conglomerado de interés requiere menos costos de recopilación y almacenamiento, así como menos costos administrativos.
- El conglomerado considera una población grande en términos de fragmentos de conglomerados. Dado el gran tamaño de estos grupos/conglomerados, implementar cualquier otra técnica resultaría muy difícil. La agrupación solo es viable cuando se trata de poblaciones grandes con conglomerados estadísticamente significativos.
- Se observa una reducción en la variabilidad de las estimaciones con otros métodos de muestreo, pero esta puede no ser la situación ideal en todos los casos.

Desventajas:

- El error de muestreo es elevado debido al diseño del proceso de muestreo. La razón entre el número de sujetos en el estudio de conglomerados y el número de sujetos en un estudio no conglomerado, muestreado aleatoriamente y con la misma fiabilidad se denomina *efecto de diseño*, lo que causa el elevado error de muestreo.
- Sesgo de muestreo: La muestra elegida en el muestreo por conglomerados se considerará representativa de toda la población y, si ese conglomerado tiene una opinión sesgada, se infiere que toda la población comparte la misma opinión. Esto podría no ser así.

Antes de mostrar el muestreo por conglomerados, crearemos conglomerados artificialmente en nuestros datos mediante la subdivisión de los datos por transacción internacional. Subdividiremos los datos con una declaración condicional sobre la transacción internacional. Aquí puede ver que estamos creando cinco clústeres artificialmente.

```
> # Antes de explicar el muestreo por conglomerados, intentemos subdividir los datos
> # de forma que tengamos muestras claras para explicar su importancia.
> # Subdividir los datos en 5 subgrupos.
> Data_Subset_Clusters_1 <-subset(data, IntTransc >2&IntTransc <5)
> Data_Subset_Clusters_2 <-subset(data, IntTransc >10&IntTransc <13)
> Data_Subset_Clusters_3 <-subset(data, IntTransc >18&IntTransc <21)
> Data_Subset_Clusters_4 <-subset(data, IntTransc >26&IntTransc <29)
> Data_Subset_Clusters_5 <-subset(data, IntTransc >34)
>Data_Subset_Clusters<-rbind(Data_Subset_Clusters_1,Data_Subset_Clusters_2,
+   Data_Subset_Clusters_3, Data_Subset_Clusters_4,Data_Subset_Clusters_5)
```

```
> str(Data_Subset_Clusters)
```

```
Classes 'data.table' and 'data.frame':1291631 obs. of  14 variables:
 $ creditLine : int  1 1 1 1 1 1 1 1 1 1 ...
 $ gender     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ state      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ CustomerID : int  136032 726293 1916600 2180307 3186929 3349887 3726743 5121051 7595816 8058527 ...
 $ NumOfCards : int  1 1 1 1 1 1 1 1 2 1 ...
 $ OutsBal    : int  2000 2000 2000 2000 2000 2000 0 0 2000 2000 ...
 $ DomesTransc: int  78 5 5 44 43 51 57 23 5 15 ...
 $ IntTransc  : int  3 4 3 3 4 4 3 3 4 3 ...
 $ FraudFlag  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ State      : chr  "Alabama" "Alabama" "Alabama" "Alabama" ...
 $ PostalCode : chr  "AL" "AL" "AL" "AL" ...
 $ Gender     : chr  "Male" "Male" "Male" "Male" ...
 $ CardType   : chr  "American Express" "American Express" "American Express" "American Express" ...
 $ CardName   : chr  "SimplyCash\256 Business Card from American Express" "SimplyCash\256 Business Card from American Express" "SimplyCash\256
Business Card from American Express" "SimplyCash\256 Business Card from American Express" ...
 - attr(*, ".internal.selfref")=<externalptr>
```

Creamos clústeres explícitamente basados en las transacciones internacionales. Los clústeres se crean para mostrar el muestreo por conglomerados.

El muestreo por conglomerados en una etapa implica seleccionar aleatoriamente conglomerados de entre cinco conglomerados para su análisis. Mientras que el muestreo en dos etapas implica seleccionar aleatoriamente algunos conglomerados y luego realizar un muestreo aleatorio estratificado a partir de ellos. En la figura 13, primero crearemos conglomerados utilizando k-medias (discutido en temas anteriores) y luego aplicaremos el muestreo estratificado, asumiendo que el conglomerado es la variable de estrato.

La función k-medias crea conglomerados basándose en el método de conglomerados k-medias basado en centroides. Dado que hemos creado explícitamente cinco conglomerados, llamaremos a k-medias para formar cinco conglomerados basados en los valores de las transacciones internacionales. Ya sabemos que la función nos dará exactamente cinco conglomerados, como los que creamos en el paso anterior. Esto se ha hecho solo con fines ilustrativos; en situaciones reales, es necesario determinar los conglomerados presentes en los datos de población.

```
> # Now we will treat the Data_Subset_Clusters as our population
> library(stats)
> kmeans_clusters <-kmeans(Data_Subset_Clusters$IntTransc, 5, nstart=25)
> cat("The cluster center are ",kmeans_clusters$centers)
The cluster center are  42.44327 3.455786 58.14917 30.23833 13.72113
```

A continuación, tomamos una muestra aleatoria de registros para representarlos gráficamente, ya que representarlos con un gran número de registros no será claro.

```
> set.seed(937)
> # For plotting lets use only 100000 records randomly chosen from total data.
> library(splitstackshape)
> PlotSample<-Data_Subset_Clusters[sample(nrow(Data_Subset_Clusters),
```

```
+   size=100000, replace = TRUE, prob = NULL),]
> plot(PlotSample$IntTransc, col = kmeans_clusters$cluster)
> points(kmeans_clusters$centers, col=1:5, pch=8)
> cluster_sample_combined<-cbind(Data_Subset_Clusters,kmeans_clusters$cluster)
> setnames(cluster_sample_combined,"V2","ClusterIdentifier")
```

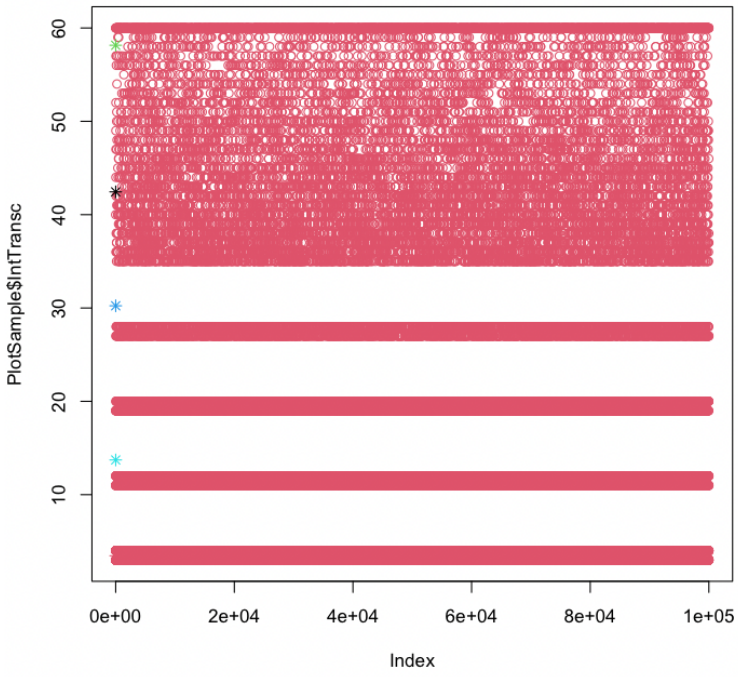

Figura 13: Datos de entrada segmentados por el conjunto de cinco clases según el número de transacciones internacionales.

Ahora, te mostramos el número de registros resumidos por cada conglomerado. Toma nota de estos números, ya que te mostraremos el muestreo por conglomerados en dos etapas. La muestra tendrá la misma proporción entre los conglomerados.

```
> print("Summary of no. of records per clusters")
[1] "Summary of no. of records per clusters"
> table(cluster_sample_combined$ClusterIdentifier)
     1      2.      3      4      5
 88512 751932  78493  62402 310292
```

Se asume el identificador del conglomerado como la variable de estrato y se utiliza la función stratified() para extraer una muestra que representa el 10% de la población del estrato, respectivamente.

```
> set.seed(937)
> cluster_sample_10_percent<-stratified(cluster_sample_combined,group=c("ClusterIdentifier"),
+   size=0.1,replace=FALSE)
```

Este paso ha creado la muestra por conglomerados en dos etapas; es decir, se ha seleccionado aleatoriamente el 10% de los registros de cada conglomerado. Grafiquemos los conglomerados con sus centros.

```
> print("Plotting the clusters for random sample from clusters")
[1] "Plotting the clusters for random sample from clusters"
> plot(cluster_sample_10_percent$IntTransc, col = kmeans_clusters$cluster)
> points(kmeans_clusters$centers, col =1:5, pch =8)
```

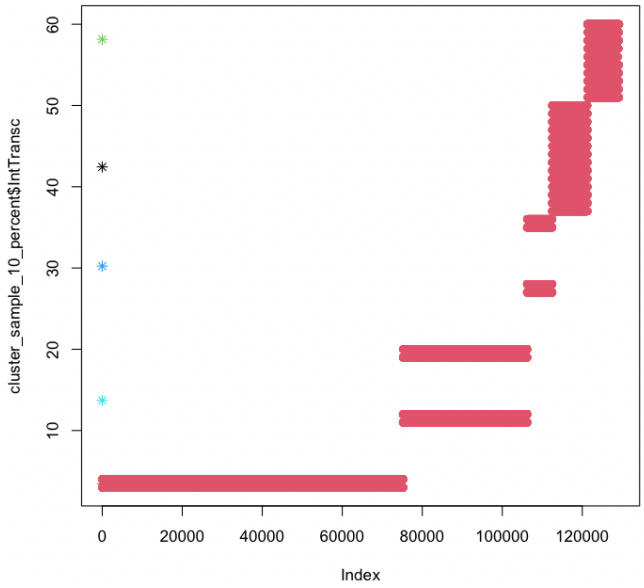

Figura 14: Conglomerados formados por K-medias (el asterisco representa el centroide del conglomerado).

A continuación, se muestra la distribución de frecuencias de la muestra por conglomerados. Por favor, revisa las mismas proporciones que en la población utilizada para el conglomerado. El muestreo estratificado en la segunda etapa del muestreo por conglomerados ha garantizado que las proporciones de los puntos de datos se mantengan iguales, es decir, el 10% del tamaño del estrato.

```
> print("Summary of no. of records per clusters")
[1] "Summary of no. of records per clusters"
> table(cluster_sample_10_percent$ClusterIdentifier)
   1       2       3       4       5
8851  75193  7849  6240  31029
```

Ahora, mostremos cómo el muestreo por conglomerados ha impactado la distribución del saldo pendiente en comparación con las muestras poblacionales y por conglomerados.

```
> population_summary <-summarise(group_by(data,CardType),
+   OutstandingBalance_Population=mean(OutsBal))
> cluster_summary<-summarise(group_by(cluster_sample_10_percent,CardType),
+   OutstandingBalance_Cluster=mean(OutsBal))
> summary_mean_compare<-merge(population_summary,cluster_summary, by="CardType")
> print(summary_mean_compare)
```

	CardType	OutstandingBalance_Population	OutstandingBalance_Cluster
1	American Express	3820.896	3816.467
2	Discover	4962.420	4931.525
3	MasterCard	3818.300	3826.441
4	Visa	4584.042	4620.322

Este resumen muestra cómo la media del saldo pendiente se vio afectada por el muestreo por conglomerados basado en las transacciones internacionales. Para una inspección visual, crearemos histogramas en la figura 15. Verás que la distribución se ve afectada marginalmente. Esto podría deberse a que los conglomerados que creamos, asumiendo que los segmentos de transacciones internacionales eran homogéneos y, por lo tanto, no tuvieron un gran impacto en el saldo pendiente. Para mayor seguridad, se recomienda realizar una t.test() para comprobar si las medias son significativamente iguales.

```
> par(mfrow =c(1,2))
> hist(data$OutsBal, breaks=50, col="red", xlab="Outstanding Balance",
+   main="Cluster Population")
> hist(cluster_sample_10_percent$OutsBal, breaks=50, col="green",
+   xlab="Outstanding Balance", main="Cluster Random Sample")
```

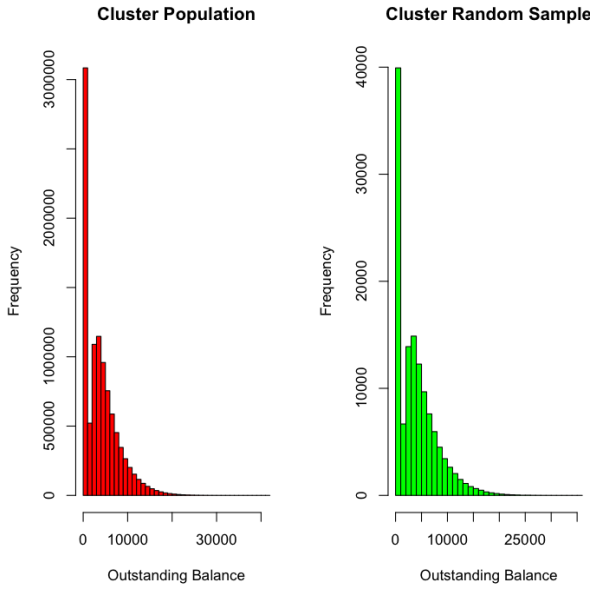

Figura 15: Distribución de la población por conglomerados y la muestra aleatoria por conglomerados.

En otras palabras, el muestreo por conglomerados es igual que el muestreo estratificado; la única diferencia es que la variable startum existe en los datos y es una propiedad intrínseca de los mismos, mientras que en el conglomerado primero identificamos los conglomerados y luego realizamos un muestreo aleatorio de ellos.

Puntos clave:

- El muestreo por conglomerados debe realizarse solo cuando exista evidencia sólida de conglomerados en la población y existan razones comerciales sólidas que justifiquen los conglomerados y su impacto en el resultado del modelo.
- El muestreo por conglomerados no debe confundirse con el muestreo estratificado. En el muestreo estratificado, los estratos se forman a partir de los atributos del conjunto de datos, mientras que los conglomerados se crean en función de la similitud de los sujetos en la población mediante alguna relación, por ejemplo, la distancia al centroide, las mismas características multivariantes, etc. Presta mucha atención al implementar el muestreo por conglomerados; los conglomerados deben existir y deben justificar su homogeneidad.

Muestreo Bootstrap

En estadística, el bootstrap es cualquier método, prueba o medida de muestreo que se basa en un muestreo aleatorio con reemplazo. En teoría, con el bootstrap se puede crear una población de tamaño infinito para muestrear. Es un tema avanzado en estadística y se utiliza ampliamente cuando se deben calcular medidas de muestreo, como la media, la varianza, el sesgo, etc., a partir de una estimación muestral.

El bootstrapping permite estimar la distribución muestral de casi cualquier estadística mediante métodos de muestreo aleatorio. El método Jackknife es anterior a la técnica moderna de bootstrapping. El estimador Jackknife de un parámetro se obtiene omitiendo repetidamente una observación y calculando la estimación. Una vez agotados todos los puntos de observación, se toma como estimador el promedio de las estimaciones. Para un tamaño de muestra de N, la estimación Jackknife también puede obtenerse agregando las estimaciones de cada N-1 estimación de la muestra. Es importante comprender el enfoque Jackknife, ya que proporciona la idea básica del método bootstrapping para la estimación de una métrica muestral.

La estimación Jackknife de un parámetro puede obtenerse estimando el parámetro para cada submuestra y omitiendo la i-ésima observación para estimar el valor previamente desconocido de un parámetro (por ejemplo, $\bar{x}_i$).

$$\bar{x}_i = \frac{1}{n-1}\sum_{j\neq i}^{n} x_j$$

La técnica jackknife se puede utilizar para estimar la varianza de un estimador.

$$\mathrm{Var}_{(\text{jackknife})} = \frac{n-1}{n}\sum_{i=2}^{n}\left(\bar{x}_i - \bar{x}_{(.)}\right)^2$$

Donde $\bar{x}_i$ es la estimación del parámetro obtenida al omitir la i-ésima observación, y $\bar{x}_{(.)}$ es el estimador basado en todas las submuestras.

En 1977, B. Efron, de la Universidad de Stanford, publicó su reconocido artículo "Métodos Bootstrap: Otra Mirada a Jackknife". Este artículo ofrece la primera explicación detallada del bootstrap para diversos problemas de estimación de métricas muestrales. Estadísticamente, el artículo intentó abordar el siguiente problema: dada una muestra aleatoria, X = (x1, x2,...Xn) de una distribución de probabilidad desconocida F, estimar la distribución muestral de una variable aleatoria preespecificada R(X, F), con base en los datos observados x. Dejamos a tu criterio la exploración de los detalles estadísticos del método.

Cuando se desconoce la distribución de la población (o ni siquiera se tiene una población), el bootstrap resulta útil para crear pruebas de hipótesis para las estimaciones muestrales. La técnica de bootstrap toma datos de la distribución empírica obtenida de la muestra. Si se asume que un conjunto de observaciones proviene de una población independiente e idénticamente distribuida, esto puede implementarse mediante la construcción de varias remuestras con reemplazo del conjunto de datos observados (y de igual tamaño). Esto resulta muy útil cuando se dispone de un conjunto de datos pequeño y se duda de la distribución del estimador para realizar pruebas de hipótesis.

Ventajas:

- Fácil de implementar; proporciona una manera sencilla de calcular errores estándar e intervalos de confianza para distribuciones muestrales complejas e desconocidas.
- Con el aumento de la potencia de cálculo, los resultados del bootstrap mejoran.
- Una aplicación popular del bootstrap es comprobar la estabilidad de las estimaciones.

Desventajas:

- El bootstrap es asintóticamente consistente, pero no proporciona consistencia para muestras finitas.
- Se trata de una técnica avanzada, por lo que es necesario conocer plenamente los supuestos y las propiedades de las estimaciones derivadas de los métodos bootstrap.

En nuestro ejemplo de R, mostraremos cómo usar el bootstrap para estimar un parámetro poblacional y crear un intervalo de confianza en torno a dicha estimación. Esto ayuda a comprobar la estabilidad de la estimación del parámetro y a realizar una prueba de hipótesis. Crearemos el ejemplo con una metodología de regresión lineal relevante para el negocio.

Nota. Las técnicas de bootstrap son más relevantes para los problemas de estimación cuando se cuenta con un tamaño de muestra muy pequeño y resulta difícil determinar la distribución de la población real.

Primero, ajustamos un modelo de regresión lineal a los datos poblacionales (sin intersección). El modelo se ajustará con la variable de respuesta como variable pendiente y el predictor como el número de transacciones nacionales. La intuición empresarial indica que el saldo pendiente debe tener una correlación positiva con el número de transacciones nacionales. Una correlación positiva entre las variables dependientes e independientes implica que el signo del coeficiente de regresión lineal debe ser positivo.

El coeficiente que obtenemos es el valor real de la estimación proveniente de la población. Esta es la estimación del parámetro poblacional, ya que calcula la población completa.

```
> set.seed(937)
> library(boot)
> # Now we need the function we would like to estimate
> #First fit a linear model and know the true estimates, from population data
> summary(lm(OutsBal ~0 +DomesTransc, data = data))

Call:
lm(formula = OutsBal ~ 0 + DomesTransc, data = data)

Residuals:
   Min     1Q Median     3Q    Max
 -7713  -1080   1449   4430  39091

Coefficients:
           Estimate Std. Error t value Pr(>|t|)
DomesTransc 77.13469    0.03919    1968   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4867 on 9999999 degrees of freedom
Multiple R-squared:  0.2792,  Adjusted R-squared:  0.2792
F-statistic: 3.874e+06 on 1 and 9999999 DF,  p-value: < 2.2e-16
```

Puedes ver el resumen del ajuste del modelo de regresión lineal a los datos poblacionales. Ahora tomaremos una pequeña muestra de la población (fracción de muestreo =

10000/10000000 = 1/1000). Por lo tanto, nuestro reto es estimar el coeficiente de transacciones nacionales a partir de un conjunto de datos muy pequeño.

En este contexto, el muestreo puede entenderse como un proceso para crear un conjunto más amplio de muestras a partir de un conjunto pequeño de valores y obtener una estimación de la distribución real de dicha estimación.

```
> set.seed(937)
> # Supongamos que solo contamos con 10 000 puntos de datos y debemos realizar la prueba de
> # hipótesis sobre la significancia del coeficiente de las transacciones nacionales.
> # Dado que el conjunto de datos es pequeño, utilizaremos el método de arranque para crear
> # la distribución del coeficiente y, a continuación, crearemos un intervalo de confianza para probar
la hipótesis.
> sample_10000 <-data[sample(nrow(data),size=10000, replace=FALSE, prob=NULL),]
```

Ahora contamos con una muestra pequeña con la que trabajar. Definamos una función llamada Coeff, que devolverá el coeficiente de la variable de transacciones nacionales.

Tiene tres argumentos:

- data: Este será el pequeño conjunto de datos que se desea arrancar. En nuestro caso, se trata del conjunto de datos de muestra de 10000 registros.
- b: Un frame aleatorio de índices que se seleccionará cada vez que se llame a la función. Esto garantizará que cada vez que se seleccione un conjunto de datos de un modelo, este se elija aleatoriamente de los datos de entrada.
- formula: Este es un campo opcional. Sin embargo, esta será la forma funcional del modelo que se estimará mediante la regresión lineal.

Aquí acabamos de incorporar la fórmula en la declaración de retorno.

```
> # Function to return Coefficient of DomesTransc
> Coeff =function(data, b, formula){
+   # b is the random indexes for the bootstrap sample
+   d =data[b,]
+   return(lm(OutsBal ~0 +DomesTransc, data = d)$coefficients[1])
+   # thats for the beta coefficient
+ }
```

Ahora podemos iniciar el arranque, por lo que utilizaremos la función boot() de la biblioteca boot. Es una función muy potente para el arranque tanto paramétrico como no paramétrico. Consideramos este un tema avanzado y no cubriremos los detalles de esta función. Se recomienda a los lectores interesados consultar la documentación de la función de CRAN.
Las entradas que utilizamos para nuestro ejemplo son:

- data: Estos son los pequeños datos de muestra que creamos en el paso anterior.
- statistics: Esta función devuelve el valor estimado del parámetro de interés. En este caso, nuestra función Coeff devuelve el valor del coeficiente de las transacciones nacionales.
- R: Este es el número de muestras bootstrap que deseas crear. Como regla general, cuantas más muestras bootstrap tengas, más estrecha será la banda de confianza.

Nota. En este ejemplo, consideramos un número menor de muestras para asegurarnos de que la banda de confianza sea amplia y con qué seguridad podemos ver la estimación original de la población.

Aquí llamamos a la función con R=50.

```
> set.seed(937)
> # R is how many bootstrap samples
> bootbet =boot(data=sample_10000, statistic=Coeff, R=50)
> names(bootbet)
 [1] "t0"      "t"       "R"       "data"    "seed"    "statistic" "sim"     "call"    "stype"   "strata"
"weights"
```

Ahora, grafiquemos los histogramas y los gráficos qq para los valores estimados del coeficiente.

```
> plot(bootbet)
```

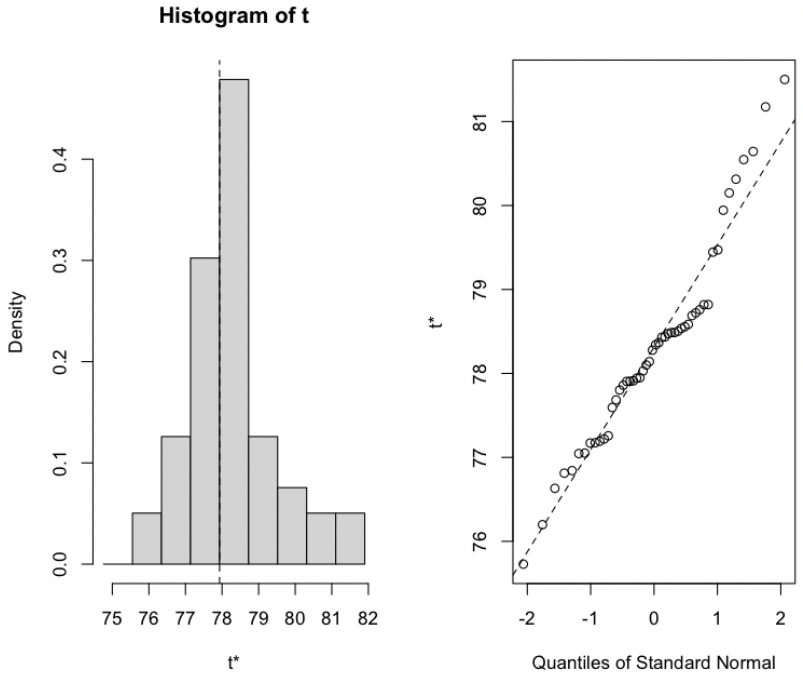

Figura 16: Histograma y gráfico qq del coeficiente estimado.

Ahora, grafiquemos el histograma de la estimación del parámetro. Podemos observar que la muestra bootstrap reveló la distribución del parámetro. Podemos formar un intervalo de confianza en torno a esto y realizar pruebas de hipótesis.

```
> hist(bootbet$t, breaks=5)
```

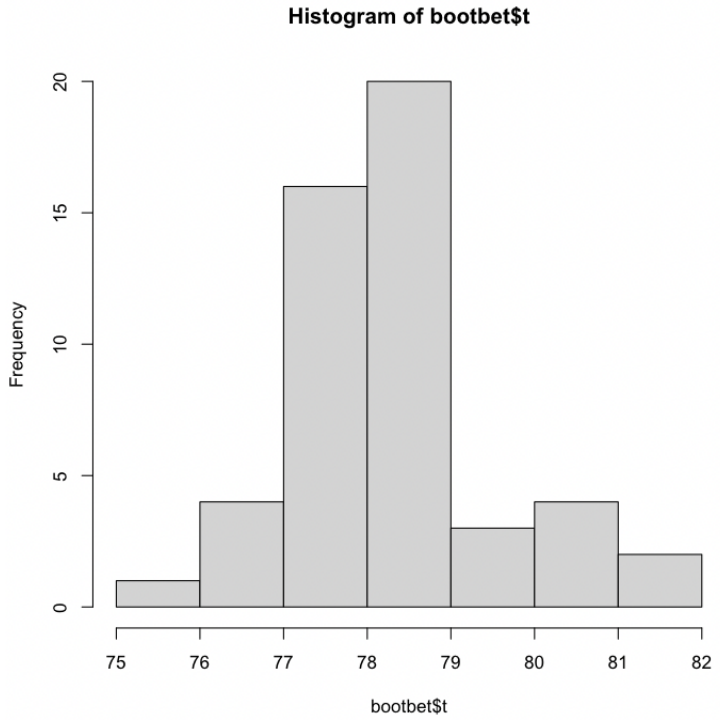

Figura 17: Histograma de la estimación del parámetro a partir del bootstrap.

Aquí calculamos la media y la varianza de los valores estimados mediante el método bootstrap. Considerando que la distribución del coeficiente es normal, se puede crear una banda de confianza alrededor de la media para el valor real.

```
> mean(bootbet$t)
[1] 78.31239
> var(bootbet$t)
         [,1]
[1,] 1.486702
```

Además, para mostrar cómo se ve la distribución superpuesta a una distribución normal a partir de los parámetros anteriores, haz lo siguiente:

```
> x <-bootbet$t
> h<-hist(x, breaks=5, col="red", xlab="Boot Strap Estimates",
+   main="Histogram with Normal Curve")
> xfit<-seq(min(x),max(x),length=40)
```

```
> yfit<-dnorm(xfit,mean=mean(bootbet$t),sd=sqrt(var(bootbet$t)))
> yfit <-yfit*diff(h$mids[1:2])*length(x)
> lines(xfit, yfit, col="blue", lwd=2)
```

En la figura 18 se puede ver que hemos podido encontrar la distribución del coeficiente y, por lo tanto, podemos realizar pruebas de hipótesis sobre ella. Esto también nos proporcionó una estimación aproximada del coeficiente real. Si se observa con atención, esta idea es muy similar a la propuesta original de Jackknife. Con mayor capacidad de cálculo, hemos ampliado el alcance de ese método desde la media y la desviación estándar a cualquier estimación de parámetros.

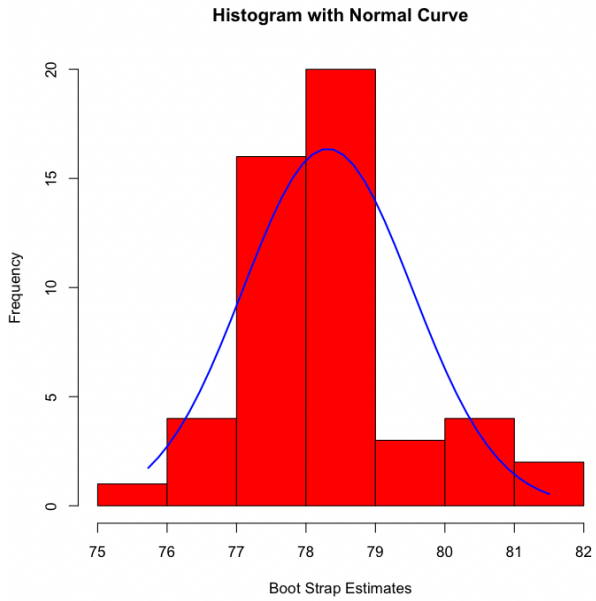

Figura 18: Histograma con función de densidad normal.

El siguiente código realiza una prueba t.test() sobre los valores bootstrap de los coeficientes con la estimación real del coeficiente a partir de los datos poblacionales. Esto nos indicará cuán cerca estamos de la estimación con una muestra más pequeña y con qué confianza podemos aceptar o rechazar el coeficiente bootstrap.

```
> t.test(bootbet$t, mu=77.13)

    One Sample t-test

data:  bootbet$t
t = 6.857, df = 49, p-value = 1.105e-08
alternative hypothesis: true mean is not equal to 77.13
95 percent confidence interval:
 77.96587 78.65892
sample estimates:
mean of x
 78.31239
```

Puntos clave:

- El bootstrapping es una técnica potente que resulta útil cuando se tiene poco conocimiento de la distribución de parámetros y solo se dispone de un conjunto de datos pequeño.
- Esta técnica es avanzada e implica muchas suposiciones, por lo que se requieren conocimientos estadísticos adecuados para utilizarlas.

Método de Monte Carlo: Método de aceptación-rechazo

En la actualidad, los métodos de Monte Carlo se han convertido en un campo de estudio independiente en estadística. Los métodos de Monte Carlo aprovechan las técnicas de muestreo aleatorio, que requieren un alto volumen computacional, para estimar los parámetros subyacentes. Esta técnica es importante en ecuaciones estocásticas donde no es posible una solución exacta. Las técnicas de Monte Carlo son muy populares en el mundo financiero, específicamente en la valoración y previsión de instrumentos financieros.

En estadística, los métodos de aceptación-rechazo son técnicas muy básicas para muestrear observaciones de una distribución. En este método, el muestreo aleatorio se realiza a partir de una distribución y, según condiciones preestablecidas, la observación se acepta o rechaza, por lo que se encuentra en el amplio espectro del método de Monte Carlo.

En este método, primero estimamos la distribución empírica del conjunto de datos (función de densidad empírica r: ·EDF) observando la distribución de probabilidad acumulada. Tras obtener la EDF, establecemos los parámetros para otra distribución conocida. La distribución conocida cubrirá la EDF.

Ahora comenzamos a muestrear a partir de la distribución conocida y aceptamos las observaciones si se encuentran dentro de la FDE. De lo contrario, la rechazamos. En otras palabras, el muestreo de rechazo se puede realizar siguiendo estos pasos:

1. Muestrear un punto de la distribución propuesta (digamos x).
2. Dibujar una línea vertical en este punto de muestra x hasta la curva de la distribución propuesta (Figura 19).
3. Muestrear uniformemente a lo largo de esta línea desde 0 hasta el máximo (PDF). PDF significa función de densidad de probabilidad. Si el valor de una muestra es mayor que el valor máximo, rechazarla; de lo contrario, aceptarla.

Este método nos permite extraer una muestra de cualquier distribución a partir de la distribución conocida. Estos métodos son muy populares en el cálculo estocástico para la valoración de productos financieros y otros procesos estocásticos.

Para ilustrar este método, extraeremos una muestra de una distribución beta con parámetros de (3,10). La distribución beta se muestra en la figura 19.

```
> curve(dbeta(x, 3,10),0,1)
```

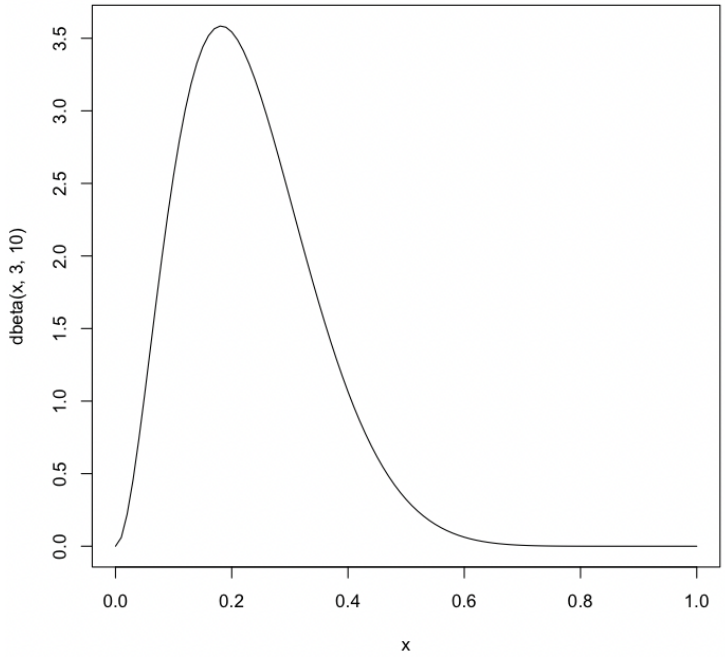

Figura 19: Gráfico de distribución beta.

Primero creamos una muestra de 5000 valores aleatorios entre 0 y 1. Ahora calculamos la densidad beta correspondiente a la muestra de 5000 valores aleatorios.

```
> set.seed(937)
> sampled <-data.frame(proposal =runif(5000,0,1))
> sampled$targetDensity <-dbeta(sampled$proposal, 3,10)
```

Ahora, calculamos la densidad de probabilidad máxima para nuestra distribución propuesta (beta PDF). Una vez que tengamos la densidad máxima y la densidad muestral para 5000 casos, comenzamos nuestro muestreo por rechazo de la siguiente manera. Creamos un número aleatorio entre 0 y 1:

```
> # Rechaza el valor como proveniente de la distribución beta si el valor es mayor
> # que la densidad de muestra que calculamos para la distribución beta conocida previamente.
> maxDens = max(sampled$targetDensity, na.rm = T)
```

```
> sampled$accepted = ifelse(runif(5000,0,1) < sampled$targetDensity / maxDens, TRUE, FALSE)
```

La figura 20 muestra un gráfico de la FDE de beta (3,10) y el histograma del conjunto de datos muestral. Podemos ver que hemos podido crear la muestra deseada al aceptar valores de números aleatorios que se encuentran por debajo de la línea roja, es decir, la PDF de la distribución beta.

```
> hist(sampled$proposal[sampled$accepted], freq=F, col="grey", breaks=100)
> curve(dbeta(x, 3,10),0,1, add =T, col ="red")
```

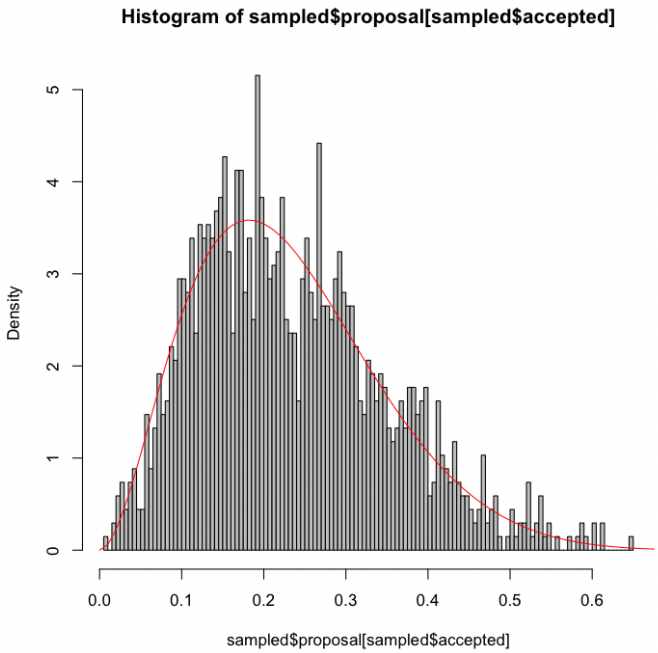

Figura 20: Muestreo por rechazo.

Una explicación cualitativa del ahorro computacional mediante el muestreo

Esta sección muestra un pequeño ejemplo para ayudarte a comprender cómo el muestreo también ayuda a reducir los costos computacionales. Para demostrarlo, primero ajustaremos un modelo de regresión lineal utilizando el conjunto de datos poblacionales y luego ajustaremos el mismo modelo con una muestra más pequeña.

Sabemos, por nuestra discusión sobre el muestreo, que si se realiza correctamente, podemos estimar los parámetros poblacionales con un alto nivel de confianza. A modo de ilustración, mostraremos el ajuste de una regresión lineal en una población de 10 millones de registros y una muestra de 10000.

A continuación, llamamos a la función sys.time(), que devuelve la hora actual del sistema. Con esta función, calculamos el tiempo de cálculo de la función para la población y la muestra.
Primero, ajustamos un modelo de regresión lineal con los datos de la población total.

```
> # estimate parameters
> library(MASS)
> start.time <-Sys.time()
> population_lm<-lm(OutsBal ~DomesTransc +Gender, data = data)
> end.time <-Sys.time()
> time.taken_1 <-end.time -start.time
> cat("Time taken to fit linear model on Population",time.taken_1 )
Time taken to fit linear model on Population 21.19568
```

Ahora, ajustemos el mismo modelo a una muestra aleatoria de 10000 valores.

```
> start.time <-Sys.time()
> sample_lm<-lm(OutsBal ~DomesTransc +Gender, data = sample_10000)
> end.time <-Sys.time()
> time.taken_2 <-end.time -start.time
> cat("Time taken to fit linear model on Sample ",time.taken_2 )
Time taken to fit linear model on Sample  14.41127
```

Podemos observar la diferencia en los tiempos de ambos cálculos. (Nota: Los tiempos mostrados se basan en la capacidad de cálculo de mi computadora Imac; es posible que obtengas tiempos diferentes según la configuración de tu sistema).

En esencia, la operación de población tardó mucho tiempo. La estimación con datos de población tardó más de 1000 veces más que la misma estimación con la muestra.

Resumen

En este documento, abordamos diferentes técnicas de muestreo y mostramos cómo estas reducen el volumen de datos a procesar y, al mismo tiempo, conservan sus propiedades. El mejor método de muestreo para cualquier población es el muestreo aleatorio simple sin reemplazo.

También analizamos el muestreo bootstrap, un concepto importante, ya que permite estimar la distribución de cualquier parámetro mediante este método. Finalmente, mostramos una ilustración del muestreo por rechazo, que permite crear cualquier distribución a partir de distribuciones conocidas. Esta técnica se basa en la simulación de Monte Carlo y es muy

popular en el sector financiero. Este documento desempeña un papel importante en la reducción del volumen de datos que se aplican a nuestros algoritmos de aprendizaje automático, manteniendo así intacta la varianza poblacional.

En un tema que no trataremos en el curso, se podrían analizar las propiedades de los datos con visualización. Si utilizamos el muestreo adecuado, la visualización y las tendencias aparecerán de la misma manera tanto en las poblaciones como en la muestra.