Análisis de encuestas con R

Andrés Gutiérrez[[1]](#footnote-20),Stalyn Guerrero[[2]](#footnote-22), Cristian Téllez[[3]](#footnote-24), Giovany Babativa[[4]](#footnote-26)

2025-09-29

Índice de figuras

Índice de cuadros

# Prefacio

La versión online de este libro está licenciada bajo una [Licencia Internacional de Creative Commons para compartir con atribución no comercial 4.0](http://creativecommons.org/licenses/by-nc-sa/4.0/).

Este libro es el resultado de un compendio de las experiencias internacionales prácticas adquiridas por el autor como Experto Regional en Estadísticas Sociales de la CEPAL.

# 1 Introducción

FALTA ESTO: BADEHOG en la Cepal

Las encuestas de hogares se han consolidado como una de las principales fuentes de información estadística para el diseño, la evaluación y el seguimiento de políticas públicas. Su relevancia se extiende más allá del ámbito nacional: constituyen un insumo indispensable para el monitoreo de los Objetivos de Desarrollo Sostenible (ODS) en el marco de la Agenda 2030. Gracias a la amplitud temática que abarcan —educación, salud, empleo, ingresos, condiciones de vida y pobreza, entre otros—, las encuestas de hogares permiten elaborar indicadores que reflejan la situación de las poblaciones y orientan la formulación de estrategias de desarrollo inclusivas y sostenibles.

Sin embargo, la utilidad de estas encuestas no depende únicamente de la calidad del operativo de campo. La producción de estadísticas oficiales exige un compromiso adicional: garantizar que los resultados sean precisos, confiables y comparables. En este sentido, las oficinas nacionales de estadística y los investigadores deben adoptar métodos robustos que respeten los Principios Fundamentales de las Estadísticas Oficiales de las Naciones Unidas, particularmente aquellos relacionados con la responsabilidad y transparencia (Principio 3) y la prevención del uso indebido de la información (Principio 4). Estos principios enfatizan la necesidad de documentar y difundir con claridad las fuentes, los métodos y los procedimientos utilizados, así como de pronunciarse cuando se detectan interpretaciones incorrectas de los datos.

En las últimas décadas, la revolución informática y el acceso cada vez mayor a software estadístico han multiplicado las posibilidades de análisis. Hoy en día, los datos de encuestas de hogares se utilizan tanto en aplicaciones descriptivas (como la estimación de medias poblacionales, tasas de pobreza o distribuciones de frecuencia) como en aplicaciones analíticas o inferenciales (pruebas de hipótesis, construcción de modelos de regresión, estimaciones de bienestar multidimensional). Sin embargo, este avance trae consigo un riesgo: los programas estadísticos tradicionales, cuando se emplean sin tener en cuenta la complejidad del diseño muestral, pueden producir estimaciones sesgadas e intervalos de confianza engañosos, induciendo a conclusiones erróneas que pueden repercutir en la toma de decisiones públicas.

El desafío central reside en que las encuestas de hogares rara vez utilizan un muestreo aleatorio simple. Por el contrario, se basan en diseños de muestreo complejos que incorporan procedimientos como:

* Estratificación, donde la población se divide en subgrupos homogéneos (por ejemplo, región, área urbana/rural o nivel socioeconómico), a partir de los cuales se seleccionan muestras independientes. Este procedimiento asegura representatividad y mejora la precisión de las estimaciones.
* Conglomeración, que consiste en agrupar las unidades de observación en unidades primarias de muestreo (manzanas, sectores censales, áreas de enumeración). En una primera etapa se seleccionan los conglomerados, y en etapas subsiguientes los hogares dentro de estos.
* Probabilidades desiguales de selección, diseñadas para garantizar la inclusión adecuada de subpoblaciones de interés, especialmente aquellas pequeñas o con alta heterogeneidad.
* Ajustes de ponderación, mediante los cuales los factores de expansión iniciales se corrigen para compensar problemas de cobertura, no respuesta o para calibrar los resultados frente a totales poblacionales conocidos.

Estas características convierten al análisis de encuestas en una tarea especializada, que requiere técnicas específicas para obtener estimaciones no sesgadas y una correcta cuantificación de la incertidumbre.

En este marco, el presente libro tiene como objetivo principal ofrecer al lector una guía práctica y accesible para el análisis de encuestas complejas, con un énfasis particular en las dinámicas de los hogares. Se propone como un material tanto de formación como de consulta, que combina la explicación conceptual con la aplicación práctica en el software estadístico R. La elección de R responde a tres razones fundamentales: (i) es un software de código abierto, disponible gratuitamente para cualquier usuario o institución; (ii) cuenta con un ecosistema robusto de paquetes especializados, entre ellos survey, que facilita el trabajo con diseños muestrales complejos; y (iii) es ampliamente adoptado por la comunidad académica y los institutos de estadística, lo que lo convierte en una herramienta idónea para la enseñanza, la replicación y la transparencia en los procesos analíticos.

El lector encontrará a lo largo del libro no solo los fundamentos teóricos, sino también ejemplos reproducibles, tablas y gráficos generados íntegramente en R, acompañados del código necesario para su replicación. De esta manera, se busca que cada usuario pueda adaptar los procedimientos a sus propios datos y necesidades.

La estructura del libro se organiza en capítulos progresivos:

* El Capítulo 2 introduce los conceptos esenciales sobre encuestas complejas y su importancia en el análisis de hogares.
* Los Capítulos 3 y 4 desarrollan la teoría de las variables aleatorias discretas y continuas en el contexto del muestreo probabilístico, sentando las bases para el análisis inferencial.
* El Capítulo 5 aborda la estimación de modelos de regresión lineal, mientras que el Capítulo 6 introduce el ajuste de modelos de regresión logística, ambos pilares del análisis aplicado en encuestas de hogares.
* Los Capítulos 7 y 8 amplían el enfoque hacia modelos lineales generalizados y multiniveles, necesarios para capturar relaciones más complejas y estructuras jerárquicas de los datos.
* El Capítulo 9 se centra en la imputación de datos, una herramienta indispensable para enfrentar el aumento de la no respuesta, fenómeno que se intensificó tras la pandemia.
* Finalmente, el Capítulo 10 explora las técnicas de visualización y comunicación de resultados, un aspecto crucial para garantizar que los hallazgos lleguen de manera clara y efectiva a audiencias técnicas y no técnicas.

Con esta estructura, el libro pretende dotar al lector de un conjunto integral de herramientas que abarcan desde los fundamentos conceptuales hasta las aplicaciones más avanzadas. Más allá de ser un manual técnico, busca contribuir a una cultura estadística orientada a la calidad, la transparencia y la utilidad de la información derivada de encuestas de hogares.

En síntesis, este libro se propone como una guía para investigadores, estudiantes y profesionales que deseen comprender y aplicar los métodos de análisis de encuestas complejas en R, y como un aporte al fortalecimiento de las capacidades estadísticas en la región. Al integrar teoría, práctica y principios internacionales, aspira a fomentar análisis más rigurosos y decisiones más informadas, contribuyendo así al avance del conocimiento y al logro de los objetivos de desarrollo en nuestras sociedades.

# 2 Conceptos básicos en encuestas de hogares

El análisis riguroso de encuestas de hogares parte de una comprensión clara de sus fundamentos conceptuales y metodológicos. Como señalan *Sarndal, Swensson & Wretman (1992)* y *Gutiérrez (2016)*, el punto de partida es la definición de la población objetivo, el universo de estudio y el marco muestral, que sirven de base para el diseño de la encuesta y la selección de la muestra. Estos elementos son esenciales para garantizar la validez de las inferencias y la representatividad de los resultados.

En este contexto, un aspecto crucial es la consideración del diseño muestral. Para obtener conclusiones válidas sobre la población, es necesario adoptar un enfoque de inferencia basada en el diseño, el cual reconoce que la muestra no surge de una selección aleatoria simple, sino de un plan probabilístico bien definido. En dicho plan, cada unidad de la población tiene una probabilidad conocida y distinta de cero de ser incluida en la muestra, lo que constituye la garantía fundamental de que los resultados puedan generalizarse a la población de referencia.

Bajo este enfoque, las estimaciones son insesgadas (o prácticamente insesgadas) en relación con el esquema de muestreo, sin depender de supuestos sobre la distribución de la variable de interés. Esto otorga a la inferencia basada en el diseño un carácter robusto y ampliamente aceptado en el análisis de encuestas de hogares.

Un elemento central de este proceso son los pesos de muestreo, que indican cuántas unidades de la población está representando cada unidad seleccionada. Dichos pesos permiten ajustar las estimaciones a las particularidades del diseño, de modo que los resultados reflejen con fidelidad la estructura poblacional. Además, un diseño bien documentado facilita el análisis estadístico, respalda la interpretación rigurosa de los datos y posibilita obtener conclusiones significativas sobre fenómenos complejos.

En contraste, ignorar el diseño muestral y aplicar métodos de análisis que asumen una muestra aleatoria simple puede llevar a estimaciones sesgadas, errores de inferencia y conclusiones equivocadas. Por ello, el análisis de encuestas de hogares no puede desligarse de su diseño de selección: este no es un detalle técnico accesorio, sino el núcleo mismo que sustenta la validez de la información producida.

## 2.1 Universo de estudio y población objetivo

El término encuesta se encuentra directamente relacionado con una población finita compuesta de individuos a los cuales es necesario entrevistar. El *universo de estudio* lo constituye el total de individuos o elementos que poseen dichas características a ser estudiadas. Ahora bien, conjunto de unidades de interés sobre los cuales se tendrán resultados recibe el nombre de *población objetivo*. Por ejemplo, *la Encuesta Nacional de Empleo y Desempleo* de Ecuador define su población objetivo como todas las personas mayores de 10 años residentes en viviendas particulares en Ecuador.

## 2.2 Unidades de análisis

Corresponden a los diferentes niveles de desagregación establecidos para consolidar el diseño probabilístico y sobre los que se presentan los resultados de interés. En México, la *Encuesta Nacional de Ingresos y Gastos de los Hogares* define como unidades de análisis el ámbito al que pertenece la vivienda, urbano alto, complemento urbano y rural. La *Gran Encuesta Integrada de Hogres* de Colombia tiene cobertura nacional y sus unidades de análisis están definidas por 13 grandes ciudades junto con sus áreas metropolitanas.

## 2.3 Unidades de muestreo

El diseño de una encuesta de hogares en América Latina plantea la necesidad de seleccionar en varias etapas ciertas *unidades de muestreo* que sirven como medio para seleccionar finalmente a los hogares que participarán de la muestra. La *Pesquisa Nacional por Amostra de Domicilios* en Brasil se realiza por medio de una muestra de viviendas en tres etapas, cada etapa se define como una unidad de muestreo. Por ejemplo, las unidades de muestreo en PNAD son:

* Las unidades primarias de muestreo (UPM) son los municipios,
* Las unidades secundarias de muestreo (USM) son los sectores censales, que conforman una malla territorial conformada en el último Censo Demográfico.
* Las últimas unidades en ser seleccionadas son las viviendas.

## 2.4 Marcos de muestreo

Para realizar el proceso de selección sistemática de los hogares es necesario contar con un marco de muestreo que sirva de *link* entre los hogares y las unidades de muestreo y que permita tener acceso a la población de interés. En este sentido, el *marco muestral* es el conjunto en el cual se identifican a todos los elementos que componen la población objeto de estudio, de la cual se selecciona la muestra. Los marcos de muestreo más utilizados en encuestas complejas son de áreas geográficas que vinculan directamente a los hogares o personas.

A modo de ejemplo, la *Encuesta Nacional de Hogares* de Costa Rica utiliza un marco muestral construido a partir de los censos nacionales de población y vivienda de 2011. Dicho marco corresponde a uno de áreas en donde sus unidades son superficies geográficas asociadas con las viviendas. Este marco permite la definición de UPM con 150 viviendas en las zonas urbanas y 100 viviendas en las zonas rurales. Este marco está conformado por 10461 UPM (64.5% urbanas y 35.5% rurales).

## 2.5 Motivación

Desde que se popularizaron las encuestas de hogares en 1940, se han hecho evidentes algunas tendencias vinculadas a los avances tecnológicos tanto en las agencias estadísticas como en la sociedad, las cuales se han acelerado con la introducción del computador. *Gambino & Silva (2009)*

El muestreo surge como respuesta a la necesidad de obtener información estadística precisa sobre una población objetivo, sin recurrir a un censo completo. Como señala *Gutiérrez (2016)*, el muestreo consiste en investigaciones parciales sobre la población que permiten inferir resultados al conjunto total. En las últimas décadas, esta metodología se ha consolidado en distintos campos —especialmente en el sector gubernamental, con la producción de estadísticas oficiales que facilitan el seguimiento de políticas públicas y de los Objetivos de Desarrollo Sostenible—, pero también en el ámbito académico, privado y de comunicaciones.

En el marco de este libro, el foco se centra en el análisis de encuestas de hogares. Para que el lector disponga de ejemplos prácticos, en este capítulo se empleará la base de datos **BigCity**, que contiene información socioeconómica de 150 266 personas de una ciudad en un año específico. Entre sus variables destacan:

* *HHID:* Identificador del hogar.
* *PersonID:* Identificador de la persona dentro del hogar.
* *Stratum:* Estrato geográfico (119 en total).
* *PSU:* Unidades primarias de muestreo (1664 en total).
* *Zone:* Zona urbana o rural.
* *Sex:* Sexo del entrevistado.
* *Income:* Ingreso mensual per cápita.
* *Expenditure:* Gasto mensual per cápita.
* *Employment:* Situación laboral.
* *Poverty:* Condición de pobreza según ingresos.

### 2.5.1 La importancia de considerar el diseño muestral

Al analizar datos de encuestas de hogares, ignorar el diseño de muestreo compromete la representatividad, la precisión y la credibilidad de los resultados, lo que puede conducir a decisiones erróneas. Korn y Graubard (1995) muestran cómo las estimaciones ponderadas y no ponderadas pueden diferir sustancialmente, lo que resalta la necesidad de aplicar siempre métodos consistentes con el diseño.

Como se mencionó, las encuestas de hogares se caracterizan por:

* Diseños de muestreo complejos (estratificación, conglomeración y probabilidades desiguales de selección), que buscan mejorar la eficiencia y la precisión.
* Pesos de muestreo para cada unidad, que permiten representar adecuadamente a la población.

#### 2.5.1.1 Ejemplo ilustrativo

Supóngase un país con dos regiones: la Región A (100 habitantes, ingreso promedio de $10 000) y la Región B (900 habitantes, ingreso promedio de $2 000). El ingreso promedio verdadero es:

Si se encuestan 50 personas en cada región y se ignora el diseño de muestreo, asignando el mismo peso a todas las observaciones, se obtiene:

El sesgo es evidente: se sobreestima el ingreso nacional, pues la Región A (10 % de la población) influye tanto como la Región B (90 %).

En cambio, si se aplican **pesos proporcionales al tamaño poblacional** (2 para la Región A y 18 para la Región B), la estimación corregida es:

Lo que reproduce el valor verdadero y elimina el sesgo.

#### 2.5.1.2 Conglomeración y precisión

Otra característica crítica es la **conglomeración**. La mayoría de encuestas selecciona unidades primarias de muestreo (UPM) como sectores censales o áreas de enumeración, y dentro de ellas, submuestras de hogares. Este diseño reduce costos, pero afecta la precisión: si los hogares dentro de un conglomerado son muy similares, la información adicional que aporta cada uno disminuye.

Por ejemplo, si una encuesta selecciona 100 conglomerados y dentro de cada uno 10 hogares, se obtiene una muestra de 1 000 hogares. Si todos los hogares de un mismo conglomerado comparten la misma característica (p. ej., acceso a electricidad), la muestra ofrece la misma precisión que una muestra aleatoria simple de solo 100 hogares. Analizar ingenuamente los 1 000 hogares como si fueran independientes conduce a subestimar gravemente los errores estándar.

#### 2.5.1.3 Recomendaciones prácticas

Para un análisis adecuado es indispensable que las bases de datos de encuestas incluyan:

* Identificadores de **estratos** y **UPM**.
* Pesos de muestreo a nivel de hogar o persona.

Cuando no se dispone de esta información, se recomienda que al menos se proporcionen **pesos replicados**, junto con instrucciones claras para calcular estimaciones y errores estándar.

## 2.6 Parámetros y Estimadores

Al analizar datos de encuestas, el primer paso es definir el parámetro de interés, un valor numérico fijo que describe una característica de toda la población (). Ejemplos comunes incluyen el total y la media de la población. Dado que no es práctico observar a toda la población, se utilizan encuestas por muestreo para inferir estos parámetros a partir de una muestra ().

El enfoque de inferencia y estimación basado en el diseño se basa en la estructura de probabilidad del plan de muestreo. Las propiedades estadísticas de los estimadores, como la precisión y el sesgo, se evalúan en relación con la distribución de aleatorización que el diseño de muestreo genera.

En el muestreo probabilístico, cada unidad de la población tiene una probabilidad de inclusión conocida. Estas probabilidades son la base para el cálculo de los pesos básicos de muestreo, que a su vez se utilizan para estimar los parámetros poblacionales a través de sumas ponderadas de los datos de la encuesta. Cuando se aplican correctamente, las estimaciones son insesgadas, lo que significa que en promedio coinciden con el valor real de la población si la encuesta se repitiera bajo las mismas condiciones.

### 2.6.1 Ajustes Adicionales de los Pesos

A pesar de lo anterior, los pesos básicos de muestreo suelen necesitar ajustes adicionales para mejorar la precisión y robustez de las estimaciones. Los ajustes más comunes son:

* Ajuste por no respuesta: Los pesos de las unidades que sí respondieron se incrementan para representar también a las unidades seleccionadas que no participaron, lo cual ayuda a reducir el sesgo y a aumentar la fiabilidad de los resultados.
* Calibración: Los pesos se modifican para asegurar que las sumas ponderadas de variables de control (como edad o sexo, obtenidas de censos o proyecciones demográficas) se alineen con los valores poblacionales conocidos. Esto es una herramienta útil para identificar problemas de cobertura o no respuesta y reforzar la validez de las estimaciones.

### 2.6.2 Importancia de los Ajustes

La calibración, en particular, es crucial porque permite comparar las estimaciones iniciales con los valores de referencia antes del ajuste. Esta comparación es una herramienta valiosa para detectar posibles problemas de cobertura o de no respuesta, lo que refuerza la validez de los resultados finales.

## 2.7 Estimación de totales

La estimación de totales en encuestas constituye un paso central en el análisis estadístico aplicado a poblaciones finitas. Gran parte de los indicadores de interés para la formulación de políticas públicas, como el número de personas en situación de pobreza, el total de ocupados o el gasto agregado de los hogares, se derivan de un total poblacional. Por esta razón, comprender cómo se definen y estiman los totales resulta fundamental para garantizar la calidad y pertinencia de la información producida.

En términos formales, si denota el valor de una variable de interés para la unidad , el total poblacional se define como

y su media poblacional como

Dado que en la práctica solo se observa una muestra , es necesario recurrir a estimadores que incorporen el diseño de muestreo. El estimador de Horvitz–Thompson (HT) es el más utilizado bajo el enfoque de diseño y se expresa como

donde son los pesos básicos de diseño y son las probabilidades de inclusión de primer orden.

En la práctica, los pesos de diseño suelen modificarse para reflejar procesos adicionales como el ajuste por no respuesta o la calibración a totales poblacionales conocidos, obteniendo así los pesos ajustados . El reemplazo de por permite mejorar la precisión y reducir sesgos en las estimaciones, especialmente cuando existen fuentes auxiliares de información confiables.

No obstante, toda estimación a partir de una muestra conlleva incertidumbre. Incluso cuando el estimador es insesgado, los resultados variarán de una muestra a otra debido al azar del diseño. Esta variabilidad se cuantifica mediante la **varianza de muestreo**, el **error estándar ()** o el **coeficiente de variación ()**. Estos indicadores son herramientas indispensables para evaluar la confiabilidad de los totales estimados y, por tanto, para interpretar de manera adecuada la información estadística.

Bajo el enfoque de diseño, la varianza insesgada del estimador de Horvitz–Thompson puede expresarse como:

donde y representan las probabilidades conjuntas de inclusión. Esta expresión requiere que el diseño de muestreo cumpla para todo par de unidades .

En síntesis, la estimación de totales es la piedra angular sobre la cual se construyen indicadores más complejos. Su estudio permite entender tanto la lógica de los ponderadores como la necesidad de medir y comunicar la precisión de las estimaciones, lo que constituye un elemento esencial en la producción de estadísticas de calidad.

### 2.7.1 Ejemplo ilustrativo

Para comprender de manera más tangible la importancia de considerar el diseño muestral en la estimación de totales y sus varianzas, analicemos un ejemplo sencillo.

Supóngase una población de tamaño y una muestra aleatoria simple de tamaño , seleccionada sin reemplazo, en la que se observan los valores . Bajo este diseño, la varianza estimada del estimador de Horvitz–Thompson se calcula como

donde corresponde a la varianza muestral de los valores observados. Sustituyendo en la expresión, se obtiene

En contraste, si se ignora el diseño de muestreo, un analista inexperto podría calcular erróneamente la varianza mediante la fórmula simplificada:

lo que conduciría a una **sobreestimación de la varianza** por no considerar las características del diseño de selección.

La estimación del total poblacional es . El error estándar correcto, calculado según el diseño de muestreo, es

En cambio, si la varianza se estimara usando un método ingenuo que ignore el diseño de muestreo, el intervalo de confianza resultante sería más amplio y desalineado, lo que podría conducir a inferencias erróneas. Este ejemplo evidencia claramente la relevancia de incorporar el diseño de muestreo al estimar varianzas, errores estándar e intervalos de confianza.

Si bien la fórmula general para la estimación de la varianza es aplicable a distintos diseños de muestreo, en la práctica rara vez se utiliza porque las probabilidades de inclusión de segundo orden y los pesos pareados suelen ser desconocidos para los usuarios secundarios. Incluso los propios productores de datos evitan calcular estos valores, pues existen métodos más simples y eficientes para la estimación de varianzas, como el método linealizado, la replicación y el bootstrap, que permiten cuantificar la incertidumbre sin necesidad de contar con información tan detallada.

## 2.8 Efecto del diseño (DEFF)

De acuerdo con Kish (1965, p. 258), el efecto del diseño (DEFF) se define como la relación entre la varianza de un estimador calculado bajo un diseño de muestreo complejo y la varianza del mismo estimador cuando se emplea un muestreo aleatorio simple (MAS) con igual tamaño de muestra. Su estimación se formula como:

donde corresponde a la varianza estimada de bajo el diseño complejo , mientras que representa la varianza estimada del mismo estimador bajo un MAS de igual tamaño. Este indicador permite cuantificar cuánto aumenta la varianza debido a la conglomeración y otras características propias de los diseños complejos en comparación con un muestreo simple. Según Naciones Unidas (2008, p. 49), el DEFF puede entenderse de tres maneras: como el factor de incremento de la varianza frente al MAS, como una medida de la pérdida relativa de precisión o como una indicación del aumento en el tamaño de la muestra que sería necesario en un diseño complejo para alcanzar el mismo nivel de varianza que en un MAS.

Según Park et al. (2003), el efecto del diseño de una encuesta puede desglosarse en tres factores multiplicativos:

* **Ponderación desigual:** La presencia de pesos muestrales no uniformes suele incrementar ligeramente la varianza; por ello, el uso de pesos iguales resulta ventajoso y explica por qué los diseños auto-ponderados son preferidos en encuestas de hogares.
* **Estratificación:** Cuando se aplica correctamente, puede disminuir la varianza, aunque en la práctica su efecto reductor suele ser moderado.
* **Muestreo en varias etapas:** Generalmente incrementa la varianza, ya que las unidades dentro de un mismo conglomerado tienden a ser más homogéneas entre sí que en comparación con las de otros conglomerados.

Al analizar encuestas, el DEFF se convierte en un indicador fundamental para medir la calidad de las estimaciones y orientar el diseño de estudios futuros. Un valor elevado evidencia que el diseño complejo introduce ineficiencias que incrementan la varianza y reducen la precisión de los resultados. Por el contrario, un valor cercano a uno indica que el diseño tiene un efecto mínimo sobre la varianza. Esta información permite a los investigadores identificar si es necesario ajustar la ponderación, optimizar la estratificación o modificar el tamaño del submuestreo para aumentar la eficiencia en levantamientos posteriores.

La interpretación de un DEFF alto debe hacerse con precaución, ya que no siempre implica que el diseño muestral sea inadecuado. Es fundamental considerar el contexto de la encuesta: un valor superior a tres podría parecer alarmante, pero a menudo se debe a limitaciones prácticas como restricciones presupuestales, dificultades logísticas o la necesidad de garantizar la participación de los entrevistados. En ciertos levantamientos de hogares, puede ser indispensable seleccionar solo una fracción de los individuos elegibles en cada hogar. Además, problemas de cobertura o tasas de no respuesta pueden aumentar la variabilidad de los pesos muestrales y, en consecuencia, elevar los valores del DEFF.

## 2.9 Muestreo aleatorio simple en dos etapas estratificado

Con la finalidad de mantener un equilibrio entre los costos económicos y las propiedades estadísticas de la estrategia de muestreo se puede aprovechar la homogeneidad dentro de los conglomerados y, así, no tener que realizar censos dentro de cada Unidad Primaria de Muestreo (UPM) sino, proceder a seleccionar una sub-muestra dentro del conglomerado seleccionado.

Los diseños de muestreo en las encuestas de hogares se caracterizan por ser diseños complejos los cuales involucran, entre otras, más de una etapa en la selección de las unidades de observación, estratos y estimadores complejos. En su mayoría, las unidades primarias de muestreo son seleccionadas dentro de los estrato. Ahora bien, según la teoría de muestreo *(Cochran, W. G., 1977)* se asume que el muestreo en cada estrato respeta el principio de la independencia. Esto es, las estimaciones del total, así como el cálculo y estimación de la varianza son el resultado de añadir o sumar para cada estrato la respectiva cantidad. Dentro de cada estrato con existen unidades primarias de muestreo, de las cuales se selecciona una muestra de tamaño mediante un diseño de muestreo aleatorio simple. Suponga, además que el sub-muestreo dentro de cada unidad primaria seleccionada es también aleatorio simple. En este sentido, para cada unidad primaria de muestreo seleccionada de tamaño se selecciona una muestra de elementos de tamaño .

Como es ampliamente conocido, el proceso de estimación de un parámetro particular, por ejemplo, la media de los ingresos consiste en multiplicar la observación obtenida en la muestra por su respectivo factor de expansión y dividirlo sobre la suma de los factores de expansión de acuerdo con el nivel de desagregación que se quiera estimar. Sin embargo, cuando el diseño es complejo como es el caso de las encuestas de hogares, la estimación de la varianza se torna un poco difícil de realizar utilizando ecuaciones cerradas. Para estos casos y como lo recomienda la literatura especializada *(Hansen, M. H., & Steinberg, J., 1956))*, se procede a utilizar la técnica del último conglomerado. Esta técnica consiste en aproximar la varianza sólo teniendo en cuenta la varianza de los estimadores en la primera etapa. Para esto se debe suponer que el diseño de muestreo fue realizado con reemplazo.

Para poder utilizar los principios de estimación del último conglomerado en las encuestas de hogares se definen las siguientes cantidades:

1. , que es el factor de expansión de la -ésima UPM en el estrato .
2. , que es el factor de expansión del -ésimo hogar para la -ésima UPM.
3. , que es el factor de expansión final del -ésimo elemento para toda la población .

## 2.10 Práctica en R

En esta sección se utilizarán las funciones estudiadas en el capítulo anterior para la manipulación de la base de datos de ejemplo. Inicialmente, se cargarán las librerías ggplot2 que permitirá generar gráficos de alta calidad en R, TeachingSampling que permite tomar muestras probabilísticas utilizando los diseños de muestreo usuales, survey y srvyr que permitirán definir los diseños muestrales y por último dplyr que permite la manipulación de las bases de datos.

library(ggplot2)  
library(TeachingSampling)  
library(dplyr)  
library(survey)  
library(srvyr)

Una vez cargada las librerías, se procede a calcular la cantidad de personas en la base de datos, el total de ingresos y total de gastos para cada UPM dentro de cada estrato:

data('BigCity')  
  
 FrameI <- BigCity %>% group\_by(PSU) %>%  
 summarise(Stratum = unique(Stratum),  
 Persons = n(),  
 Income = sum(Income),  
 Expenditure = sum(Expenditure))  
   
attach(FrameI)

head(FrameI, 10)

| PSU | Stratum | Persons | Income | Expenditure |
| --- | --- | --- | --- | --- |
| PSU0001 | idStrt001 | 118 | 70911.72 | 44231.78 |
| PSU0002 | idStrt001 | 136 | 68886.60 | 38381.90 |
| PSU0003 | idStrt001 | 96 | 37213.10 | 19494.78 |
| PSU0004 | idStrt001 | 88 | 36926.46 | 24030.74 |
| PSU0005 | idStrt001 | 110 | 57493.88 | 31142.36 |
| PSU0006 | idStrt001 | 116 | 75272.06 | 43473.28 |
| PSU0007 | idStrt001 | 68 | 33027.84 | 21832.66 |
| PSU0008 | idStrt001 | 136 | 64293.02 | 47660.02 |
| PSU0009 | idStrt001 | 122 | 33156.14 | 23292.16 |
| PSU0010 | idStrt002 | 70 | 65253.78 | 37114.76 |

Ahora bien, para calcular los tamaños poblacionales de los estratos (NIh) y los tamaños de muestra dentro de cada estrato (nIh), se realiza de la siguiente manera:

sizes = FrameI %>% group\_by(Stratum) %>%  
 summarise(NIh = n(),  
 nIh = 2,  
 dI = NIh/nIh)  
   
NIh <- sizes$NIh  
nIh <- sizes$nIh

head(sizes, 10)

| Stratum | NIh | nIh | dI |
| --- | --- | --- | --- |
| idStrt001 | 9 | 2 | 4.5 |
| idStrt002 | 11 | 2 | 5.5 |
| idStrt003 | 7 | 2 | 3.5 |
| idStrt004 | 13 | 2 | 6.5 |
| idStrt005 | 11 | 2 | 5.5 |
| idStrt006 | 5 | 2 | 2.5 |
| idStrt007 | 14 | 2 | 7.0 |
| idStrt008 | 7 | 2 | 3.5 |
| idStrt009 | 8 | 2 | 4.0 |
| idStrt010 | 8 | 2 | 4.0 |

Si se desea extraer una muestra probabilística bajo un diseño aleatorio simple estratificado, se procede a utilizar la función S.STSI de la librería TeachingSampling como se muestra a continuación:

samI <- S.STSI(Stratum, NIh, nIh)  
UI <- levels(as.factor(FrameI$PSU))  
sampleI <- UI[samI]

Ahora bien, con la función left\_join se procede a pegar los tamaños muestrales a aquellas UPM’s que fueron seleccionadas en la muestra:

FrameII <- left\_join(sizes,   
 BigCity[which(BigCity$PSU %in% sampleI), ])  
attach(FrameII)

Una vez se tiene la base de datos con la muestra de UMP’s. se selecciona aquellas variables que son de inetrés para el estudio como sigue a continuación:

head(FrameII, 10) %>% select(Stratum:Zone)

| Stratum | NIh | nIh | dI | HHID | PersonID | PSU | Zone |
| --- | --- | --- | --- | --- | --- | --- | --- |
| idStrt001 | 9 | 2 | 4.5 | idHH00030 | idPer01 | PSU0003 | Rural |
| idStrt001 | 9 | 2 | 4.5 | idHH00030 | idPer02 | PSU0003 | Rural |
| idStrt001 | 9 | 2 | 4.5 | idHH00031 | idPer01 | PSU0003 | Rural |
| idStrt001 | 9 | 2 | 4.5 | idHH00031 | idPer02 | PSU0003 | Rural |
| idStrt001 | 9 | 2 | 4.5 | idHH00031 | idPer03 | PSU0003 | Rural |
| idStrt001 | 9 | 2 | 4.5 | idHH00031 | idPer04 | PSU0003 | Rural |
| idStrt001 | 9 | 2 | 4.5 | idHH00031 | idPer05 | PSU0003 | Rural |
| idStrt001 | 9 | 2 | 4.5 | idHH00032 | idPer01 | PSU0003 | Rural |
| idStrt001 | 9 | 2 | 4.5 | idHH00032 | idPer02 | PSU0003 | Rural |
| idStrt001 | 9 | 2 | 4.5 | idHH00032 | idPer03 | PSU0003 | Rural |

Luego de tener la información muestral de la primera etapa en la base FrameII se procede a calcular los tamaños de muestra dentro de cada UPM’s. En este caso, a modo de ejemplo, se tomará el 10% del tamaño de la UPM y se utilizará la función ceiling la cual aproxima al siguiente entero.

HHdb <- FrameII %>%   
 group\_by(PSU) %>%  
 summarise(Ni = length(unique(HHID)))  
   
Ni <- as.numeric(HHdb$Ni)  
ni <- ceiling(Ni \* 0.1)  
sum(ni)

## [1] 700

Teniendo el vector de tamaños de muestra para cada UMP, se procede a realizar la selección mediante un muestreo aleatorio simple con la función S.SI de la librería TeachingSampling. A modo ilustrativo, la selección en la segunda etapa del diseño se realizará, inicialmente para la primera UPM. Posterior a eso, se realizará un ciclo “for” para hacerlo con las demás UPM’s. Para la primera UPM se realiza de la siguiente manera:

sam = S.SI(Ni[1], ni[1])  
  
clusterII = FrameII[which(FrameII$PSU == sampleI[1]),]  
  
sam.HH <- data.frame(HHID = unique(clusterII$HHID)[sam])  
  
clusterHH <- left\_join(sam.HH, clusterII, by = "HHID")  
  
clusterHH$dki <- Ni[1] / ni[1]  
  
clusterHH$dk <- clusterHH$dI \* clusterHH$dki  
  
sam\_data = clusterHH

head(sam\_data, 10) %>% select(Stratum:Zone)

| Stratum | NIh | nIh | dI | PersonID | PSU | Zone |
| --- | --- | --- | --- | --- | --- | --- |
| idStrt001 | 9 | 2 | 4.5 | idPer01 | PSU0003 | Rural |
| idStrt001 | 9 | 2 | 4.5 | idPer02 | PSU0003 | Rural |
| idStrt001 | 9 | 2 | 4.5 | idPer01 | PSU0003 | Rural |
| idStrt001 | 9 | 2 | 4.5 | idPer02 | PSU0003 | Rural |
| idStrt001 | 9 | 2 | 4.5 | idPer03 | PSU0003 | Rural |
| idStrt001 | 9 | 2 | 4.5 | idPer04 | PSU0003 | Rural |
| idStrt001 | 9 | 2 | 4.5 | idPer01 | PSU0003 | Rural |
| idStrt001 | 9 | 2 | 4.5 | idPer02 | PSU0003 | Rural |

Para las demás UPM’s seleccionadas en la etapa 1,

for (i in 2:length(Ni)) {  
 sam = S.SI(Ni[i], ni[i])  
   
 clusterII = FrameII[which(FrameII$PSU == sampleI[i]),]  
   
 sam.HH <- data.frame(HHID = unique(clusterII$HHID)[sam])  
   
 clusterHH <- left\_join(sam.HH, clusterII, by = "HHID")  
   
 clusterHH$dki <- Ni[i] / ni[i]  
   
 clusterHH$dk <- clusterHH$dI \* clusterHH$dki  
   
 data1 = clusterHH  
   
 sam\_data = rbind(sam\_data, data1)  
}  
encuesta <- sam\_data  
  
attach(encuesta)

Una vez se obtiene la muestra (como se mostró anteriormente), el paso siguiente es definir el diseño utilizado y guardarlo como un objeto en R para posteriormente poderlo utilizar y realizar el proceso de estimación de parámetros y cálculo de indicadores. Para realizar esta tarea, se utilizará el paquete srvyr el cual ya fue definido en el capítulo anterior. Para este ejemplo, el diseño de muestreo utilizado fue un estratificado-multietápico en el cual, los estratos correspondieron a la variable *Stratum*, las UPM’s correspondieron a la variable *PSU*, los factores de expansión están en la variable *dk* y por último, se le indica a la función as\_survey\_design que las UPM’s están dentro de los estrato con el argumento *nest = T*. A continuación, se presenta el código computacional:

diseno <- encuesta %>%  
 as\_survey\_design(  
 strata = Stratum,  
 ids = PSU,  
 weights = dk,  
 nest = T  
 )

Ya definido el diseño de muestreo como un objeto de R se puede empezar a extraer información del mismo. Por ejemplo, se pueden extraer los pesos de muestreo de dicho diseño con la función weights y luego sumarlos para revisar hasta cuánto me está expandiendo mi muestra. El código es el siguiente:

sum(weights(diseno))

## [1] 149583.5

Como se puede observar, el tamaño poblacional estimado utilizando el diseño propuesto es de . Sin embargo, el tamaño poblacional de la base BigCity es de . Es normal que esto suceda pero debe ser corregido puesto que la suma de los factores de expansión debe sumar el total de la población. La solución para esto es calibrar los pesos de muestreo que se abordará a continuación.

## 2.11 Calibrando con R

La calibración es un ajuste que se realiza a los pesos de muestreo con el propósito de que las estimaciones de algunas variables de control reproduzcan de forma perfecta los totales poblacionales de estas variables *(Sarndal, 2003)*. Esta propiedad de consistencia es deseable en un sistema de ponderadores. En este sentido, cuando los estudios por muestreo están afectados por la ausencia de respuesta, como en muchos casos pasa en las encuestas de hogares, es deseable tener las siguientes propiedades en la estructura inferencial que sustenta el muestreo:

* Sesgo pequeño o nulo.
* Errores estándares pequeños.
* Un sistema de ponderación que reproduzca la información auxiliar disponible.
* Un sistema de ponderación que sea eficiente al momento de estimar cualquier característica de interés en un estudio multipropósito.

La calibración es usualmente el último paso en el ajuste de los ponderadores. Hace uso de información auxiliar que reduce la varianza y corrige los problemas de cobertura que no pudieron ser corregidos en los pasos previos.

Puesto que el estimador de calibración depende exclusivamente de la información auxiliar disponible, esta información puede aparecer en diversas formas:

1. Puede estar de forma explícita en el marco de unidades.
2. Puede ser un agregado poblacional proveniente de un censo o de registros administrativos.
3. Puede ser una estimación poblacional muy confiable.

Particularmente, en encuestas de hogares, existen conteos de personas disponibles a nivel de desagregaciones de interés. Por ejemplo, número de personas por edad, raza y género que se permite utilizar como información auxiliar para calibrar las estimaciones.

La necesidad de calibrar en las encuestas de hogares es porque no todos los grupos de personas se cubren apropiadamente desde el diseño de muestreo. Además, las estimaciones del número de personas en estos subgrupos son menores a las proyecciones que se tienen desde los censos. Por último, al ajustar los pesos para que sumen exactamente la cifra de los conteos censales, se reduce el sesgo de subcobertura.

Para ejemplificar el estimador de calibración en R usando la base de datos de ejemplo se utilizarán la función calibrate del paquete survey. En primer lugar, para poder calibrar se requiere construir la información poblacional a la cual se desea calibrar. En este ejemplo se calibrará a nivel de zona y sexo. Por tanto, los totales se obtienen como sigue:

library(survey)  
totales <- colSums(  
 model.matrix(~ -1 + Zone:Sex, BigCity))

En la salida anterior se puede observar que, por ejemplo, en la zona rural hay 37238 mujeres mientras que en la urbana hay 41952. De igual manera se puede leer para el caso de los hombres.

Una vez obtenido estos totales, se procede a utilizar la función calibrate para calibrar los pesos de muestreo como sigue:

diseno\_cal <- calibrate(  
 diseno, ~ -1 + Zone:Sex, totales, calfun = "linear")

Luego de que se hayan calibrado los pesos se puede observar que, al sumar los pesos calibrados estos reproducen el total poblacional de la base de ejemplo.

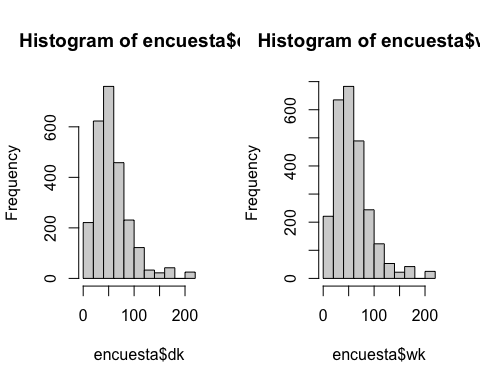
sum(weights(diseno\_cal))

## [1] 150266

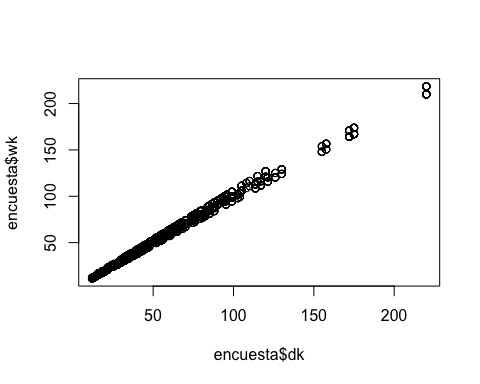
encuesta$wk <- weights(diseno\_cal)

Dado que uno de los principios de los pesos calibrados es que dichos pesos no sean muy diferentes a los pesos originales que provienen del diseño de muestreo, se puede observar a continuación, la distribución de los pesos, sin calibrar y calibrados respectivamente.

par(mfrow = c(1,2))  
hist(encuesta$dk)  
hist(encuesta$wk)



plot(encuesta$dk,encuesta$wk)



Region <- as.numeric(  
 gsub(pattern = "\\D",  
 replacement = "", x = encuesta$Stratum))  
encuesta$Region <-   
 cut(Region, breaks = 5,  
 labels = c("Norte","Sur","Centro","Occidente","Oriente"))  
encuesta %<>% mutate(  
 CatAge = case\_when(  
 Age <= 5 ~ "0-5",  
 Age <= 15 ~ "6-15",  
 Age <= 30 ~ "16-30",  
 Age <= 45 ~ "31-45",  
 Age <= 60 ~ "46-60",  
 TRUE ~ "Más de 60"  
 ),  
 CatAge = factor(  
 CatAge,  
 levels = c("0-5", "6-15", "16-30", "31-45",  
 "46-60", "Más de 60"),  
 ordered = TRUE  
 )  
)  
saveRDS(object = encuesta, file = "../Curso Tellez/Data/encuesta.rds")

# 3 Manejando una base de encuestas de hogares con R

## 3.1 Fundamentos básicos de R y Rstudio

R fue creado en 1992 en Nueva Zelanda por Ross Ihaka y Robert Gentleman. A manera introductoria, R es un software diseñado para realizar análisis estadístico tanto sencillos como complejos. Este software ha ganado popularidad en el gremio estadístico y no estadístico, puesto que su manejo es sencillo y, además, es de libre uso (puede descargarse en <https://www.r-project.org>), es decir, no requiere de ninguna licencia para su utilización. Como lo mencionan Santana Sepúlveda y Mateos Farfán (2014), R es un lenguaje de programación de libre distribución, bajo Licencia GNU, que se mantiene en un ambiente para el cómputo estadístico y gráfico. Este software está diseñado para utilizarse en distintos sistemas operativos como Windows, MacOS o Linux. El concepto de ambiente se enfoca en caracterizarlo como un sistema totalmente planificado y coherente, a diferencia de otros softwares de análisis de datos que suelen consistir en acumulaciones de herramientas específicas y poco flexibles.

Ahora bien, R es un lenguaje de programación, por lo que su interfaz puede resultar poco amigable para quienes inician en el lenguaje. Para facilitar su uso, se creó RStudio, un Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés). Esto significa que RStudio es un programa que permite manejar R de manera más cómoda y visual, optimizando la experiencia de análisis y programación.

R es un software libre y de código abierto que ha ganado gran popularidad en el procesamiento de encuestas y la investigación social, convirtiéndose en una herramienta de elección para aplicar los desarrollos científicos y metodológicos más recientes en el análisis de datos de encuestas (R Core Team, 2024). Su carácter abierto permite que investigadores de todo el mundo aporten funciones y paquetes propios al Comprehensive R Archive Network (CRAN), poniéndolos a disposición de la comunidad académica y profesional. Entre sus recursos más destacados se encuentra el paquete samplesize4surveys (Gutiérrez, 2020), que facilita el cálculo de tamaños de muestra para individuos y hogares en encuestas repetidas, de panel y rotacionales. Asimismo, los paquetes sampling (Tillé y Matei, 2016) y TeachingSampling (Gutiérrez, 2015) ofrecen soporte para seleccionar muestras probabilísticas a partir de marcos de muestreo bajo diferentes diseños y algoritmos.

Para el análisis de datos de encuestas de hogares, el paquete survey (Lumley, 2024) permite especificar el diseño muestral mediante la función svydesign() y obtener estimaciones correctas de errores estándar. El paquete convey (Pessoa et al., 2024) complementa este proceso al facilitar el cálculo de medidas de desigualdad. En el ámbito del modelado de regresiones, svydiags (Valliant, 2024) incluye herramientas de diagnóstico como análisis de residuos, valores de apalancamiento, factores de inflación de varianza y pruebas de colinealidad, mientras que PracTools (Valliant et al., 2025) proporciona utilidades para el cálculo del tamaño de muestra, el diseño de muestreo, la estimación de efectos de diseño y el análisis de componentes de varianza en esquemas multietápicos.

## 3.2 Algunas librerías de interés

Puesto que R es un lenguaje colaborativo el cual permite que la comunidad vaya haciendo aportes al desarrollo de funciones dentro de paquetes o librerías. Alguna de las librerías más usadas para el análisis de bases de datos son las siguientes:

* dplyr, dplyr es la evolución del paquete plyr, enfocada en herramientas para trabajar con marcos de datos (de ahí la d en el nombre). Según Hadley Wickham, las siguientes son las tres propiedades principales de la librería:
  1. Identificar las herramientas de manipulación de datos más importantes necesarias para el análisis de datos y hacerlas fáciles de usar desde R.
  2. Proporcionar un rendimiento ultrarrápido para los datos en memoria escribiendo piezas clave en C++.
  3. Utilizar la misma interfaz para trabajar con datos sin importar dónde estén almacenados, ya sea en un marco de datos, una tabla de datos o una base de datos.Esta librería permite manejar eficientemente las bases de datos.
* tidyverse, es una colección de paquetes disponibles en R y orientados a la manipulación, importación, exploración y visualización de datos y que se utiliza exhaustivamente en ciencia de datos. El uso de tidyverse permite facilitar el trabajo estadístico y la generación de trabajos reproducibles. Está compuesto de los siguientes paquetes: readr, dplyr, ggplot2, tibble, tidyr, purr, stringr, forcats
* readstata13, este paquete permite leer y escribir todos los formatos de archivo de Stata (versión 17 y anteriores) en un marco de datos R. Se admiten las versiones de formato de archivo de datos 102 a 119. para leer las bases de datos de STATA. Además, el paquete admite muchas características del formato Stata dta, como conjuntos de etiquetas en diferentes idiomas o calendarios comerciales.
* survey, este paquete ha sido elaborado por el Profesor Thomas Lumley (Lumley, T. 2011) y nos proporciona funciones en R útiles para analizar datos provenientes de encuestas complejas.Alguno de los parámetros que se pueden estimar usando este paquete son medias, totales, razones, cuantiles, tablas de contingencias, modelos de regresión, modelos loglineales, entre otros.
* srvyr, este paquete permite utilizar el operador *pipe operators* en las consultas que se realizan con el paquete survey.
* ggplot2, es un paquete de visualización de datos para el lenguaje R que implementa lo que se conoce como la *Gramática de los Gráficos*, que no es más que una representación esquemática y en capas de lo que se dibuja en dichos gráficos, como lo pueden ser los marcos y los ejes, el texto de los mismos, los títulos, así como, por supuesto, los datos o la información que se grafica, el tipo de gráfico que se utiliza, los colores, los símbolos y tamaños, entre otros.
* TeachingSampling, este paquete permite al usuario extraer muestras probabilísticas y hacer inferencias a partir de una población finita basada en varios diseños de muestreo. Entre los diseño empleados en esta librería están: Muestreo Aleatorio Simple (MAS), Muestreo Bernoullí, Muestreo Sistemático, PiPT, PPT, estre otros.
* samplesize4surveys, este paquete permite calcular el tamaño de muestra requerido para la estimación de totales, medias y proporciones bajo diseños de muestreo complejos.

Antes de poder utilizar las diferentes funciones que cada librería tiene, es necesario descargarlas de antemano de la web. El comando install.packages permite realizar esta tarea. Note que algunas librerías pueden depender de otras, así que para poder utilizarlas es necesario instalar también las dependencias.

install.packages("dplyr")  
install.packages("tidyverse")  
install.packages("readstata13")   
install.packages("survey")  
install.packages("srvyr")  
install.packages("ggplot2")  
install.packages("TeachingSampling")  
install.packages("samplesize4surveys")

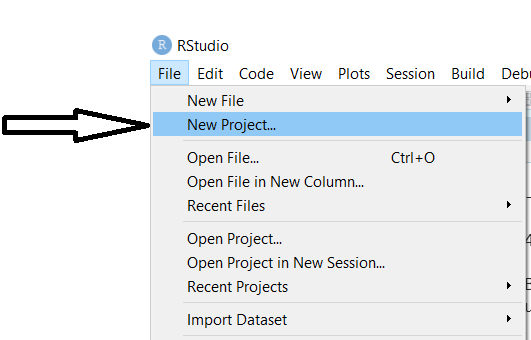
Una vez instaladas las librerías hay que informarle al software que vamos a utilizarlas con el comando library. *Recuerde que es necesario haber instalado las librerías para poder utilizarlas*.

rm(list = ls())  
  
library("dplyr")  
library("tidyverse")  
library("readstata13")   
library("survey")  
library("srvyr")  
library("ggplot2")  
library("TeachingSampling")  
library("samplesize4surveys")

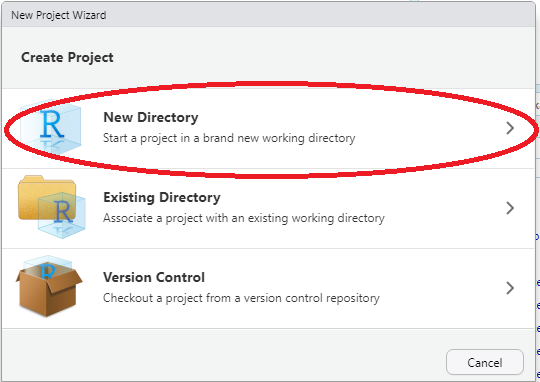
## 3.3 Cración de proyectos en R

Una vez se descargan e instalan las librerías o paquetes en R el paso siguientes es crear proyectos. Un proyecto de R se define como un archivo que contiene los archivos de origen y contenido asociados con el trabajo que se está realizando. Adicionalmente, contiene información que permite la compilación de cada archivo de R a utilizar, mantiene la información para integrarse con sistemas de control de código fuente y ayuda a organizar la aplicación en componentes lógicos.

Ahora bien, por una cultura de buenas practicas de programación, se recomienda crear un proyecto en el cual se tenga disponible toda la información a trabajar. A continuación, se muestran los pasos para crear un proyecto dentro de RStrudio.

* **Paso 1:** Abrir RStudio.
* **Paso 2:** ir a file -> New Project 
* *Paso 3:* Tipos de proyecto.

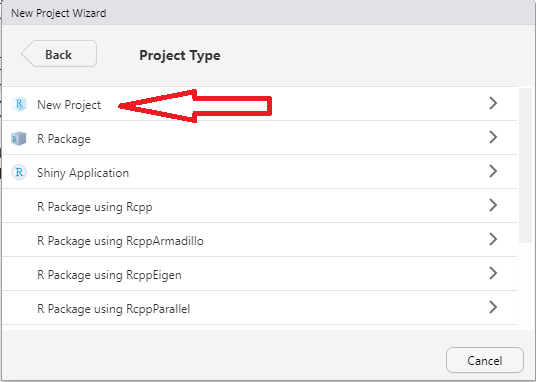
Para este ejemplo se tomará *New Directory*



*Tipos de proyectos*

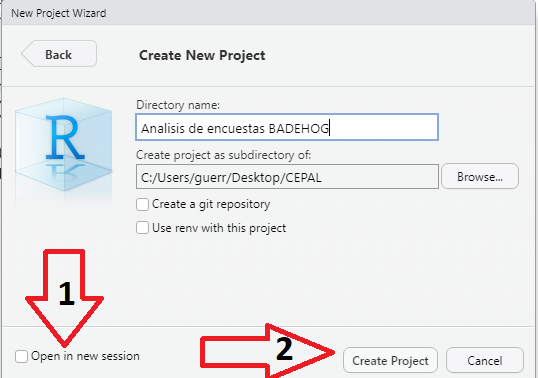
Algo a tener en cuenta en este paso es que en *New Directory* RStudio brinda una variedad de opciones dependiendo las características del procesamiento que desea realizar. Ahora bien, si se cuenta con algunos código previamente desarrollados y se desea continuar con ese proyecto, se debe tomar la opción *Existing Directory* . Por último, Si se cuenta con cuenta en *Git* y se desea tener una copia de seguridad, se debe emplear la opción *Version Control*.

* *Paso 4:* Seleccionar el tipo de proyecto.



*Seleccionar el tipo de proyecto*

* *Paso 5*: Diligenciar el nombre del proyecto y la carpeta de destino.



*Nombre de proyecto*

Al realizar esto pasos permite que todas rutinas creadas dentro del proyecto estén ancladas a la carpeta del proyecto.

## 3.4 Lectura de las bases de datos y manipulación

Es muy usual que al trabajar proyectos en R sea necesario importar bases de datos con información relevante para un estudio en particular. En Colombia, por ejemplo, en la *Encuesta de Calidad de Vida (ECV, por sus siglas)* es necesario, una vez se realiza el trabajo de campo, importar la información recolectada para poder ajustar los factores de expansión y posteriormente estimar los parámetros. Los formatos de bases de datos que R permite importar son diversos, entre ellos se tienen xlsx, csv, txt, STATA, etc. Particularmente, para la lectura de bases de datos provenientes de STATA 13 se realiza con la función read.dta13. Una vez leída la base de datos en el formato mencionado anteriormente se procede a transformar en el formato .RDS el cual es un formato más eficiente y propio de R. Para ejemplificar los procedimientos en R se utilizará la base de datos de *Pesquisa Nacional por Amostra de Domicílios 2015*  de Brasil la cual está en formato .dta el cual se lee en R con la función read.dta13. Posteriormente se transformará al formato .rds con la función saveRDS el cual es un formato propio de R y por último se cargar esta base. Lo pasos anteriores se realiza como sigue:

Primero se carga la base en formato dta con la librería read.dta13 y se guarda en formato rds con la función saveRDS `

data1 <- read.dta13("Z:/BC/BRA\_2015N.dta")  
saveRDS(data1, "../data/BRA\_2015N.rds")

Una vez guardada la base en nuestros archivos de trabajo, se procede a cargar la base a R con la función readRDS para poder utilizar toda la información que en ella se contiene.

data2 <- readRDS("Data/BRA\_2015N.rds")

Una vez cargada la base de datos en R ésta se puede empezar a manipular según las necesidades de cada investigador. En este sentido, una de las primeras revisiones que se realizan al cargar las bases de datos es revisar su dimensión, es decir, chequear la cantidad de filas y columnas que tenga la base. Lo anterior se puede hacer con la función nrow. Dicha función identifica el número de registros (unidades efectivamente medidas) en la base de datos y la función ncol muestra el número de variables en la base de datos. Los códigos computacionales son los siguientes:

nrow(data2)

## [1] 356904

ncol(data2)

## [1] 109

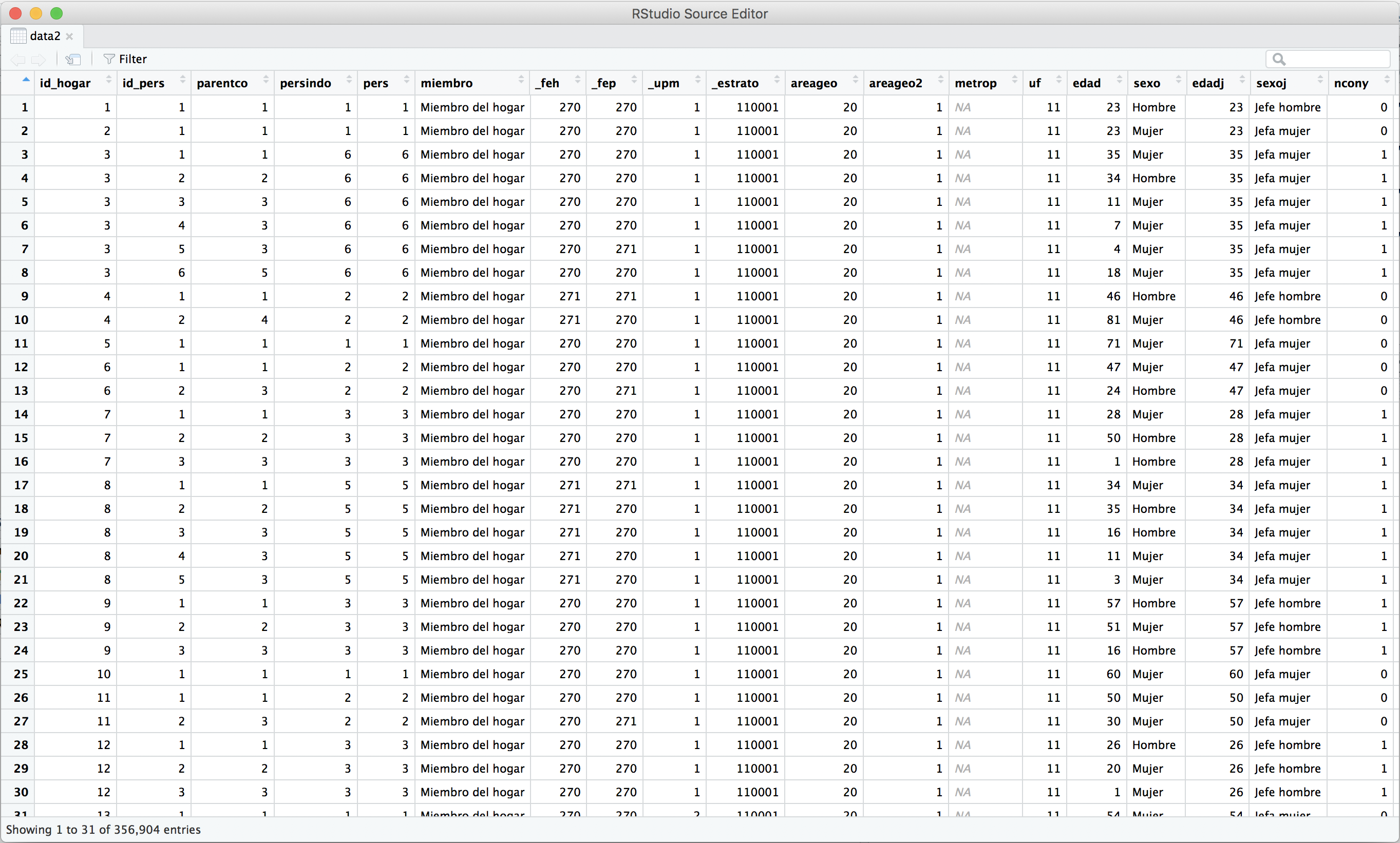
Una forma resumida de revisar la cantidad de filas y columnas que tiene la base de datos es usar la función dim. Esta función nos devuelve un vector indicado en su primera componente la cantidad de fila y en su segundo la cantidad de columnas como se muestra a continuación:

dim(data2)

## [1] 356904 109

Es usual que en las encuestas de hogares las bases de datos sean muy extensas, es decir, contengan una cantidad importante de variables medidas (filas) y por lo general, el tamaño de la muestra de estos estudios con grandes. Es por lo anterior que, para poder visualizar dichas bases una vez cargadas en R, es necesario hacerlo de manera externa. Esto es, abrir una pestaña diferente en R y hacer la navegación de la base como un texto plano. Lo anterior se realiza con la función View como se muestra a continuación:

View(data2)



*Visor de bases de datos de RStudio*

Otro chequeo importante que se debe realizar al momento de cargar una base de datos en R es el reconocimiento de las variables que incluye. Esto se puede hacer utilizando la función names la cual identifica las variables de la base de datos.

names(data2)

La función names solo devuelve un vector un vector con los nombres de las variables que contiene la base. Sin embargo, si se quiere profundizar en qué información contiene cada variable, La función str muestra de manera compacta la estructura de un objeto y sus componentes. Para nuestra base se utilizaría de la siguiente manera:

str(data2)

Como se puede observar en la salida anterior, por ejemplo, la variable *id\_hogar* es de tipo *Entero* al igual que *id\_pers* mientras que *cotiza\_ee* es un factor con 2 niveles. Como se observa, esta función es muy útil al momento de querer tener un panorama amplio del contenido y clase de cada variable en una base de datos, particularmente, en una encuesta de hogares en donde se tiene, por la misma estructura del estudio, muchas clases o tipos de variables medidas.

## 3.5 El operador pipe

El software estadístico R es un lenguaje de programación creado por estadísticos para estadísticos. Una de las contribuciones recientes es el desarrollo de los pipelines que permiten de una forma intuitiva generar consultas y objetos desde una base de datos. El operador *pipe*, %>%, viene del paquete magrittr (Bache, S. et al., 2022) y está cargado automáticamente en los paquetes del Tidyverse.

El objetivo del operador pipe es ayudar a escribir código de una manera que sea más fácil de leer y entender. En este sentido, el operador %>% permite “encadenar” operaciones en el sentido que el resultado de una operación anterior se convierta en el input de la siguiente operación. A continuación, ejemplificaremos el uso del %>% en la base de datos de Brasil haciendo un conteo del total de elementos que contiene la base de datos utilizando la función count.

data2 %>% count()

## n  
## 1 356904

Otra operación que se puede realizar en R es re-codificar los niveles de los factores que en muchas ocasiones son necesarios en las encuestas de hogares. El siguiente código permite generar los nombres de los estados en Brasil.

data2$estados <- factor(data2$uf,   
 levels = c(11:17, 21:29, 31:33, 35, 41:43, 50:53),   
 labels = c("Rondonia", "Acre", "Amazonas", "Roraima",   
 "Para", "Amapa", "Tocantins", "Maranhao",   
 "Piaui", "Ceara", "RioGrandeNorte", "Paraiba",  
 "Pernambuco", "Alagoas", "Sergipe", "Bahia",   
 "MinasGerais", "EspirituSanto", "RioJaneiro",   
 "SaoPaulo", "Parana", "SantaCatarina",   
 "RioGrandeSur", "MatoGrossoSur", "MatoGrosso",   
 "Goias", "DistritoFederal"))

Adicionalmente, para efectos de visualización en tablas y gráficos es conviene codificar los nombres de las variables. Para este ejemplo, se codificarán de la siguiente manera:

data2$deptos <- factor(data2$uf,   
 levels = c(11:17, 21:29, 31:33, 35, 41:43, 50:53),   
 labels = c("RO", "AC", "AM", "RR", "PA",   
 "AP", "TO", "MA", "PI", "CE", "RN", "PB",   
 "PE", "AL", "SE", "BA", "MG", "ES", "RJ", "SP",  
 "PR", "SC", "RS", "MS", "MT", "GO", "DF"))

Por otro lado, existe una gama amplia de funciones que se pueden utilizar con el operador %>%, A continuación, se enlistan una serie de funciones muy útiles al momento de hacer análisis con bases de datos provenientes de encuestas de hogares:

* **filter**: mantiene un criterio de filtro sobre alguna variable o mezcla de variables.
* **select**: selecciona columnas por nombres.
* **arrange**: ordena las filas de la base de datos.
* **mutate**: añade nuevas variables a la base de datos.
* **summarise**: reduce variables a valores y los presenta en una tabla.
* **group\_by**: ejecuta funciones y agrupa el resultado por las variables de interés.

Ejemplificando alguna de las funciones mostradas anteriormente, una de las primeras consultas que se realizan en las encuestas de hogares es saber el número de encuestas (personas) realizadas y que están contenida en la base de datos. Usando %>% se realiza de la siguiente manera:

data2 %>% count()

## n  
## 1 356904

Otro de los ejercicios que se hacen usualmente con las encuestas de hogares está relacionado con saber la cantidad de hogares que hay en el país de estudio. Una de las formas más sencillas de hacer esta revisión es usar la función filter. Las encuestas de hogares muchas veces recopilan información a nivel de viviendas, hogares y personas. Particularmente, las bases de datos que están disponibles en BADEHOG están a nivel de persona. Ahora bien, para saber la cantidad de hogares que se encuestaron basta con filtrar por hogar porque sólo hay un jefe de hogar por hogar, como se muestra a continuación:

datahogar1 <- data2 %>% filter(parentco == 1)  
datahogar2 <- data2 %>% filter(paren\_ee == "Jefe")

Por otro lado, si el interés ahora es filtrar la base de datos por la ubicación de la persona en el área rural y urbana se realiza de la siguiente manera:

dataurbano <- data2 %>%   
 filter(area\_ee == "Area urbana")  
datarural <- data2 %>%   
 filter(area\_ee == "Area rural")

En este mismo sentido, si el objetivo ahora es filtrar la base de datos por algunos ingresos particulares mensuales por personas, por ejemplo, altos o bajos, se realiza de la siguiente manera:

dataingreso1 <- data2 %>%   
 filter(ingcorte %in% c(50, 100))  
  
dataingreso2 <- data2 %>%   
 filter(ingcorte %in% c(1000, 2000))

Otra función muy útil en el análisis en encuestas de hogares es la función select la cual, como se mencionó anteriormente permite seleccionar un grupo de variables de interés a analizar. Si por ejemplo, se desea seleccionar de la base de ejemplo solo las variables identificación del hogar (id\_hogar), unidades primarias de muestreo (\_upm), factores de expansión (\_feh) y estratos muestrales ( \_estrato) se realiza de la siguiente manera:

datared <- data2 %>% select(`id\_hogar`, `\_upm`,  
 `\_feh`, `\_estrato`)  
  
datablue <- data2 %>% select(id\_pers, edad,   
 sexo, ingcorte)

La función select no solo sirve para seleccionar variables de una base de datos, también se puede utilizar para eliminar algunas variables de la base de datos que ya no son de interés para el análisis o que simplemente se generaron en la manipulación de la base de datos como variables puentes para realizar algunos cálculos de interés. Por ejemplo, si se desea eliminar de la base de datos de ejemplo las variables identificación del hogar (id\_hogar) e identificación de las personas (id\_pers) se realiza introduciendo un signo “menos” (-) delante del nombre de la variable como sigue:

datagrey <- data2 %>% select(-id\_hogar, -id\_pers)

Por otro lado, si el objetivo ahora en análisis de las encuestas de hogares es ordenar las filas de la base por alguna variable en particular, se utiliza en R la función arrange para realizar esta operación. A continuación, se ejemplifica con la base de datos de ejemplo, cómo se ordena la base de acuerdo con la variable *ingcorte*:

datadog <- datablue %>% arrange(ingcorte)  
datadog %>% head()

## id\_pers edad sexo ingcorte  
## 1 1 38 Mujer 0  
## 2 2 12 Mujer 0  
## 3 1 26 Hombre 0  
## 4 2 29 Mujer 0  
## 5 1 50 Hombre 0  
## 6 1 53 Mujer 0

Es posible utilizar la función arrange para hacer ordenamientos más complicados. Por ejemplo, ordenar por más de una variable. A modo de ejemplo, ordenemos la base de datos *datablue* de acuerdo con las variables *sexo* y *edad*

datablue %>% arrange(sexo, edad) %>% head()

## id\_pers edad sexo ingcorte  
## 1 6 0 Hombre 660.4400  
## 2 6 0 Hombre 162.5000  
## 3 3 0 Hombre 381.6667  
## 4 5 0 Hombre 320.0000  
## 5 6 0 Hombre 375.0000  
## 6 4 0 Hombre 1425.0000

También es posible utilizar la función arrange junto con la opción desc() para que el ordenamiento sea descendente.

datablue %>% arrange(desc(edad)) %>% head()

## id\_pers edad sexo ingcorte  
## 1 2 115 Mujer 103.0000  
## 2 4 110 Mujer 1156.5300  
## 3 2 107 Hombre 415.5904  
## 4 1 107 Mujer 1754.4600  
## 5 3 105 Mujer 380.7904  
## 6 2 105 Mujer 898.3200

## 3.6 Funciones **mutate, summarise y group\_by** en encuestas de hogares

Las funciones mutate, summarise y group\_by están cargadas en el paquete tidyverse y son muy importantes al momento de realizar análisis en encuestas de hogares. En primer lugar, la función mutate permite computar transformaciones de variables en una base de datos. Usualmente, en las encuestas de hogares es necesario crear nuevas variables, por ejemplo, si el hogar está en estado de pobreza extrema o no la cual se calcula a partir de los ingresos del hogar, la función mutate proporciona una interface clara para realizar este tipo de operaciones. A modo de ejemplo, utilizaremos la base de ejemplo para crear una nueva variable llamada *ingreso2* la cual es el doble de los ingresos por persona dentro de un hogar. Los códigos computacionales se muestran a continuación:

datablue2 <- datablue %>%   
 mutate(ingreso2 = 2 \* ingcorte)  
datablue2 %>% head()

## id\_pers edad sexo ingcorte ingreso2  
## 1 1 23 Hombre 800.0 1600.0  
## 2 1 23 Mujer 1150.0 2300.0  
## 3 1 35 Mujer 904.4 1808.8  
## 4 2 34 Hombre 904.4 1808.8  
## 5 3 11 Mujer 904.4 1808.8  
## 6 4 7 Mujer 904.4 1808.8

No solo se puede crear una nueva variable, si es necesario, se pueden crear más de una variable en la base de datos. Cabe recalcar que la función mutate reconoce sistemáticamente las variables que van siendo creadas de manera ordenada. A continuación, se presenta cómo crear más de una nueva variable en la base de datos:

datacat <- datablue %>%   
 mutate(ingreso2 = 2 \* ingcorte,  
 ingreso4 = 2 \* ingreso2)  
datacat %>% head()

## id\_pers edad sexo ingcorte ingreso2 ingreso4  
## 1 1 23 Hombre 800.0 1600.0 3200.0  
## 2 1 23 Mujer 1150.0 2300.0 4600.0  
## 3 1 35 Mujer 904.4 1808.8 3617.6  
## 4 2 34 Hombre 904.4 1808.8 3617.6  
## 5 3 11 Mujer 904.4 1808.8 3617.6  
## 6 4 7 Mujer 904.4 1808.8 3617.6

Ahora bien, la función summarise funciona de forma similar a la función mutate, excepto que en lugar de añadir nuevas columnas crea un nuevo data frame. Como se mencionó anteriormente esta función sirve para resumir o “colapsar filas”. Toma un grupo de valores como input y devuelve un solo valor; por ejemplo, hallar la media de los ingresos, percentiles o medidas de dispersión.

Por otro lado, la función group\_by permite agrupar información de acuerdo con una(s) variable(s) de interés. El siguiente código permite generar el número de encuestas efectivas en cada uno de los estados de Brasil. El comando group\_by agrupa los datos por estados, el comando summarise hace los cálculos requeridos y el comando arrange ordena los resultados

data2 %>%   
 group\_by(estados) %>%   
 summarise(n = n()) %>% arrange(desc(n)) %>% head()

## # A tibble: 6 × 2  
## estados n  
## <fct> <int>  
## 1 SaoPaulo 40008  
## 2 MinasGerais 32933  
## 3 RioGrandeSur 26259  
## 4 Bahia 26155  
## 5 RioJaneiro 25858  
## 6 Para 22489

Hay otro tipos de análisis que se quieren realizar en encuestas de hogares, por ejemplo, generar el número de encuestas efectivas discriminado por el sexo del respondiente. A continuación, se presenta el código computacional:

data2 %>%   
 group\_by(sexo) %>%   
 summarise(n = n()) %>% arrange(desc(n))

## # A tibble: 2 × 2  
## sexo n  
## <fct> <int>  
## 1 Mujer 183681  
## 2 Hombre 173223

Si ahora se desea realizar la consulta del número de encuestas efectivas por área geográfica, se realiza de la siguiente manera:

data2 %>%   
 group\_by(area\_ee) %>%   
 summarise(n = n()) %>% arrange(desc(n))

## # A tibble: 2 × 2  
## area\_ee n  
## <fct> <int>  
## 1 Area urbana 304564  
## 2 Area rural 52340

Otras consultas que se realizan de manera frecuente en encuestas de hogares es reporta el número efectivo de encuestas clasificado por parentezco (jefe de hogar, hijos, conyugues, etc)

data2 %>%   
 group\_by(paren\_ee) %>%   
 summarise(n = n()) %>% arrange(desc(n))

## # A tibble: 6 × 2  
## paren\_ee n  
## <fct> <int>  
## 1 Hijos 126206  
## 2 Jefe 117939  
## 3 Cónyuge 73725  
## 4 Otros parientes 36508  
## 5 Otros no parientes 2342  
## 6 Servicio doméstico 184

## 3.7 Medidas descriptivos y reflexiones

En estadística, según *Tellez Piñerez, C. F., & Lemus Polanía, D. F. (2015)* las medidas descriptivas permiten la presentación y caracterización de un conjunto de datos con el fin de poder describir apropiadamente las diversas características presentes en la información de la muestra. Involucra cualquier labor o actividad para resumir y describir los datos univariados o multivariados sin tratar de hacer inferencia más allá de los mismos. Este tipo de análisis son primordiales en cualquier encuesta de hogares dado que, permiten tener una idea inicial del comportamiento de la población en ciertas variables de estudio. A continuación, se presentan las funciones básicas en R para realizar análisis descriptivo.

* Media: mean()
* Mediana: median()
* Varianza: var()
* Desviación estándar: sd()
* Percentiles: quantile()
* Algunas medidas descriptivas: summary()
* Covarianza: cov( , )
* Correlación: cor( , )

Ahora bien, para continuar con lo análisis de las encuestas de hogares es necesario que el lector tenga claro algunos conceptos básicos en el muestreo probabilístico. A continuación, se dan unas definiciones básicas:

* *¿Qué es una encuesta?*

Según Groves, R. M., et al (2011) una encuesta es un método sistemático para recopilar información de una muestra de elementos con el propósito de construir descriptores cuantitativos de los parámetros de la población.

* *¿Qué es una muestra?*

La definición más básica de una muestra es un subconjunto de la población. Esta definición es muy general dado que, no es específico de si la muestra es representativa de una población o no.

* *¿Qué es una muestra representativa?*

Según *Gutiérrez (2016)* una muestra representativa es un modelo reducido de la población y de aquí se desprende un argumento de validez sobre la muestra. En pocas palabras, se desea que la muestra representativa tenga la cantidad de información suficiente para poder hacer una inferencia adecuada a la población.

* *¿Está bien sacar conclusiones sobre una muestra?*

Si la muestra es representativa, las conclusiones que se obtienen de la población utilizando las técnicas de muestreo adecuadas, son correctas. Sin embargo, si se toma una muestra no representativa, no es correcto realizar inferencias dado que estas no representan la realidad de la población.

## 3.8 Algunas reflexiones generales

Como se mencionó anteriormente, antes de realizar los análisis en las encuestas de hogares es necesario hacernos algunas preguntas que nos permiten dar claridad de los análisis que se desean hacer. A continuación, se presentan las preguntas:

* *Si calculamos el promedio de los ingresos en una encuesta, ¿qué significa esa cifra?*

Esta cifra representa los ingresos medios que reportaron las personas entrevistadas en el estudio. En ningún momento se puede hablar de que este valor representa a la población a la cual queremos hacer inferencia. Para poder realizar las conclusiones a nivel poblacional se deben utilizar los factores de expansión que se obtuvieron empleando el diseño muestral.

* *Si calculamos el total de los ingresos en una encuesta, ¿qué significa esa cifra?*

Similar a lo anterior, significa los ingresos totales que reportaron los entrevistados en el estudio. Se recalca que, bajo ninguna circunstancia se puede inferir que este valor muestral representa a la población de estudio.

* *¿Qué necesitamos para que la inferencia sea precisa y exacta?*

Se requiere de un buen diseño muestral, que la muestra que se recolecte sea representativa de la población en estudio y que el tamaño de muestra sea suficiente para poder inferir en todas las desagregaciones, tanto geográficas como temáticas que se plantearon en el diseño muestral.

* *¿Qué es el principio de representatividad?*

La representatividad es la característica más importante de una muestra probabilística, y se define como la capacidad que tiene una muestra de poder representar a la población a la cual se desea hacer inferencia. En este sentido, el muestreo adquiere todo su sentido en cuanto se garantice que las características que se quieren medir en la población quedan reflejadas adecuadamente en la muestra. Cabe resaltar que, una muestra representativa no es aquella que se parece a la población, de tal forma que las categorías aparecen con las mismas proporciones que en la población dado que, en algunas ocasiones es fundamental sobre-representar algunas categorías o incluso seleccionar unidades con probabilidades desiguales para poderlas medir con precisión *(Tillé, 2006)*

* *¿Qué es el factor de expansión?*

Según *Guitiérrez (2016)* el factor de expansión es el número de elementos menos uno de la población (no incluidos en la muestra) representados por el elemento incluido. También se conoce como el inverso de la probabilidad de inclusión.

Dadas las definiciones hechas anteriormente, una encuesta de hogares requiere el análisis de todas las variables que dispuestas en la encuesta. Este proceso debe ser llevado a cabo por separado para asegurar la calidad y consistencia de los datos recolectados. Sin embargo, *no* vamos a adentrarnos en el análisis de las variables en la muestra, porque los datos muestrales no son de interés para el investigador. El interés se centra en lo que suceda a nivel poblacional y este análisis se debe abordar desde la teoría del muestreo.

## 3.9 **¡Observación importante!**

Los siguientes resultados no tienen interpretación poblacional y se realizan con el único propósito de ilustrar el manejo de las bases de datos de las encuestas.

## 3.10 Medias y totales

La función summarise permite conocer el total de los ingresos en la base de datos y la media de los ingresos sobre los respondientes.

data2 %>% summarise(total.ing = sum(ingcorte),  
 media.ing = mean(ingcorte))

## total.ing media.ing  
## 1 422286293 1183.193

También se puede calcular medias de manera agrupada. Particularmente, si se desea calcular la media de los ingresos por área se hace de la siguiente manera:

data2 %>% group\_by(area\_ee) %>%  
 summarise(n = n(),  
 media = mean(ingcorte))

## # A tibble: 2 × 3  
## area\_ee n media  
## <fct> <int> <dbl>  
## 1 Area urbana 304564 1278.  
## 2 Area rural 52340 634.

Si ahora el análisis de los ingresos se desea hacer por sexo se realiza de la siguiente manera:

data2 %>% group\_by(sexo) %>%  
 summarise(n = n(),  
 media = mean(ingcorte))

## # A tibble: 2 × 3  
## sexo n media  
## <fct> <int> <dbl>  
## 1 Hombre 173223 1192.  
## 2 Mujer 183681 1174.

## 3.11 Medianas y percentiles

La función summarise también permite conocer algunas medidas de localización de los ingresos en la base de datos.

data2 %>% summarise(mediana = median(ingcorte),  
 decil1 = quantile(ingcorte, 0.1),  
 decil9 = quantile(ingcorte, 0.9),  
 rangodecil = decil9 - decil1)

## mediana decil1 decil9 rangodecil  
## 1 732.8571 244.8872 2308.5 2063.613

## 3.12 Varianza, desviación estándar y rangos

Utilizando la función summarise podemos conocer también el comportamiento variacional de los ingresos sobre los respondientes.

data2 %>% summarise(varianza = var(ingcorte),  
 desv = sd(ingcorte))

## varianza desv  
## 1 3407496 1845.94

data2 %>% summarise(mini = min(ingcorte),  
 maxi = max(ingcorte),  
 rango = maxi - mini,  
 rangoiq = IQR(ingcorte))

## mini maxi rango rangoiq  
## 1 0 171000 171000 869.8312

Ahora bien, si se desea realizar el cálculo de la media, la desviación estándar y el rango de los ingresos por hogares, se realiza de la siguiente manera:

data2 %>% filter(paren\_ee == "Jefe") %>%  
 group\_by(sexoj) %>%  
 summarise(n = n(),  
 media = mean(ingcorte),  
 desv = sd(ingcorte),  
 rangoiq = IQR(ingcorte))

## # A tibble: 2 × 5  
## sexoj n media desv rangoiq  
## <fct> <int> <dbl> <dbl> <dbl>  
## 1 Jefe hombre 70154 1456. 2325. 1026.  
## 2 Jefa mujer 47785 1334. 2076. 943.

y por condicción de ocupación se realizaría:

data2 %>% group\_by(condact) %>%  
 summarise(n = n(),  
 media = mean(ingcorte),  
 desv = sd(ingcorte),  
 rangoiq = IQR(ingcorte))

## # A tibble: 4 × 5  
## condact n media desv rangoiq  
## <int> <int> <dbl> <dbl> <dbl>  
## 1 -1 22937 764. 1136. 524.  
## 2 1 165325 1458. 2191. 1028.  
## 3 2 17896 695. 949. 497.  
## 4 3 150746 1003. 1527. 706.

a nivel de hogar:

data2 %>% filter(paren\_ee == "Jefe") %>%   
 group\_by(condact) %>%  
 summarise(n = n(),  
 media = mean(ingcorte),  
 desv = sd(ingcorte),  
 rangoiq = IQR(ingcorte))

## # A tibble: 3 × 5  
## condact n media desv rangoiq  
## <int> <int> <dbl> <dbl> <dbl>  
## 1 1 77852 1526. 2459. 1096.  
## 2 2 4469 535. 778. 441.  
## 3 3 35618 1256. 1730. 880.

Si se desea hacer un descriptivo a nivel de hogar para el ingreso se realizaría de la siguiente manera:

data2 %>% filter(paren\_ee == "Jefe") %>%   
 group\_by(pobreza) %>%  
 summarise(n = n(),  
 media = mean(ingcorte),  
 desv = sd(ingcorte),  
 rangoiq = IQR(ingcorte))

## # A tibble: 3 × 5  
## pobreza n media desv rangoiq  
## <fct> <int> <dbl> <dbl> <dbl>  
## 1 Pobreza extrema 3918 79.9 52.7 88.9  
## 2 Pobreza no extrema 13688 269. 62.5 107.   
## 3 Fuera de la pobreza 100333 1614. 2355. 1055.

# 4 Análisis de las variables continuas en encuestas de hogares

Los desarrollos estadísticos están en permanente evolución, surgiendo nuevas metodologías y desarrollando nuevos enfoques en el análisis de encuestas. Estos desarrollos parten de la academia, luego son adoptados por las empresas (privadas o estatales) y entidades estatales, las cuales crean la necesidad que estos desarrollos sean incluidos en software estadísticos licenciados, proceso que puede llevar mucho tiempo.

Algunos investigadores para acortar los tiempos y poner al servicio de la comunidad sus descubrimientos y desarrollos, hacen la implementación de sus metodologías en paquetes estadísticos de código abierto como **R** o **Python**. Teniendo **R** un mayor número de desarrollos en el procesamiento de las encuestas.

Como se ha venido mencionando anteriormente, dentro del software *R* se disponen de múltiples librerías para el procesamiento de encuestas, estas varían dependiendo del enfoque de programación desarrollado por el autor o la necesidad que se busque suplir. Como es el objetivo de este libro y como se ha venido trabajando en los capítulos anteriores nos centraremos en las libreria survey y srvyr. Se incluiran más librerías de acuerdo a las necesidades que se presenten.

## 4.1 Lectura de bases de datos y definición del diseño muestral

Las bases de datos (tablas de datos) pueden estar disponibles en una variedad de formatos (.xlsx, .dat, .csv, .sav, .txt, etc.). Sin embargo, por experiencia es recomendable leer cualquiera de estos formatos y proceder inmediatamente a guardarlos en un archivo de extensión .rds, que es nativa de R. Los archivos .rds permiten almacenar cualquier objeto o información en R, como marcos de datos, vectores, matrices, listas, entre otros. Se caracterizan por su flexibilidad y compatibilidad completa con R, lo que facilita su reutilización en análisis posteriores.

Por otro lado, existen otros tipos de archivos propios de R, como .Rdata. La diferencia principal es que los archivos .rds contienen un solo objeto, mientras que .Rdata puede almacenar múltiples objetos. Por ello, se recomienda trabajar preferentemente con archivos .rds.

Para ejemplificar la sintaxis que se utilizará en R, se tomará la misma base del capítulo anterior, que contiene una muestra de 2,427 registros y proviene de un muestreo complejo. A continuación, se muestra cómo cargar un archivo con extensión .rds:

library(tidyverse)  
  
encuesta <- readRDS("Data/encuesta.rds")  
head(encuesta)

## HHID Stratum NIh nIh dI PersonID PSU Zone Sex Age MaritalST  
## 1 idHH00031 idStrt001 9 2 4.5 idPer01 PSU0003 Rural Male 68 Married  
## 2 idHH00031 idStrt001 9 2 4.5 idPer02 PSU0003 Rural Female 56 Married  
## 3 idHH00031 idStrt001 9 2 4.5 idPer03 PSU0003 Rural Female 24 Married  
## 4 idHH00031 idStrt001 9 2 4.5 idPer04 PSU0003 Rural Male 26 Married  
## 5 idHH00031 idStrt001 9 2 4.5 idPer05 PSU0003 Rural Female 3 <NA>  
## 6 idHH00041 idStrt001 9 2 4.5 idPer01 PSU0003 Rural Female 61 Widowed  
## Income Expenditure Employment Poverty dki dk wk Region CatAge  
## 1 409.87 346.34 Employed NotPoor 8 36 34.50371 Norte Más de 60  
## 2 409.87 346.34 Employed NotPoor 8 36 33.63761 Norte 46-60  
## 3 409.87 346.34 Employed NotPoor 8 36 33.63761 Norte 16-30  
## 4 409.87 346.34 Employed NotPoor 8 36 34.50371 Norte 16-30  
## 5 409.87 346.34 <NA> NotPoor 8 36 33.63761 Norte 0-5  
## 6 823.75 392.24 Employed NotPoor 8 36 33.63761 Norte Más de 60

Según Naciones Unidas (2005, sec. 7.8), es fundamental que la estructura de los diseños de muestreo complejos se tenga en cuenta en el proceso de inferencia al estimar estadísticas oficiales basadas en encuestas de hogares. Ignorar este aspecto puede generar estimaciones sesgadas y errores de muestreo subestimados. En este contexto, los programas estadísticos ofrecen funcionalidades clave para el manejo de datos provenientes de este tipo de diseños (Heeringa, West y Berglund, 2017, Apéndice A).

De manera general, herramientas estadísticas como **R, Stata, SAS y SPSS** cuentan con módulos y bibliotecas que optimizan la estimación de varianzas en muestras complejas, incorporando métodos de replicación para varianzas basadas en el diseño. Mientras que R es de acceso libre, los otros programas requieren licencias de pago. Estas plataformas permiten calcular estadísticas descriptivas (medias, totales, proporciones, percentiles, razones) y ajustar modelos de regresión considerando la estructura del diseño de la encuesta. Los programas especializados generan automáticamente el efecto del diseño, facilitando la interpretación de la variabilidad de las estimaciones.

El uso de estos paquetes implica que el usuario suministre información clave del diseño muestral, como factores de expansión, estratificación e identificadores de conglomerados. A continuación, se ofrece un resumen general, aunque no exhaustivo, de las funciones y posibilidades que brindan los principales programas estadísticos para el análisis de encuestas complejas.

Una vez cargada la muestra de hogares en R, el siguiente paso es definir el diseño muestral del cual proviene dicha muestra. Para ello se utilizará el paquete srvyr, que como se mencionó anteriormente, surge como un complemento de survey. Estas librerías permiten definir objetos tipo **survey.design**, a los que se aplican las funciones de estimación y análisis de encuestas, y que pueden ser combinadas con la programación de tubería (%>%) del paquete tidyverse.

A manera de ejemplificar los conceptos mencionados, a continuación se define en R el diseño de muestreo del cual proviene la muestra contenida en el objeto **encuesta**:

options(survey.lonely.psu = "adjust")   
  
library(srvyr)  
  
diseno <- encuesta %>%   
 as\_survey\_design(  
 strata = Stratum,   
 ids = PSU,   
 weights = wk,   
 nest = T)

En el código anterior se puede observar que, en primera instancia se debe definir la base de datos en la cual se encuentra la muestra seleccionada. Seguido de eso, se debe definir el tipo de objeto en R con el cual se trabajará, para nuestro caso, será un objeto *survey\_design* el cual se define usando la función *as\_survey\_design*. ahora bien, una vez definido el tipo de objeto se procede a definir los parámetros del diseño definido. Para este caso fue un diseño de muestreo estratificado y en varias etapas. Estos argumentos se definen dentro de la función *as\_survey\_design* como sigue. Para definir los estratos de utiliza el argumento *strata* y se define en qué columna están los estratos en mi base de datos. Ahora bien, para definir las UPM´s, en el argumento *ids* se definen la columna donde se encuntran los conglomerados seleccionados en la primera etapa. También, se definen los pesos de muestreo en el argumento *weights* y, por último, con el argumento *nest=T* se define que las UPM´s están dentro de los estratos.

## 4.2 Análisis gráfico: Histogramas y Boxplot

Una vez cargada la muestra a R y definido el diseño muestral del cual proviene se pueden hacer los primeros análisis. Como es natural, se inician con análisis gráficos. A continuación, se muestran los códigos computacionales con los cuales se hacen histogramas en R para la variable ingresos teniendo en cuenta el diseño muestral y los factores de expansión haciendo uso la función svyhist de la librería survey.

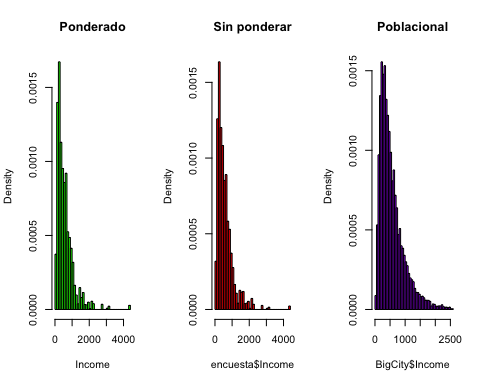
library(survey)  
library(srvyr)  
svyhist(  
 ~ Income ,  
 diseno,  
 main = "Ingresos por hogar",  
 col = "grey80",  
 xlab = "Ingreso",  
 probability = FALSE  
)



Como se pudo observar en el código anterior, para generar un histograma teniendo en cuenta el diseño muestral se usó la función svyhist. En primer lugar, se definió la variable a graficar, que para nuestro caso fue *Income*. Seguido, se define el diseño muestral utilizado en la encuesta. Luego, se definen los argumentos relacionados con la estética del gráfico como lo son: el título principal (*main*), el color (*col*) y el título horizontal (*xlab*). Finalmente, se define si el histograma es de frecuencias o probabilidades con el argumento *probability*. Para este ejemplo, se tomó la opción *probability = False* indicando que es un histograma de frecuencias.

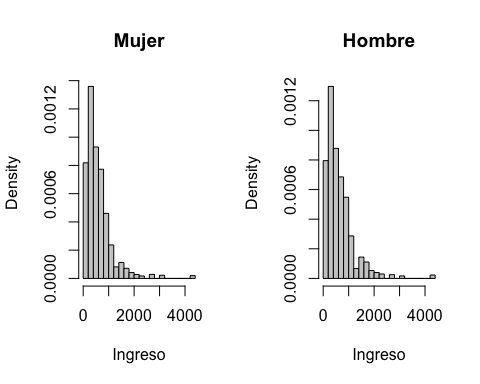
Una pregunta que surge de manera natural es ¿cuál es la diferencia entre los histogramas sin usar los factores de expansión y utilizándolo? A continuación, se generan 3 histogramas, en el primero se grafica la variable ingreso utilizando los factores de expansión, en el segundo se grafica la misma variable sin usar los factores de expansión y en el tercero, se hace el gráfico poblacional.

library(survey)  
data("BigCity", package = "TeachingSampling")  
par(mfrow = c(1,3))  
  
svyhist(~ Income,  
 diseno, main = "Ponderado",  
 col = "green", breaks = 50)  
  
hist( encuesta$Income,  
 main = "Sin ponderar",  
 col = "red", prob = TRUE, breaks = 50)  
  
hist(BigCity$Income,  
 main = "Poblacional",  
 col = "purple", prob = TRUE,  
 xlim = c(0, 2500), breaks = 500)



Uno de los análisis gráficos más comunes que se realizan ene encuestas de hogares están relacionados con subgrupos geográficos como lo son las zonas (urbano - rural) o también realizar desagregaciones temáticas como lo son por sexo (hombre mujer). A continuación, se muestra la sintaxis en R como se realizan histogramas para hombres y mujeres mayores de 18 años:

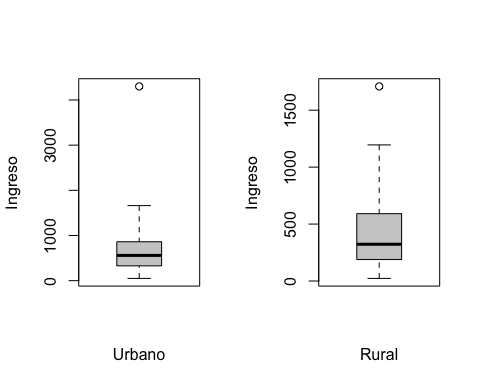
sub\_Mujer <- diseno %>% filter(Sex == "Female")  
sub\_Hombre <- diseno %>% filter(Sex == "Male")  
  
par(mfrow = c(1,2))  
  
svyhist(  
 ~ Income ,  
 design = subset(sub\_Mujer, Age >= 18),  
 main = "Mujer",  
 breaks = 30,  
 col = "grey80",  
 xlab = "Ingreso")  
  
svyhist(  
 ~ Income ,  
 design = subset(sub\_Hombre, Age >= 18),  
 main = "Hombre",  
 breaks = 30,  
 col = "grey80",  
 xlab = "Ingreso")



Como se puede observar, los argumentos utilizando para realizar los gráficos son los mismo que se utilizaron y ejemplificaron anteriormente. Cabe notar que la función *subset* permite hacer un subconjunto de la población, que para nuetro caso son aquellos hombres y mujeres mayores o iguales a 18 años.

Si el objetivo ahora es realizar análisis de localización y variablidad, por ejemplo, graficar Bloxplot teniendo en cuenta los factores de expansión, a continuación, se muestran las sintaxis de como realizarlo en R.

sub\_Urbano <- diseno %>% filter(Zone == "Urban")  
sub\_Rural <- diseno %>% filter(Zone == "Rural")  
  
par(mfrow = c(1,2))  
svyboxplot(  
 Income~1 ,  
 sub\_Urbano,  
 col = "grey80",  
 ylab = "Ingreso",  
 xlab = "Urbano")  
  
svyboxplot(  
 Income ~ 1 ,  
 sub\_Rural,  
 col = "grey80",  
 ylab = "Ingreso",  
 xlab = "Rural")



Los argumentos usados en la función *svyboxplot* para generar el gráfico son muy similares a los usados en la función *svyhist*. Algo a recalcar el los argumentos de esta función es que el símbolo “Income ~ 1” hace referencia a que todas las personas pertenecen a un solo grupo que puede ser urbano o rural dependiendo del caso y por eso se requiere indicarle a R esa restricción, lo cual se hace con el símbolo “~1”.

## 4.3 Estimación de totales, medias y razones

Al trabajar con encuestas de hogares, el análisis de datos numéricos implica con frecuencia calcular estadísticas descriptivas como medias, totales y razones, ya que estas permiten sintetizar las principales características de la población y sirven de base para la toma de decisiones. Dichas estimaciones pueden calcularse para la población en su conjunto o para subgrupos específicos, según los objetivos de la investigación. Tal como destacan Heeringa, West y Berglund (2017), el cálculo de totales y medias poblacionales, junto con sus varianzas, ha sido esencial para el desarrollo de la teoría del muestreo probabilístico y la interpretación adecuada de los resultados de encuestas de hogares.

### 4.3.1 Estimación puntual

Una vez exploradas las tendencias de las variables continuas mediante análisis gráfico, el siguiente paso consiste en obtener las estimaciones puntuales de los parámetros medidos. Estas estimaciones pueden calcularse de forma general o desagregada por niveles de análisis, dependiendo de las necesidades de la investigación. En el contexto de encuestas de hogares, las estimaciones puntuales comprenden el cálculo de totales, promedios, razones y otras medidas agregadas.

Heeringa et al. (2017) señalan que la estimación del total o promedio de una población y su varianza muestral es fundamental en la teoría del muestreo probabilístico, ya que permite obtener valores precisos sobre la situación de los hogares estudiados, facilitando la toma de decisiones informadas en políticas públicas.

### 4.3.2 Estimación de totales e intervalos de confianza

Una vez definido el diseño muestral (como se hizo en la sección anterior), se procede a realizar los procesos de estimación de los parámetros de interés. Para efectos de este texto, se iniciará con la estimación del total de los ingresos de los hogares.

En su mayoría, los paquetes estadísticos actuales no implementan técnicas avanzadas como estimadores generales de regresión (GREG) o métodos de calibración. Sin embargo, Valliant et al. (2000) desarrollaron una librería en *S-plus* que permite realizar estos procedimientos de estimación, los cuales también pueden implementarse en R (Valliant et al., 2013).

Para la estimación de totales con diseños muestrales complejos que incluyen estratificación () y muestreo por conglomerados (cuyos conglomerados están dentro del estrato , indexados por ), el estimador del total se puede expresar como:

El estimador insesgado de la varianza para este total es:

La determinación de los totales poblacionales constituye uno de los pilares del análisis de encuestas. Tanto las medias como las proporciones y las razones derivan de los totales. Un total se define como la suma de una variable específica (por ejemplo, ingreso o gasto) a nivel de toda la población. Para estimar el ingreso total de todos los hogares de un país, se combinan los datos de la muestra aplicando los pesos muestrales que reflejan el diseño y aseguran representatividad.

En el caso de variables numéricas simples, las estimaciones básicas son los totales y medias, mientras que las razones permiten establecer comparaciones entre dos variables numéricas. Estos cálculos pueden realizarse para toda la población o de manera desagregada por dominios de estudio, dependiendo de las preguntas de investigación.

Para encuestas con estratificación () y submuestreo en las UPM (ubicadas dentro de cada estrato , identificadas por ), el total poblacional se estima mediante:

Cuando se cuenta con respuesta completa, la varianza de puede calcularse usando el estimador de Ultimate Cluster.

El intervalo de confianza de nivel para el total poblacional se calcula como:

A medida que los grados de libertad aumentan, la distribución de Student tiende a la normal, lo que explica por qué muchas Oficinas Nacionales de Estadística (ONE) utilizan esta aproximación para reportar intervalos de confianza. No obstante, es importante considerar que esta aproximación puede ser menos fiable cuando el tamaño de la muestra es reducido, aunque suele ofrecer buenos resultados en encuestas de hogares extensas.

Como se puede observar, calcular la estimación del total y su varianza estimada es complejo.

#### 4.3.2.1 Enfoques para la estimación de la varianza

Tal como se mencionó anteriormente, al trabajar con encuestas de hogares es fundamental proporcionar no solo estimaciones puntuales, sino también cuantificar la incertidumbre asociada a dichas estimaciones. Comprender y estimar esta incertidumbre constituye una parte crítica del análisis de los datos provenientes de encuestas de hogares. Mediante la aplicación de métodos apropiados, los usuarios pueden medir la precisión de sus estimaciones.

Existen diversos métodos para estimar dicha precisión y, con el apoyo de software moderno, estos enfoques pueden implementarse de manera eficiente para respaldar análisis rigurosos y significativos. Entre los principales métodos se encuentran:

* **Ecuaciones de estimación:** ofrecen un marco flexible para estimar totales, medias, razones y otros parámetros, así como sus varianzas correspondientes, integrando una idea unificadora de la teoría de muestreo (Binder, 1983).
* **Linealización de Taylor:** consiste en aproximar estadísticas no lineales complejas mediante expresiones lineales y posteriormente estimar la varianza de esta cantidad aproximada.
* **Método del Clúster Último:** utilizado con frecuencia en encuestas que emplean muestreo estratificado en múltiples etapas; se basa en calcular la varianza a partir de las diferencias entre las estimaciones obtenidas a nivel de las unidades primarias de muestreo (PSU). Suele combinarse con la Linealización de Taylor para estimar la varianza de estadísticas no lineales, como medias o razones.
* **Bootstrap y otros métodos de replicación:** se fundamentan en tomar repetidas submuestras del conjunto de datos observado, calcular estimaciones para cada réplica y luego utilizar la variabilidad entre estas estimaciones replicadas para inferir la varianza del estimador principal.

#### Ecuaciones de estimación y linealización de Taylor

Como se mencionó en el apartado anterior, uno de los enfoques más utilizados para cuantificar la incertidumbre en encuestas de hogares se basa en la formulación de **ecuaciones de estimación** y en la aplicación de la **linealización de Taylor**. Estos métodos proporcionan un marco general que permite definir los parámetros de interés, obtener sus estimadores muestrales y, a partir de ellos, aproximar sus varianzas.

Muchos parámetros poblacionales pueden expresarse como soluciones de ecuaciones de estimación que involucran totales poblacionales. Aunque los detalles técnicos pueden ser complejos, la idea fundamental es que los mismos principios utilizados para estimar totales pueden aplicarse también para la estimación de varianzas. Este marco general hace que el método sea sencillo y versátil, permitiendo una implementación eficiente en software especializado.

Una ecuación de estimación poblacional genérica se expresa como:

donde es una función de estimación evaluada para la unidad y representa el parámetro poblacional de interés. Estas ecuaciones proporcionan un marco general para definir y calcular diversos parámetros de la población, como totales, medias y razones.

* **Para el total poblacional:** . La ecuación de estimación es , cuya solución es .
* **Para la media poblacional:** . La ecuación es , cuya solución es .
* **Para razones de totales:** . La ecuación conduce a la razón poblacional .

La idea de definir parámetros poblacionales como soluciones de ecuaciones de estimación a nivel de población conduce naturalmente a un método general para obtener los **estimadores muestrales**. En este caso, se utilizan ecuaciones de la forma:

donde son los pesos de diseño y la función de estimación evaluada para cada unidad de la muestra. Bajo un muestreo probabilístico y asumiendo respuesta completa, la suma muestral es insesgada respecto a su análoga poblacional, lo que garantiza que las soluciones de estas ecuaciones sean estimadores consistentes de los parámetros poblacionales.

La **linealización de Taylor** constituye un complemento natural a este marco, ya que permite aproximar la varianza de estimadores no lineales. El procedimiento consiste en aplicar una expansión de Taylor de primer orden alrededor del parámetro estimado, con el fin de reemplazar el estimador no lineal por una expresión lineal. De esta manera, se facilita el cálculo de varianzas en situaciones donde no existen fórmulas exactas o su derivación resulta demasiado compleja.

Un estimador consistente de la varianza, derivado mediante linealización de Taylor para soluciones de ecuaciones de estimación muestrales, puede expresarse como:

donde .

Este resultado muestra cómo la linealización de Taylor convierte la estimación de varianzas de parámetros complejos en un problema de estimación de totales, lo que explica su amplia adopción en software especializado para el análisis de encuestas.

#### Ultimate Cluster

El método del Ultimate Cluster constituye un enfoque directo y robusto para estimar la varianza de totales en encuestas que emplean diseños de muestreo por conglomerados estratificados en múltiples etapas. Propuesto por Hansen, Hurwitz y Madow (1953), este método simplifica la complejidad de los diseños multinivel al centrarse únicamente en la variación entre las Unidades Primarias de Muestreo (PSU). Se asume que, dentro de cada estrato de muestreo, las PSU fueron seleccionadas de manera independiente con reemplazo (posiblemente con probabilidades desiguales), aunque en la práctica la selección suele realizarse sin reemplazo.

El método se basa en la variación entre las estadísticas calculadas a nivel de PSU. Cuando se aplica correctamente, refleja implícitamente cualquier submuestreo realizado dentro de las PSU, permitiendo estimaciones de varianza más simples pero confiables. Es especialmente útil en diseños complejos que incluyen estratificación y probabilidades desiguales de selección tanto de PSU como de unidades de niveles inferiores (hogares e individuos).

Los requisitos para aplicar este método son:

* Disponibilidad de estimaciones insesgadas de totales para las variables de interés en cada PSU muestreada.
* Al menos dos PSU muestreadas por estrato, si la muestra se estratifica en la primera etapa.
* Información completa sobre PSU, estratos y pesos en el conjunto de datos de la encuesta.

Considere un diseño de muestreo en múltiples etapas donde se seleccionan PSU en el estrato , . Sea

una estimación del total poblacional de la PSU en el estrato . Un estimador insesgado del total poblacional se expresa como

El estimador Ultimate Cluster de la varianza correspondiente se calcula mediante:

Para más detalles, véase Hansen, Hurwitz y Madow (1953, vol. I, p. 257) o Wolter (2007).

Aunque originalmente se diseñó para calcular varianzas de estimadores de totales, el método puede combinarse con linealización de Taylor o ecuaciones de estimación para derivar varianzas de otros parámetros poblacionales formulables como soluciones de ecuaciones de estimación. Esta flexibilidad hace que el método sea aplicable a diversos contextos de análisis de encuestas de hogares.

Un supuesto clave es que, dentro de cada estrato, las PSU se eligen de forma independiente y con reemplazo. En la práctica, la mayoría de las encuestas selecciona PSU sin reemplazo, generando diseños más eficientes. Así, las varianzas calculadas bajo la hipótesis de independencia constituyen aproximaciones de las verdaderas varianzas de muestreo. Cuando la fracción muestral es pequeña (por ejemplo, <5 %), estas aproximaciones suelen ser suficientemente precisas para su uso por oficinas nacionales de estadística o analistas secundarios.

El método Ultimate Cluster destaca por su simplicidad y robustez, lo que lo hace muy atractivo en la práctica. Aunque los métodos más sofisticados que consideran todas las etapas del diseño pueden ofrecer estimaciones de varianza ligeramente más precisas, su aplicación requiere información más detallada y mayor complejidad computacional. Por el contrario, el método Ultimate Cluster proporciona una aproximación confiable y eficiente, especialmente útil al estimar totales o medias en encuestas de hogares. Para un análisis detallado sobre la precisión de esta aproximación y posibles alternativas, véase Särndal, Swensson y Wretman (1992, p. 153).

#### Bootstrap

En muchos casos, los microdatos de encuestas públicas omiten información esencial del diseño, como identificadores de estratos o de unidades primarias de muestreo (UPM), para proteger la confidencialidad de los encuestados. Esta omisión limita la capacidad de los usuarios para calcular varianzas válidas. En tales situaciones, se recomienda que las oficinas nacionales de estadística (NSO) proporcionen pesos de replicación, lo que permite a los analistas estimar errores estándar de manera correcta. Sin estos datos, los usuarios secundarios no pueden reproducir los errores estándar publicados ni considerar adecuadamente el diseño complejo de la encuesta.

Los métodos de replicación estiman la varianza generando subconjuntos de la muestra original, calculando estimaciones para cada uno y utilizando la variabilidad observada entre estas estimaciones para aproximar la varianza del estimador principal. Son particularmente útiles cuando no se dispone de información sobre estratos o UPM, situación en la que no se puede aplicar el método de Ultimate Cluster.

El Bootstrap es una herramienta de replicación robusta y versátil. Originalmente introducido por Efron (1979) para datos que no provenían de encuestas, su adaptación más usada para encuestas de hogares es el Bootstrap de Reescalamiento Rao-Wu-Yue (Rao, Wu y Yue, 1992). Este método se ajusta de manera óptima a diseños de muestreo estratificados y multietápicos, y es ampliamente empleado para la estimación de varianzas en encuestas complejas.

El procedimiento consiste en generar muchas réplicas de la muestra original, simulando extracciones repetidas de la población. Cada réplica se construye mediante la creación de columnas adicionales de pesos de replicación en la base de datos, siguiendo este proceso:

* Para cada estrato, se seleccionan aleatoriamente las UPM con reemplazo; algunas pueden repetirse y otras no aparecer. Cada UPM elegida se incorpora con todas sus observaciones. Si el tamaño de la muestra de primera etapa en el estrato es mayor que dos (), el número de UPM seleccionadas por réplica es .
* Este proceso se repite muchas veces, habitualmente cientos, generando un gran número de réplicas. La cantidad de veces que una UPM del estrato aparece en la réplica se denota , variando entre 0 y .
* A partir de cada réplica se calculan nuevos pesos bootstrap para todas las unidades, reflejando cuántas veces fue seleccionada su UPM. El peso de la unidad en la réplica se calcula como:

Si los pesos originales incluyen ajustes por no respuesta o calibración, estos deben aplicarse también a cada conjunto de pesos bootstrap.

Cuando la NSO proporciona únicamente pesos de replicación Bootstrap, los analistas pueden estimar errores estándar correctamente, aun sin identificadores de estratos o UPM. Para cada réplica , se calcula el parámetro de interés usando los pesos bootstrap . La varianza del estimador original se aproxima mediante la variabilidad entre todas las réplicas:

Este enfoque asegura que la dispersión entre réplicas capture fielmente la incertidumbre del parámetro.

El Bootstrap ofrece múltiples ventajas. A pesar de requerir un mayor procesamiento computacional, es eficaz para diseños de encuesta complejos y permite estimar parámetros difíciles de calcular con métodos tradicionales, como medianas u otras estadísticas no lineales. Es especialmente útil para analistas que trabajan con bases de datos sin identificadores de estratos y UPM, pero con pesos de replicación.

La simplicidad del método facilita su aplicación incluso sin software estadístico especializado. Sin embargo, la mayoría de los paquetes estadísticos modernos ya incluyen procedimientos para aplicar Bootstrap y calcular varianzas, ampliando su disponibilidad y robustez. No obstante, su uso no es recomendable en encuestas repetidas con muestras superpuestas ni en situaciones con fracciones de muestreo grandes y tamaños de muestra pequeños (Bruch, 2011).

En este ejemplo se muestra cómo estimar totales y sus intervalos de confianza para diferentes variables de interés en R, utilizando dos funciones de la librería survey: la función svytotal para calcular los totales y la función confint para obtener los intervalos de confianza. A continuación, se presentan los códigos correspondientes:

total\_Ingresos<- svytotal(~Income, diseno, deff=T, )  
total\_Ingresos

## total SE DEff  
## Income 85793667 4778674 11

confint(total\_Ingresos, level = 0.95)

## 2.5 % 97.5 %  
## Income 76427637 95159697

Los argumentos que utiliza de la función svytotal con muy sencillos. Para el ejemplo, se le introduce primero la variable en la cual está la información que se desea estimar (Income). Posterior a esto, se introduce el diseño muestral del cual proviene la muestra y, por último, se indica si desea que se reporte el deff de la estimación o no.

Por otro lado, para el cálculo del intervalo de confianza, lo único que requiere es indicarle a la función confint el estimador y la confianza requerida.

Paras seguir ilustrando el uso de la función svytotal y de confint, estimemos el total de gastos de los hogares, pero ahora el intervalo de confianza se calculará al 90% de confianza. Los siguientes códigos realizan las estimaciones:

total\_gastos<- svytotal (~Expenditure, diseno, deff=T)  
total\_gastos

## total SE DEff  
## Expenditure 55677504 2604138 10.222

confint(total\_gastos, level = 0.9)

## 5 % 95 %  
## Expenditure 51394077 59960931

Si el objetivo ahora es estimar el total de los ingreso de los hogares pero discriminado por sexo, se utilizará ahora la función cascadede la libraría srvyr, la cual permite agregar la suma de las categorías al final la tabla. También se utilizará la función group\_by la cual permite obtener resultados agrupados por los niveles de interés.

diseno %>% group\_by(Sex) %>%  
 cascade(Total = survey\_total(  
 Income, level = 0.95,  
 vartype = c("se", "ci")),  
 .fill = "Total ingreso")

## # A tibble: 3 × 5  
## Sex Total Total\_se Total\_low Total\_upp  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Female 44153820. 2324452. 39551172. 48756467.  
## 2 Male 41639847. 2870194. 35956576. 47323118.  
## 3 Total ingreso 85793667. 4778674. 76331414. 95255920.

Como se pudo observar en lo códigos anteriores, otra forma de obtener las estimaciones del total, su desviación estándar y el intervalo de confianza es usando el argumento vartype e indicándole las opciones “se”, “ci” respectivamente.

### 4.3.3 Estimación de la media e intervalo de confianza

La estimación de la media poblacional es un parámetro muy importante en las encuestas de hogares, dado que, por ejemplo, uno de los indicadores trazadores en este tipo de encuestas son los ingresos medios por hogar. Además, este tipo de parámetros no permiten describir y analizar las tendencias centrales de estas variables en poblaciones de interés. Según **Gutiérrez (2016)** un estimador de la media poblacional se puede escribir como una razón no lineal de dos totales de población finitas estimados como sigue:

Como una observación tenga en cuenta que, si es una variable binaria, la media ponderada estima la proporción de la población. Por otro lado, como no es una estadística lineal, no existe una fórmula cerrada para la varianza de este estimador. Es por lo anterior que, se deben recurrir a usar métodos de remuestreo o series de Taylor. Para este caso en particular, usando series de Taylor el estimador insesgado de la varianza para este estimador es:

Como se puede observar, el cálculo de la estimación de la varianza tiene componentes complejos de calcular de manera analítica, como la covarianza entre el total estimado y el tamaño poblacional estimado. Sin embargo, R tiene funciones que incorpora estos estimadores. A continuación, se presenta la sintaxis para hacer dichos cálculos.

Media\_ingresos<- svymean(~Income, diseno, deff=T)  
Media\_ingresos

## mean SE DEff  
## Income 570.945 28.478 8.8211

confint(Media\_ingresos, level = 0.95)

## 2.5 % 97.5 %  
## Income 515.1299 626.7607

Como se puede observar, los argumentos que utiliza la función svymean para realizar la estimación de la media de los ingresos de los hogares y la desviación estándar estimada del estimador son similares a los utilizando con la función svytotal. Similarmente ocurre con el intervalo de confianza.

Por otro lado, tal como se realizó con el total, a manera de ejemplo, se estima la media de los gastos en los hogares como sigue a continuación:

Media\_gastos<- svymean (~Expenditure, diseno, deff=T)  
Media\_gastos

## mean SE DEff  
## Expenditure 370.526 13.294 6.0156

confint(Media\_gastos)

## 2.5 % 97.5 %  
## Expenditure 344.4697 396.5829

También se pueden realizar estimaciones de la media por subgrupos siguiendo el mismo esquema mostrado para la función svytotal. Particularmente, los gastos de los hogares discriminados por sexo es:

diseno %>% group\_by(Sex) %>%  
 cascade(  
 Media = survey\_mean(  
 Expenditure, level = 0.95,  
 vartype = c("se", "ci")),  
 .fill = "El gasto medio" ) %>%  
 arrange(desc(Sex))

## # A tibble: 3 × 5  
## Sex Media Media\_se Media\_low Media\_upp  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Male 374. 16.1 343. 406.  
## 2 Female 367. 12.3 343. 391.  
## 3 El gasto medio 371. 13.3 344. 397.

Por zona,

diseno %>% group\_by(Zone) %>%  
 cascade(  
 Media = survey\_mean(  
 Expenditure, level = 0.95,  
 vartype = c("se", "ci")),  
 .fill = "El gasto medio")%>%  
 arrange(desc(Zone))

## # A tibble: 3 × 5  
## Zone Media Media\_se Media\_low Media\_upp  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Urban 460. 22.2 416. 504.  
## 2 Rural 274. 10.3 254. 294.  
## 3 El gasto medio 371. 13.3 344. 397.

Por sexo y zona,

diseno %>% group\_by(Zone, Sex) %>%  
 cascade(  
 Media = survey\_mean(  
 Expenditure, level = 0.95,  
 vartype = c("se", "ci")),  
 .fill = "El gasto medio") %>%  
 arrange(desc(Zone), desc(Sex)) %>%  
 data.frame()

## Zone Sex Media Media\_se Media\_low Media\_upp  
## 1 Urban Male 469.8124 26.96068 416.4276 523.1973  
## 2 Urban Female 450.8151 20.11853 410.9784 490.6518  
## 3 Urban El gasto medio 459.6162 22.20655 415.6450 503.5874  
## 4 Rural Male 275.3018 10.24848 255.0088 295.5948  
## 5 Rural Female 272.6769 11.61470 249.6786 295.6751  
## 6 Rural El gasto medio 273.9461 10.26141 253.6275 294.2647  
## 7 El gasto medio El gasto medio 370.5263 13.29444 344.2020 396.8506

Las medias o promedios poblacionales son esenciales para describir la tendencia central de una variable. Por ejemplo, el gasto promedio de los hogares es un indicador representativo del comportamiento económico de la población. Su cálculo consiste en dividir el total estimado de la variable entre el tamaño poblacional estimado, de modo que su precisión depende de la exactitud en ambos componentes.

El estimador de la media poblacional se puede expresar como:

Dado que es un estimador no lineal, su varianza exacta no puede expresarse en forma cerrada. Por ello, los paquetes estadísticos especializados en encuestas complejas emplean métodos como el remuestreo o la aproximación de Taylor, que se implementan automáticamente para facilitar su cálculo.

### 4.3.4 Estimación de medidas de dispersión y localización

En las encuestas de hogares siempre es necesario estimar medidas de dispersión de las variables estudiadas. Esto con el fin de, por ejemplo, ver qué tan disímiles son los ingresos medios de los hogares en un país determinado y con esto poder tomar acciones de política pública. Por lo anterior, es importante estudiar este parámetro en este texto. A continuación, se presenta el estimador de la desviación estándar:

Para llevar a cabo la estimación en R de la desviación estándar en encuestas de hogares, se utilizan la función survey\_var la cual se ejemplifica a continuación:

(sd\_Est <- diseno %>% group\_by(Zone) %>%  
 summarise(Sd = sqrt(  
 survey\_var(  
 Income,  
 level = 0.95,  
 vartype = c("se", "ci"),  
 ) )))

## # A tibble: 2 × 5  
## Zone Sd Sd\_se Sd\_low Sd\_upp  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Rural 310. 117. 263. 352.  
## 2 Urban 582. 285. 422. 707.

Como se pudo ver en el ejemplo anterior, se estimó la desviación estándar de los ingresos por zona reportando el error estándar en la estimación y un intervalo de confianza al 95%. Los argumentos que utiliza la función survey\_var son similares a los usados en las funciones anteriores para estimar medias y totales.

Si el interés ahora se centra en estimar la desviación estándar clasificando por sexo y zona, los códigos computacionales son los siguientes:

(sd\_Est <- diseno %>% group\_by(Zone, Sex) %>%  
 summarise(Sd = sqrt(  
 survey\_var(  
 Income,  
 level = 0.95,  
 vartype = c("se", "ci"),  
 )  
))) %>% data.frame()

## Zone Sex Sd Sd\_se Sd\_low Sd\_upp  
## 1 Rural Female 294.8683 111.6203 249.5537 334.0921  
## 2 Rural Male 325.7584 124.9643 274.2209 370.1890  
## 3 Urban Female 568.3920 286.4585 400.7312 696.8166  
## 4 Urban Male 596.7756 288.9435 436.8362 722.1194

Las medidas de posición no central (Percentiles) se diseñaron con el fin de conocer otros puntos característicos de la distribución de los datos que no son los valores centrales. Entre las medidas de posición no central más importantes están la mediana, cuartiles y percentiles. En la mayoría de las encuestas de hogares no solo estiman totales, medias y proporciones. En algunos indicadores es necesario estimar otros parámetros, por ejemplo, medianas y percentiles. Como lo menciona **Tellez et al (2015)** la mediana una medida de tendencia central la cual, a diferencia del promedio, no es fácilmente influenciada por datos atípicos y, por esto, se conoce como una medida robusta. La mediana es el valor que divide la población en dos partes iguales. Lo que implica que, la mitad de las observaciones de la característica de interés está por encima de la media y la otra mitad está por debajo.

Por otro lado, la estimación de percentiles de ingresos en un país determinado puede definir el inicio de una política pública. por ejemplo, poner a tributar aquellas personas naturales que son el 10% más alto de la distribución de los ingresos o por el contrario, generar subsidios de transporte a aquellas familias que están en el 15% inferior de la distribución de los ingresos.

La estimación de cuantiles **(Loomis et al., 2005)** se basa en los resultados relacionados con el estimador ponderado para totales, empleando una estimación de la función de distribución (CDF, por sus siglas en inglés) acumulada de la población. Específicamente, la CDF para una variable y en una población finita dada de tamaño se define de la siguiente manera:

Donde, es una variable indicadora la cual es igual a 1 si es menor o igual a un valor específico , 0 en otro caso. Un estimador de la CDF en un diseño complejo (encuesta de hogares) de tamaño está dado por:

Una vez estimada la CDF utilizando los pesos del diseño muestral, el cuantil q-ésimo de una variable es el valor más pequeño de tal que la CDF de la población es mayor o igual que . Como es bien sabido, la mediana es aquel valor donde la CDF es mayor o igual a 0.5 y, por tanto, la media estimada es aquel valor donde la estimación de CDF es mayor o igual a 0.5.

Siguiendo las recomendaciones de *Heeringa et al (2017)* para estimar cuantiles, primero se considera las estadísticas de orden que se denotan como , y encuentra el valor de tal que:

Ahora bien, la estimación del q-ésimo cuantil en un diseño de muestreo complejo está dado por:

Para la estimación de la varianza e intervalos de confianza de cuantiles, **Kovar et al. (1988)** muestra los resultados de un estudio de simulación en donde recomienda el uso de Balanced Repeated Replication (BRR) para estimarla.

Los estimadores y procedimientos antes mencionados para la estimación de percentiles y sus varianzas están implementados en R. Particularmente, la estimación de la mediana se realiza usando la función survey\_median. A continuación, se muestra la sintaxis de cómo calcular la mediana de los gastos, la desviación estándar y el intervalo de confianza al 95% de los hogares en la base de datos de ejemplo.

diseno %>% summarise(Mediana =  
 survey\_median(  
 Expenditure,  
 level = 0.95,  
 vartype = c("se", "ci"),  
 ))

## # A tibble: 1 × 4  
## Mediana Mediana\_se Mediana\_low Mediana\_upp  
## <dbl> <dbl> <dbl> <dbl>  
## 1 298. 8.83 282. 317.

Como se puede observar, los argumentos de la función survey\_median son similares a los del total y la media.

Ahora bien, al igual que con los demás parámetros, si el objetivo ahora es estimar la mediana de los gastos de los hogares, pero esta vez discriminada por zona y también por sexo, el código computacional sería el siguiente:

diseno %>% group\_by(Zone) %>%  
 summarise(Mediana =  
 survey\_median(  
 Expenditure,  
 level = 0.95,  
 vartype = c("se", "ci"),  
 ))

## # A tibble: 2 × 5  
## Zone Mediana Mediana\_se Mediana\_low Mediana\_upp  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Rural 241. 11.0 214. 258.  
## 2 Urban 381. 19.8 337. 416.

diseno %>% group\_by(Sex) %>%  
 summarise(Mediana =  
 survey\_median(  
 Expenditure,  
 level = 0.95,  
 vartype = c("se", "ci"),  
 ))

## # A tibble: 2 × 5  
## Sex Mediana Mediana\_se Mediana\_low Mediana\_upp  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Female 300. 10.5 282. 324.  
## 2 Male 297. 9.29 277. 314.

Si el objetivo ahora es estimar cuantiles, por ejemplo, el cuantil 0.25 de los gastos de los hogares, se realizaría usando la función survey\_quantile como sigue:

diseno %>%  
 summarise(  
 Q = survey\_quantile(  
 Expenditure,  
 quantiles = 0.5,  
 level = 0.95,  
 vartype = c("se", "ci"),  
 interval\_type = "score"  
 ))

## # A tibble: 1 × 4  
## Q\_q50 Q\_q50\_se Q\_q50\_low Q\_q50\_upp  
## <dbl> <dbl> <dbl> <dbl>  
## 1 298. 12.0 265. 312.

si ahora se desea estimar el cuantil 0.25 pero discriminando por sexo y por zona se realizaría como sigue:

diseno %>% group\_by(Sex) %>%  
 summarise(  
 Q = survey\_quantile(  
 Expenditure,  
 quantiles = 0.25,  
 level = 0.95,  
 vartype = c("se", "ci"),  
 interval\_type = "score"  
 ))

## # A tibble: 2 × 5  
## Sex Q\_q25 Q\_q25\_se Q\_q25\_low Q\_q25\_upp  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Female 210. 14.9 169. 228.  
## 2 Male 193. 10.4 163. 205.

diseno %>% group\_by(Zone) %>%  
 summarise(  
 Q = survey\_quantile(  
 Expenditure,  
 quantiles = 0.25,  
 level = 0.95,  
 vartype = c("se", "ci"),  
 interval\_type = "score"  
 ))

## # A tibble: 2 × 5  
## Zone Q\_q25 Q\_q25\_se Q\_q25\_low Q\_q25\_upp  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Rural 160. 4.64 145. 163.  
## 2 Urban 258. 9.05 256. 292.

## 4.4 Estimación del coeficiente de Gini en encuestas de hogares

Para iniciar esta sección tengamos en cuenta la siguiente reflexión: *Definir lo justo siempre será difícil y es algo a lo que quizá sea poco realista aspirar a conseguir. Sin embargo, si estamos un poco más conscientes de cómo la desigualdad afecta nuestra libertad y cómo se refleja en el bienestar y calidad de vida de las personas, podremos poner en contexto una discusión que tendremos cada vez más presente en el mundo y en el país.*

La desigualdad es un problema común en todos los países del mundo. En el ámbito económico, representa uno de los desafíos más relevantes para gobiernos y organismos internacionales. De hecho, Naciones Unidas lo reconoce explícitamente dentro de los Objetivos de Desarrollo Sostenible (ODS).

Entre las métricas más utilizadas para medir la desigualdad económica se encuentra el coeficiente de Gini (CG), el cual compara la distribución de ingresos observada con una distribución perfectamente equitativa. Su valor oscila entre 0 (igualdad perfecta) y 1 (máxima desigualdad). Así, un valor de implica que todos los hogares tienen el mismo nivel de ingresos, mientras que valores más altos reflejan concentraciones crecientes de ingresos en una fracción menor de la población. Este indicador es clave para:

* Evaluar los cambios en la distribución del ingreso a lo largo del tiempo.
* Comparar los niveles de desigualdad entre regiones o países.
* Contextualizar políticas públicas y sociales orientadas a la equidad.

En encuestas de hogares, el cálculo del coeficiente de Gini debe incorporar los pesos muestrales ajustados al diseño de la encuesta, que con frecuencia se normalizan para simplificar el procesamiento.

De acuerdo con Binder y Kovacevic (1995), el estimador del coeficiente de Gini puede expresarse como:

donde:

* es el peso normalizado.
* representa la función de distribución acumulada (CDF) estimada en el conglomerado del estrato .
* corresponde a la media ponderada de los ingresos.

Autores como Osier (2009) y Langel y Tillé (2013) profundizan en aspectos técnicos adicionales relacionados con la estimación de la varianza de este estimador en el contexto de encuestas complejas.

Para calcular el índice de Gini y su varianza estimada en una encuesta de hogares, R tiene cargados los procedimientos en la librería convey. A continuación, se muestra la sintaxis de cómo se realiza la estimación del índice de Gini para los hogares en la base de ejemplo de este capítulo.

library(convey)  
 diseno\_gini <- convey\_prep(diseno)  
svygini( ~Income, design = diseno\_gini) %>%  
 data.frame()

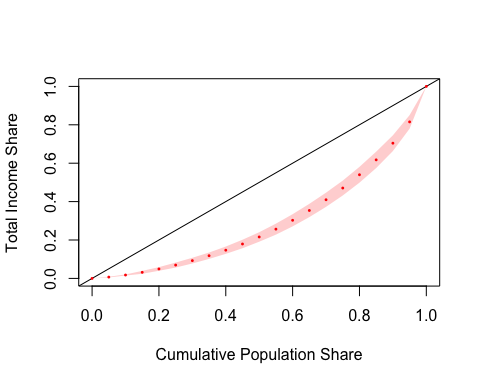
## gini Income  
## Income 0.4132757 0.0186633

En primer lugar, se carga el diseño de muestreo con la función convey\_prep. Luego, se estima el índice Gini con la función svygini. En los argumentos de esta última función se introducen la variable ingresos y el diseño muestral complejo.

Por otro lado, si el interés ahora es estimar la **curva de Lorenz**. La cual, según *Kovacevic, M. S. et. al (1997)* para una distribución dada de ingresos, traza el porcentaje acumulado de la población (desplegado desde el más pobre hasta el más rico) frente a su participación en el ingreso total. El área entre la curva de Lorenz y la línea de 45 grados se conoce como el área de Lorenz. El índice de Gini es igual al doble del área de Lorenz. Una población con la curva de Lorenz más cerca de la línea de 45 grados tiene una distribución de ingresos más equitativa. Si todos los ingresos son iguales, la curva de Lorenz degenera a la línea de 45 grados.

Para realizar la curva de Lorenz en R se utiliza la función svylorenz. A continuación, se muestran los códigos computacionales para realizar la curva de Lorenz para los ingresos:

library(convey)  
svylorenz(formula = ~Income,  
 design = diseno\_gini,  
 quantiles = seq(0,1,.05),  
 alpha = .01 )



## lorenz SE  
## L(0) 0.0000000 0.0000  
## L(0.05) 0.0068191 0.0008  
## L(0.1) 0.0175964 0.0013  
## L(0.15) 0.0316596 0.0033  
## L(0.2) 0.0492230 0.0041  
## L(0.25) 0.0694365 0.0056  
## L(0.3) 0.0925871 0.0064  
## L(0.35) 0.1181331 0.0071  
## L(0.4) 0.1469261 0.0082  
## L(0.45) 0.1791978 0.0095  
## L(0.5) 0.2158231 0.0106  
## L(0.55) 0.2565784 0.0123  
## L(0.6) 0.3027002 0.0137  
## L(0.65) 0.3537989 0.0149  
## L(0.7) 0.4096304 0.0159  
## L(0.75) 0.4706565 0.0167  
## L(0.8) 0.5398749 0.0177  
## L(0.85) 0.6174169 0.0183  
## L(0.9) 0.7042464 0.0176  
## L(0.95) 0.8151774 0.0152  
## L(1) 1.0000000 0.0000

Los argumentos que requiere la función son, inicialmente, los ingresos de los hogares y el diseño muestral complejo. Adicionalmente, se definen una secuencia de probabilidades que define la suma de los cuantiles a calcular (quantiles) y por último, un número que especifica el nivel de confianza para el gráfico (alpha).

## 4.5 Análisis de correlación en encuestas de hogares

En el estudio de encuestas de hogares, además de describir variables de forma individual, resulta fundamental analizar cómo se relacionan entre sí. Una de las herramientas más utilizadas es el coeficiente de correlación de Pearson, que mide la fuerza y la dirección de la relación lineal entre dos variables numéricas. Sus valores oscilan entre –1 y 1:

* Un valor positivo indica que ambas variables tienden a aumentar al mismo tiempo.
* Un valor negativo señala que cuando una variable crece, la otra tiende a disminuir.
* Valores próximos a cero sugieren una relación lineal débil o inexistente.

Por ejemplo, puede ser de interés analizar si el ingreso de los hogares está asociado con su nivel de gasto, y en qué magnitud.

### 4.5.1 Ajuste al diseño muestral

En encuestas complejas, la correlación no puede calcularse como en una muestra simple. Es necesario incorporar los pesos muestrales para que la estimación sea representativa de la población. Este ajuste tiene en cuenta la estratificación, la conglomeración y las probabilidades de selección desiguales.

El cálculo ponderado implica evaluar la covarianza entre las dos variables y dividirla entre el producto de sus desviaciones estándar ponderadas, eliminando así la influencia de las unidades de medida.

### 4.5.2 Expresión formal

El coeficiente de correlación de Pearson ajustado por pesos se expresa como:

Cuando las variables son categóricas u ordinales, se deben emplear otras medidas de asociación (como el coeficiente de contingencia o de correlación policórica).

### 4.5.3 Implementación en R

El paquete survey cuenta con la función svyvar que permite obtener matrices de covarianzas ponderadas, a partir de las cuales se puede calcular la correlación. Otra opción más directa es usar svycor de paquetes complementarios como svycor o implementar el cálculo manualmente.

Ejemplo con ingreso y gasto de los hogares:

# Covarianza ponderada  
svy\_cov <- svyvar(~Income + Expenditure, design = diseno)

El resultado entrega la matriz de correlaciones entre las variables analizadas bajo el diseño complejo.

## 4.6 Análisis de la relación entre dos variable continuas

En muchos análisis de variables relacionadas con encuestas de hogares no solo basta con analizar el comportamiento de variables de manera individual, por ejemplo, ingresos medios de hombres y mujeres en un país sino también, analizar la diferencia entre los ingresos de los hombres y las mujeres. Esto último con el fin de ir cerrando la brecha salarial que existe.

En este capítulo se estudiará la prueba de hipótesis para diferencia de medias, se darán las herramientas computacionales para estimar razones y contrastes.

## 4.7 Pruebas para la comparación de grupos

El análisis de diferencias entre grupos es un objetivo frecuente en las encuestas de hogares. Un ejemplo clásico es la pregunta: *¿Existen diferencias estadísticamente significativas en el ingreso medio entre hogares dirigidos por hombres y aquellos dirigidos por mujeres?*

Para responder a este tipo de interrogantes se aplican **pruebas de hipótesis**, procedimientos estadísticos que contrastan afirmaciones sobre parámetros poblacionales a partir de evidencia muestral. En el contexto de encuestas, estas pruebas deben ajustarse al diseño de muestreo (estratificación, conglomerados y ponderación) para garantizar inferencias válidas.

### 4.7.1 Planteamiento general de hipótesis

Una prueba de hipótesis parte de dos proposiciones antagónicas:

Dependiendo del contexto, puede plantearse de manera unilateral ( o ). El proceso consiste en evaluar si la evidencia contenida en la muestra es suficiente para rechazar en favor de .

En muchos casos, los parámetros de interés (medias, proporciones, totales o razones) pueden expresarse como **combinaciones lineales de estadísticas descriptivas**:

con estimador:

y varianza:

Este marco general abarca, entre otros casos, la **diferencia entre medias poblacionales**.

### 4.7.2 Diferencia de medias poblacionales

Sea la media poblacional del dominio , con . El parámetro de interés es:

Por ejemplo, podría ser el ingreso medio en hogares con jefatura masculina y el ingreso medio en hogares con jefatura femenina.

El estimador de la diferencia es:

cuyo error estándar se calcula como:

### 4.7.3 Estadístico de prueba e intervalo de confianza

El contraste se realiza con el estadístico:

donde los grados de libertad se aproximan como (número de UPMs menos número de estratos).

El intervalo de confianza para al nivel se define como:

### 4.7.4 Ejemplo aplicado en R

En encuestas complejas, estas pruebas se implementan con la función svyttest() del paquete survey, que incorpora automáticamente los ajustes del diseño.

**Ejemplo 1. Diferencia de ingresos por sexo (población total):**

svyttest(Income ~ Sex, design = diseno, level = 0.95)

##   
## Design-based t-test  
##   
## data: Income ~ Sex  
## t = 1.3625, df = 118, p-value = 0.1756  
## alternative hypothesis: true difference in mean is not equal to 0  
## 95 percent confidence interval:  
## -12.82205 69.38503  
## sample estimates:  
## difference in mean   
## 28.28149

**Resultados (ejemplo ficticio):**

* Estadístico t = -1.48
* p-valor = 0.14
* Intervalo de confianza (95%): (-77.35 , 11.41)

Interpretación: con un 95% de confianza no existe evidencia estadística suficiente para afirmar que los ingresos medios difieren por sexo.

**Ejemplo 2. Diferencia de ingresos por sexo en zona urbana:**

svyttest(Income ~ Sex, design = sub\_Urbano, level = 0.95)

##   
## Design-based t-test  
##   
## data: Income ~ Sex  
## t = 1.5667, df = 63, p-value = 0.1222  
## alternative hypothesis: true difference in mean is not equal to 0  
## 95 percent confidence interval:  
## -12.31754 101.74023  
## sample estimates:  
## difference in mean   
## 44.71134

Resultado: nuevamente no se rechaza , lo que indica ausencia de diferencias significativas en el ingreso medio por sexo en la zona urbana.

**Ejemplo 3. Diferencia de ingresos por sexo, restringido a mayores de 18 años:**

svyttest(Income ~ Sex,  
 design = diseno %>% filter(Age > 18),  
 level = 0.95)

##   
## Design-based t-test  
##   
## data: Income ~ Sex  
## t = 1.5263, df = 118, p-value = 0.1296  
## alternative hypothesis: true difference in mean is not equal to 0  
## 95 percent confidence interval:  
## -10.72746 82.85253  
## sample estimates:  
## difference in mean   
## 36.06253

Conclusión: tampoco se rechaza . Es decir, no se observa evidencia estadística de diferencias en los ingresos medios entre hombres y mujeres mayores de 18 años.

El procedimiento descrito no se limita a las medias, sino que también puede aplicarse a proporciones, totales, razones o cualquier función diferenciable de totales. En todos los casos, el contraste se fundamenta en la estimación puntual, su varianza (incluyendo covarianzas cuando corresponde) y la comparación con la distribución t ajustada al diseño muestral.

## 4.8 Estimando razones en encuestas de hogares

Un caso particular de una función no lineal de totales es la **razón poblacional**, definida como el cociente de dos totales poblacionales de características de interés. En encuestas de hogares, este parámetro es relevante cuando, por ejemplo, se requiere conocer la cantidad de hombres por cada mujer, la proporción de ocupados respecto a la población en edad de trabajar, o la cantidad de mascotas por cada hogar. Puesto que la razón es un cociente de totales, tanto el numerador como el denominador son cantidades desconocidas y, por lo tanto, deben estimarse *(Bautista, 1998)*. Formalmente, la razón poblacional se define como:

y su estimador puntual en el marco de un muestreo complejo se expresa como:

Sin embargo, dado que es un cociente entre dos estimadores —es decir, dos variables aleatorias— el cálculo de su varianza no es trivial. Para ello se emplea la linealización de Taylor, como lo muestra *Gutiérrez (2016)*, o bien métodos de remuestreo. En términos de la función de estimación, se define:

### Implementación computacional en R

En la práctica, estos cálculos se facilitan gracias a los paquetes especializados. En particular, la función survey\_ratio implementa la estimación de razones y sus varianzas en el marco de encuestas complejas. Para ello es necesario especificar claramente:

* La variable del **Expenditure** (numerator),
* La variable del **Income** (denominator),
* El nivel de confianza (level) para los intervalos, y
* Las estadísticas de resumen deseadas (vartype).

A continuación, se ilustra cómo estimar la razón entre el gasto y el ingreso de los hogares:

diseno %>% summarise(  
 Razon = survey\_ratio(  
 numerator = Expenditure,  
 denominator = Income,  
 level = 0.95,  
 vartype = c("se", "ci")  
 ))

## # A tibble: 1 × 4  
## Razon Razon\_se Razon\_low Razon\_upp  
## <dbl> <dbl> <dbl> <dbl>  
## 1 0.649 0.0232 0.603 0.695

### Interpretación aplicada

Las razones permiten expresar la relación entre dos variables, lo cual resulta especialmente útil para construir indicadores comparativos y de seguimiento. Un ejemplo directo en encuestas de hogares es la razón **gasto/ingreso**, que ayuda a identificar patrones de consumo y niveles de sostenibilidad económica de los hogares.

Más allá de este caso, las razones también se utilizan en marcos internacionales. Por ejemplo, el **Indicador 2.1.1 de los Objetivos de Desarrollo Sostenible (ODS)** —prevalencia de subalimentación— se calcula a partir de la razón entre el consumo de alimentos, medido en calorías ingeridas, y los requerimientos energéticos mínimos de la dieta, determinados según edad, sexo y nivel de actividad física.

De esta forma, las razones constituyen una herramienta central para transformar los resultados de encuestas en indicadores significativos que facilitan el análisis de realidades socioeconómicas y poblacionales.

Como se puede observar, la razón entre el gasto y el ingreso es, aproximando, 0.71. Lo que implica que por cada unidad 100 unidades monetarias que le ingrese al hogar, se gastan 71 unidades, consiguiendo un intervalo de confianza al 95% de 0.65 y 0.76.

Si ahora el objetivo es estimar la razón entre mujeres y hombres en la base de ejemplo, se realiza de la siguiente manera:

diseno %>% summarise(  
 Razon = survey\_ratio(  
 numerator = (Sex == "Female"),  
 denominator = (Sex == "Male"),  
 level = 0.95,  
 vartype = c("se", "ci")  
 ))

## # A tibble: 1 × 4  
## Razon Razon\_se Razon\_low Razon\_upp  
## <dbl> <dbl> <dbl> <dbl>  
## 1 1.11 0.0351 1.04 1.18

Como la variable sexo en la base de datos es una variable categórica, se tuvo la necesidad de generar las variables dummys para su cálculo realizando, Sex == “Female” para el caso de las mujeres y Sex == “Male” para el caso de los hombres. Los resultados del ejercicio anterior muestran que en la base de datos hay más mujeres que hombres, generando una razón de 1.13. Esto significa que, por cada 100 hombres hay aproximadamente 113 mujeres con un intervalo que varía entre 1.04 y 1.21.

Si se desea hacer la razón de mujeres y hombres pero en la zona rural, se haría de la siguiente manera:

sub\_Rural %>% summarise(  
 Razon = survey\_ratio(  
 numerator = (Sex == "Female"),  
 denominator = (Sex == "Male"),  
 level = 0.95,  
 vartype = c("se", "ci")  
 ))

## # A tibble: 1 × 4  
## Razon Razon\_se Razon\_low Razon\_upp  
## <dbl> <dbl> <dbl> <dbl>  
## 1 1.07 0.0352 0.997 1.14

Obteniendo nuevamente que hay más mujeres que hombres. Ahora bien, otro análisis de interés es estimar la razón de gastos pero solo en la población femenina. A continuación, se presentan los códigos computacionales.

sub\_Mujer %>% summarise(  
 Razon = survey\_ratio(  
 numerator = Expenditure,  
 denominator = Income,  
 level = 0.95,  
 vartype = c("se", "ci")  
 ))

## # A tibble: 1 × 4  
## Razon Razon\_se Razon\_low Razon\_upp  
## <dbl> <dbl> <dbl> <dbl>  
## 1 0.658 0.0199 0.619 0.698

Dando como resultado que por cada 100 unidades monetarias que le ingresan a las mujeres se gastan 70 con un intervalo de confianza entre 0.65 y 0.76. Por último, análogamente para los hombres, la razón de gastos resulta muy similar que para las mujeres.

sub\_Hombre %>% summarise(  
 Razon = survey\_ratio(  
 numerator = Expenditure,  
 denominator = Income,  
 level = 0.95,  
 vartype = c("se", "ci")  
 ))

## # A tibble: 1 × 4  
## Razon Razon\_se Razon\_low Razon\_upp  
## <dbl> <dbl> <dbl> <dbl>  
## 1 0.639 0.0288 0.582 0.696

## 4.9 Estimando contrastes en encuestas de hogares

En muchas ocasiones, en encuestas de hogares se requiere comparar más de dos poblaciones al mismo tiempo, por ejemplo, comparar los ingresos medios de los hogares en 3 regiones o municipalidades en la postpandemia con el fin de verificar y sectorizar aquellas municipalidades o regiones donde más impacto en el desempleo y falta de ingresos tuvo el Covid-19 en los hogares. En casos como estos la diferencia de media que estudiamos en capítulos anteriores se queda corta dado que permite solo comprar parejas de poblaciones y por ende que, hacer contraste resulta una muy buena alternativa para abordar este tipo de problemas.

Recurriendo en las definiciones que se han trabajado en este capítulo, un contraste es una combinación lineal de parámetros de la forma:

Una estimación de esta función está dada por:

cuya varianza del estimador se calcula como sigue:

Los procedimientos metodológicos para implementar los contrastes en diseños de muestreo complejos están desarrolladas en la función svycontrast. A continuación, se muestra el uso de dicha función para el cálculo de contraste en la base de datos de ejemplo, comparando el promedio de ingresos por región.

Como primer ejemplo, se realizará la comparación de dos poblaciones, las regiones Norte y Sur () y luego sí se compararán todas las regiones.

Puesto que esto es un contraste en donde hay 5 regiones y solo se construirá el contraste para la región Norte y la Sur, el contraste queda definido de la siguiente manera:

que de forma matricial queda de la siguiente manera:

Como se puede observar, en este caso el vector de contraste es .

Ahora bien, para realizar el procesos de la construcción del estimador del contraste y su varianza estimada paso a paso se inicia con calcular las medias estimadas por región con la función svyby como se muestra a continuación:

prom\_region <- svyby(formula = ~Income,  
 by = ~Region,  
 design = diseno,  
 FUN = svymean,  
 na.rm=T,  
 covmat = TRUE,  
 vartype = c("se", "ci"))  
prom\_region

## Region Income se ci\_l ci\_u  
## Norte Norte 552.3637 55.35987 443.8603 660.8670  
## Sur Sur 625.7740 62.40574 503.4610 748.0870  
## Centro Centro 650.7820 61.46886 530.3053 771.2588  
## Occidente Occidente 517.0071 46.22077 426.4161 607.5982  
## Oriente Oriente 541.7543 71.66487 401.2938 682.2149

La función svyby permite aplicar una función, en este caso la media (svymean) por región (by) utilizando el diseño muestral empleado (design). Las demás componentes de la función ya se han utilizado previamente. Como resultado de aplicar esta función se obtienen las medias estimadas de los ingresos por región. Se tomarán solo los ingresos medios estimados de las regiones Norte y Sur y calcularemos su diferencia:

# Paso 1: diferencia de estimaciones (Norte - Sur)  
552.4 - 625.8

## [1] -73.4

El paso siguiente es calcular la matriz de varianzas y covarianzas y de allí extraer las varianzas y covarianzas de las regiones Norte y Sur:

# Paso 2: Matriz de varianzas y covarianzas  
vcov(prom\_region)

## Norte Sur Centro Occidente Oriente  
## Norte 3064.715 0.000 0.00 0.000 0.000  
## Sur 0.000 3894.476 0.00 0.000 0.000  
## Centro 0.000 0.000 3778.42 0.000 0.000  
## Occidente 0.000 0.000 0.00 2136.359 0.000  
## Oriente 0.000 0.000 0.00 0.000 5135.854

Para calcular el error estándar de la diferencia (contraste) se usará las propiedades de la varianza como es tenemos:

sqrt(3065 + 3894 - 2\*0)

## [1] 83.42062

Finalmente, la función svycontrast nos devuelve el contraste estimado y su error estándar. Los argumentos de esta función son los promedios de los ingresos estimados (stat) y las constantes de contraste (contrasts).

svycontrast(stat = prom\_region,  
 contrasts = list(diff\_NS = c(1, -1, 0, 0, 0))) %>%  
 data.frame()

## contrast diff\_NS  
## diff\_NS -73.41034 83.42176

Obteniendo como resultado que los ingresos medios estimados para la región Sur es 73.4 unidades monetarias mayor que los ingresos en la región Norte con un error estándar de 83.42 unidades.

Ahora bien, si el objetivo es estimar los siguientes contrastes:

* ,

Que escritas de forma matricial se tiene:

Ahora, aplicando la función svycontrast en R se obtiene:

svycontrast(stat = prom\_region,  
 contrasts = list(  
 Norte\_sur = c(1, 0, -1, 0, 0),  
 Sur\_centro = c(0, 1, -1, 0, 0),  
 Occidente\_Oriente = c(0, 0, 0, 1, -1))) %>% data.frame()

## contrast SE  
## Norte\_sur -98.41834 82.72324  
## Sur\_centro -25.00800 87.59507  
## Occidente\_Oriente -24.74720 85.27727

De lo cual se puede concluir que, las regiones con los ingresos medios de los hogares más similares son la región sur y la región centro.

También es posible construir contraste en variables que estén correlacionadas. Por ejemplo, Ingreso y Sexo. Como se hizo en el ejemplo anterior, se inicia con el promedio estimado por sexo.

prom\_sexo <- svyby(formula = ~Income,  
 by = ~Sex,  
 design = diseno,  
 FUN = svymean,  
 na.rm=T,  
 covmat = TRUE,  
 vartype = c("se", "ci"))  
prom\_sexo

## Sex Income se ci\_l ci\_u  
## Female Female 557.5681 25.82995 506.9423 608.1939  
## Male Male 585.8496 34.58759 518.0592 653.6400

El contraste a estimar es:

Por tanto, usando la función svycontrast se obtiene el contraste estimado:

svycontrast(stat = prom\_sexo,  
 contrasts = list(diff\_Sexo = c(1, -1))) %>%  
 data.frame()

## contrast diff\_Sexo  
## diff\_Sexo -28.28149 20.75651

Obteniendo como resultado que, en promedio, los hombres obtienen 28.3 unidades monetarias más que las mujeres con una desviación de 20.76.

Otra posibilidad es poder obtener resultados agregados, por ejemplo:

sum\_region <- svyby( ~ Income, ~ Region,  
 diseno, svytotal, na.rm = T,  
 covmat = TRUE,  
 vartype = c("se", "ci"))  
sum\_region

## Region Income se ci\_l ci\_u  
## Norte Norte 14277323 1507575 11322530 17232115  
## Sur Sur 16068151 1877989 12387359 19748942  
## Centro Centro 16483319 2383556 11811634 21155003  
## Occidente Occidente 16853540 1823807 13278944 20428135  
## Oriente Oriente 22111335 2833460 16557856 27664814

La matriz de contraste queda como:

el procedimiento en R es:

svycontrast(stat = sum\_region,  
 contrasts = list(  
 Agregado\_NCS = c(1, 1, 1, 0, 0))) %>% data.frame()

## contrast Agregado\_NCS  
## Agregado\_NCS 46828792 3388357

Por otro lado, si se desean obtener los promedios por categorías. Por ejemplo:

donde es el número de categorías de la variable. En R se hace de la siguiente manera:

prom\_edad <- svyby(formula = ~Income,  
 by = ~CatAge,  
 design = diseno,  
 FUN = svymean,  
 na.rm=T,  
 covmat = TRUE)  
prom\_edad

## CatAge Income se  
## 0-5 0-5 463.7715 28.86795  
## 6-15 6-15 511.6179 34.88031  
## 16-30 16-30 607.2917 37.41561  
## 31-45 31-45 573.4167 26.94744  
## 46-60 46-60 763.0610 58.97170  
## Más de 60 Más de 60 466.6133 31.20795

Cuya matriz de contraste estaría dada por:

El procedimiento en R es:

svycontrast(stat = prom\_edad,  
 contrasts = list(  
 agregado\_edad = c(1/6, 1/6, 1/6, 1/6, 1/6, 1/6))) %>% data.frame()

## contrast agregado\_edad  
## agregado\_edad 564.2954 25.40408

Puesto que los contrastes, como ya se mencionó, es una función lineal de parámetros, se puede también realizar contraste con parámetros tipo razón. Por ejemplo, la relación de gastos contra ingresos por sexo. A continuación, se muestran los códigos computacionales:

razon\_sexo <- svyby( formula = ~Income,  
 by = ~Sex,  
 denominator = ~Expenditure,  
 design = diseno,  
 FUN = svyratio,  
 na.rm=T, covmat = TRUE,  
 vartype = c("se", "ci"))  
razon\_sexo

## Sex Income/Expenditure se.Income/Expenditure ci\_l ci\_u  
## Female Female 1.519060 0.04582607 1.429243 1.608878  
## Male Male 1.564762 0.07044239 1.426698 1.702827

Cuya estimación de contraste sería:

svycontrast(stat = razon\_sexo,  
 contrasts = list(  
 diff\_sexo = c(1, -1))) %>% data.frame()

## contrast diff\_sexo  
## diff\_sexo -0.04570214 0.04163431

de lo que se puede concluir que la diferencia de las proporciones es 0.045 en favor de los hombres.

# 5 Análisis de variables categóricas en encuestas de hogares

Al analizar datos de encuestas de hogares, uno de los productos más habituales son los parámetros descriptivos, cuyo propósito es sintetizar las principales características de la población. Estas estimaciones permiten ofrecer una representación clara y comprensible de la realidad poblacional a partir de la información obtenida en una muestra representativa.

En ocasiones, no es sencillo distinguir entre las variables denominadas cualitativas y cuantitativas, puesto que algunas variables de tipo cuantitativo pueden considerarse categóricas si se divide el rango de valores en intervalos o categorías. Un ejemplo clásico es la variable edad, que en una encuesta de hogares se registra como cuantitativa, pero puede agruparse en categorías. Por ejemplo, en Colombia, las categorías podrían ser: Adolescencia (12–18 años), Juventud (14–26 años), Adultez (27–59 años) y Persona Mayor (60 años o más), incorporando también conceptos de envejecimiento y vejez.

De manera inversa, una variable categórica también puede transformarse en cuantitativa mediante análisis específicos, como un análisis de correspondencias. Esto es frecuente cuando se construyen índices compuestos, como el índice de fuerza laboral. En el contexto de encuestas, las preguntas que contienen variables categóricas son muy comunes y sus resultados suelen presentarse en porcentajes, por ejemplo: parentesco, sexo, jefe o jefa del hogar, acceso a agua potable, entre otros.

Entre los resultados más comunes de este tipo de análisis se encuentran frecuencias, proporciones, medias y totales. Las medias proporcionan información sobre el valor promedio de una variable, mientras que los totales reflejan su acumulado en toda la población. Las frecuencias cuentan cuántos hogares o individuos pertenecen a una categoría determinada —por ejemplo, el número de personas en situación de pobreza—, y las proporciones expresan la participación relativa de quienes presentan una característica específica, como el porcentaje de población pobre.

Actualmente, el análisis descriptivo va más allá de los parámetros básicos, incorporando métricas más complejas. Se estiman cuantiles de variables numéricas, como la mediana del ingreso de los hogares, para describir la distribución de los datos con mayor detalle. Además, se aplican indicadores especializados para evaluar fenómenos concretos, como los índices FGT para la medición de pobreza, los indicadores de desigualdad (Gini, Theil, Atkinson) y de polarización (Wolfson, DER), entre otros (Jacob, Damico y Pessoa, 2024).

library(tidyverse)  
  
encuesta <- readRDS("Data/encuesta.rds")  
head(encuesta)

## HHID Stratum NIh nIh dI PersonID PSU Zone Sex Age MaritalST  
## 1 idHH00031 idStrt001 9 2 4.5 idPer01 PSU0003 Rural Male 68 Married  
## 2 idHH00031 idStrt001 9 2 4.5 idPer02 PSU0003 Rural Female 56 Married  
## 3 idHH00031 idStrt001 9 2 4.5 idPer03 PSU0003 Rural Female 24 Married  
## 4 idHH00031 idStrt001 9 2 4.5 idPer04 PSU0003 Rural Male 26 Married  
## 5 idHH00031 idStrt001 9 2 4.5 idPer05 PSU0003 Rural Female 3 <NA>  
## 6 idHH00041 idStrt001 9 2 4.5 idPer01 PSU0003 Rural Female 61 Widowed  
## Income Expenditure Employment Poverty dki dk wk Region CatAge  
## 1 409.87 346.34 Employed NotPoor 8 36 34.50371 Norte Más de 60  
## 2 409.87 346.34 Employed NotPoor 8 36 33.63761 Norte 46-60  
## 3 409.87 346.34 Employed NotPoor 8 36 33.63761 Norte 16-30  
## 4 409.87 346.34 Employed NotPoor 8 36 34.50371 Norte 16-30  
## 5 409.87 346.34 <NA> NotPoor 8 36 33.63761 Norte 0-5  
## 6 823.75 392.24 Employed NotPoor 8 36 33.63761 Norte Más de 60

**Definición del diseño y creación de variables categóricas**

Se inicia este capítulo haciendo el ajuste del diseño muestral (como se mostró en capítulos anteriores) usando como ejemplo la misma base de datos del capítulo anterior. Luego, para efectos del ejemplo, se genera una variable categórica la cual indica si la persona encuestada está en estado de pobreza o no como sigue:

library(survey)  
library(srvyr)  
options(survey.lonely.psu = "adjust")  
  
diseno <- encuesta %>%   
 as\_survey\_design(  
 strata = Stratum,   
 ids = PSU,   
 weights = wk,   
 nest = TRUE)

A continuación, se define una variable categórica que nace de variables propias de la encuesta,

diseno <- diseno %>% mutate(  
 pobreza = ifelse(Poverty != "NotPoor", 1, 0),  
 desempleo = ifelse(Employment == "Unemployed", 1, 0),  
 edad\_18 = case\_when(Age < 18 ~ "< 18 anios", TRUE ~ ">= 18 anios")  
)

Como se pudo observar en el código anterior, se ha introducido la función case\_when la cual es una extensión del a función ifelse que permite crear múltiples categorías a partir de una o varias condiciones.

Como se ha mostrado anteriormente, en ocasiones se desea realizar estimaciones por sub-grupos de la población, en este caso se extraer 4 sub-grupos de la encuesta y se definen a continuación:

sub\_Urbano <- diseno %>% filter(Zone == "Urban")  
sub\_Rural <- diseno %>% filter(Zone == "Rural")  
sub\_Mujer <- diseno %>% filter(Sex == "Female")  
sub\_Hombre <- diseno %>% filter(Sex == "Male")

## 5.1 Tamaño de población y subpoblaciones en encuestas de hogares

En el análisis de encuestas de hogares, resulta esencial determinar el tamaño de las subpoblaciones, es decir, identificar cuántas personas u hogares pertenecen a categorías específicas y qué proporción representan dentro del total poblacional. Este tipo de estimaciones permite caracterizar el perfil demográfico y socioeconómico de la población, información clave para orientar la asignación de recursos, el diseño de políticas públicas y la formulación de programas sociales.

Así, es de gran utilidad conocer cuántas personas se encuentran por debajo de la línea de pobreza, cuántas no tienen empleo o cuántas han alcanzado determinado nivel educativo. Analizar cómo se distribuyen los individuos entre distintas categorías ofrece información indispensable para reducir brechas y avanzar hacia un desarrollo inclusivo.

La estimación del tamaño de una población o subpoblación se realiza a partir de variables categóricas, que segmentan a la población en grupos mutuamente excluyentes. Estas categorías pueden corresponder, por ejemplo, a quintiles de ingreso, estados de ocupación o niveles educativos alcanzados. El tamaño poblacional hace referencia al número total de individuos u hogares que, en la base de datos de la encuesta, pertenecen a una categoría determinada. Para obtener estas estimaciones, se combinan las respuestas de los encuestados con los pesos muestrales, que indican cuántas personas u hogares representa cada unidad de la muestra dentro de la población total.

El estimador del tamaño de la población se define como:

donde corresponde a la muestra de hogares o individuos en la UPM del estrato ; representa la muestra de UPM seleccionadas en el estrato ; y es el peso o factor de expansión de la unidad en la UPM del estrato .

La estimación del tamaño de una subpoblación sigue el mismo principio que el cálculo del tamaño poblacional total, pero se enfoca en un subconjunto definido por una característica específica. Para determinar cuántas personas pertenecen a una categoría particular, se identifica dicho grupo en la base de datos y se suman sus pesos muestrales. Esto permite cuantificar grupos de interés específicos y conocer su tamaño dentro de la población:

donde es una variable binaria que toma el valor de 1 si la unidad de la UPM en el estrato pertenece a la categoría de la variable discreta , y 0 en caso contrario. Si fue utilizada en la calibración de los pesos, el valor de coincidirá con el control externo aplicado.

### Estimaciones de totales en R

En esta sección se presentan los procedimientos para estimar tamaños de población y subpoblaciones usando R, con el diseño muestral definido previamente. Por ejemplo, para estimar el tamaño de la población por zona:

tamano\_zona <- diseno %>% group\_by(Zone) %>%   
 summarise( n = unweighted(n()),   
 Nd = survey\_total(vartype = c("se","ci")))  
  
tamano\_zona

## # A tibble: 2 × 6  
## Zone n Nd Nd\_se Nd\_low Nd\_upp  
## <chr> <int> <dbl> <dbl> <dbl> <dbl>  
## 1 Rural 1297 72102. 3062. 66039. 78165.  
## 2 Urban 1308 78164. 2847. 72526. 83802.

En la tabla resultante, *n* indica el número de observaciones en la muestra por zona y *Nd* representa la estimación del total de observaciones en la población. La función unweighted() calcula resúmenes no ponderados a partir de los datos muestrales.

Por ejemplo, el tamaño de muestra en la zona rural fue de 1297 personas y en la urbana de 1308. Esto permitió estimar una población de 72,102 (desviación estándar 3,062) en la zona rural y 78,164 (desviación estándar 2,847) en la zona urbana. Con un nivel de confianza del 95%, los intervalos de confianza fueron:

* Zona rural: (66,038.5, 78,165.4)
* Zona urbana: (72,526.2, 83,801.7)

De manera similar, es posible estimar el número de personas en condición de pobreza extrema, pobreza y no pobres:

tamano\_pobreza <- diseno %>% group\_by(Poverty) %>%   
 summarise(Nd = survey\_total(vartype = c("se","ci")))  
tamano\_pobreza

## # A tibble: 3 × 5  
## Poverty Nd Nd\_se Nd\_low Nd\_upp  
## <fct> <dbl> <dbl> <dbl> <dbl>  
## 1 NotPoor 91398. 4395. 82696. 100101.  
## 2 Extreme 21519. 4949. 11719. 31319.  
## 3 Relative 37349. 3695. 30032. 44666.

Estos cálculos permiten obtener estimaciones precisas y sus intervalos de confianza para cada subpoblación, facilitando el análisis socioeconómico y la toma de decisiones basadas en evidencia.

Otra variable de interés en encuestas de hogares es conocer el estado de ocupación de las personas. A continuación, se muestra el código computacional:

tamano\_ocupacion <- diseno %>%   
 group\_by(Employment) %>%   
 summarise( Nd = survey\_total(vartype = c("se","ci")))  
tamano\_ocupacion

## # A tibble: 4 × 5  
## Employment Nd Nd\_se Nd\_low Nd\_upp  
## <fct> <dbl> <dbl> <dbl> <dbl>  
## 1 Unemployed 4635. 761. 3129. 6141.  
## 2 Inactive 41465. 2163. 37183. 45748.  
## 3 Employed 61877. 2540. 56847. 66907.  
## 4 <NA> 42289. 2780. 36784. 47794.

De los resultados de la estimación se puede concluir que, 4634.8 personas están desempleadas con un intervalo de confianza de (3128.6, 6140.9). 41465.2 personas están inactivas con un intervalo de confianza de (37182.6, 45747.8) y por último, 61877.0 personas empleadas con intervalos de confianza (36784.2, 47793.5).

Utilizando la función group\_by es posible obtener resultados por más de un nivel de agregación. A continuación, se muestra la estimación ocupación desagregada por niveles de pobreza:

tamano\_ocupacion\_pobreza <- diseno %>%   
 group\_by(Employment, Poverty) %>%   
 cascade( Nd = survey\_total(vartype = c("se","ci")), .fill = "Total") %>%  
 data.frame()  
tamano\_ocupacion\_pobreza

## Employment Poverty Nd Nd\_se Nd\_low Nd\_upp  
## 1 Unemployed NotPoor 1768.375 405.3765 965.6891 2571.061  
## 2 Unemployed Extreme 1169.201 348.1340 479.8603 1858.541  
## 3 Unemployed Relative 1697.231 457.8077 790.7262 2603.736  
## 4 Unemployed Total 4634.807 760.6242 3128.6948 6140.919  
## 5 Inactive NotPoor 24346.008 1736.2770 20908.0064 27784.010  
## 6 Inactive Extreme 6421.825 1320.7349 3806.6383 9037.012  
## 7 Inactive Relative 10697.414 1460.2792 7805.9155 13588.913  
## 8 Inactive Total 41465.248 2162.8040 37182.6798 45747.816  
## 9 Employed NotPoor 44600.347 2596.1915 39459.6282 49741.065  
## 10 Employed Extreme 5127.531 1121.6461 2906.5601 7348.503  
## 11 Employed Relative 12149.142 1346.6159 9482.7078 14815.576  
## 12 Employed Total 61877.020 2540.0762 56847.4153 66906.624  
## 13 Total Total 150266.000 4181.3587 141986.4921 158545.508  
## 14 <NA> NotPoor 20683.603 1256.6158 18195.3777 23171.827  
## 15 <NA> Extreme 8800.209 2979.9150 2899.6792 14700.738  
## 16 <NA> Relative 12805.115 1551.0291 9733.9220 15876.307  
## 17 <NA> Total 42288.926 2779.9913 36784.2652 47793.586

De lo cual se puede concluir, entre otros que, 44600.3 personas que trabajan no son pobres con un intervalo de confianza (39459.6, 49741.0) y 6421.8 inactivas están en pobreza extrema con un intervalo de confianza de (3806.6, 9037.0).

## 5.2 Estimación de proporciones

En las encuestas de hogares, **las proporciones** permiten expresar el peso relativo que tienen determinados grupos dentro de la población. Por ejemplo, conocer el porcentaje de hogares que se encuentran por debajo de la línea de pobreza es esencial para evaluar desigualdades socioeconómicas. Para obtener este indicador, se calcula el promedio ponderado de una variable dicotómica, lo que asegura que la estimación represente adecuadamente la distribución poblacional.

De acuerdo con Heeringa, West y Berglund (2017), al transformar las categorías de respuesta originales en variables indicadoras con valores de 1 y 0 (por ejemplo, 1 = Sí y 0 = No), la proporción estimada se obtiene mediante:

Dado que se trata de un estimador no lineal, su varianza puede aproximarse mediante la **técnica de linealización de Taylor**, utilizando como función de estimación . Actualmente, la mayoría de los programas estadísticos generan estas proporciones junto con sus errores estándar, generalmente presentados en forma de porcentajes.

En situaciones donde las proporciones se aproximan a 0 o 1, puede ser necesario aumentar el tamaño de la muestra para garantizar resultados sólidos. También es recomendable aplicar métodos que aseguren que los intervalos de confianza permanezcan dentro del rango [0,1], ya que los intervalos convencionales basados en la normal pueden desbordar estos límites y perder su utilidad interpretativa. Una alternativa es la **transformación logit** de la proporción estimada:

donde , siendo el cuantil de la distribución t de Student con grados de libertad, dejando un área a su derecha. Este procedimiento asegura resultados interpretables incluso para proporciones extremas (0 o 1).

### 5.2.1 Estimación de proporciones en R

Otro parámetro de interés en las encuestas de hogares, particularmente con variables categóricas, es la estimación de **proporciones poblacionales** y sus errores estándar. En términos de notación, la estimación de proporciones de población se define como y las proporciones muestrales como . Es común que muchos paquetes estadísticos generen estas estimaciones en **escala de porcentaje**, mientras que R las produce en la escala [0,1].

A continuación, se muestra cómo estimar la proporción de personas por **zona** usando srvyr y el diseño muestral previamente definido:

prop\_zona <- diseno %>% group\_by(Zone) %>%   
 summarise(  
 prop = survey\_mean(vartype = c("se","ci"),   
 proportion = TRUE))  
prop\_zona

## # A tibble: 2 × 5  
## Zone prop prop\_se prop\_low prop\_upp  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Rural 0.480 0.0140 0.452 0.508  
## 2 Urban 0.520 0.0140 0.492 0.548

En este ejemplo, la función survey\_mean calcula la proporción y, con el parámetro proportion = TRUE, se indica que se desea estimar una **proporción poblacional**. Los resultados muestran que aproximadamente el 47.9% de las personas viven en zona rural (IC 95%: 45.2% - 50.7%) y el 52% en zona urbana (IC 95%: 49.2% - 54.7%).

La librería survey también cuenta con la función survey\_prop, diseñada específicamente para estimar proporciones, generando resultados equivalentes:

prop\_zona2 <- diseno %>% group\_by(Zone) %>%   
 summarise(prop = survey\_prop(vartype = c("se","ci")))  
prop\_zona2

## # A tibble: 2 × 5  
## Zone prop prop\_se prop\_low prop\_upp  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Rural 0.480 0.0140 0.452 0.508  
## 2 Urban 0.520 0.0140 0.492 0.548

El lector puede elegir la función que le resulte más cómoda, ya que ambas permiten obtener estimaciones precisas y sus intervalos de confianza, facilitando la interpretación de los resultados para variables categóricas.

Si el interés ahora se centra en estimar subpoblaciones por ejemplo, proporción de hombres y mujeres que viven en la zona urbana, el código computacional es:

prop\_sexoU <- sub\_Urbano %>% group\_by(Sex) %>%   
 summarise(prop = survey\_prop(vartype = c("se","ci")))  
prop\_sexoU

## # A tibble: 2 × 5  
## Sex prop prop\_se prop\_low prop\_upp  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Female 0.537 0.0130 0.511 0.563  
## 2 Male 0.463 0.0130 0.437 0.489

Arrojando como resultado que, el 53.6% de las mujeres y 46.4% de los hombres viven en la zona urbana y con intervalos de confianza (51%, 56.2%) y (43.7%, 48.9%) respectivamente. Los intervalos anteriores nos reflejan que, con una confianza del 95% la cantidad estimada de mujeres que viven en la zona urbana es de56% y de hombres es de 48%.

Realizando el mismo ejercicio anterior, pero ahora en la zona rural se tiene:

prop\_sexoR <- sub\_Rural %>% group\_by(Sex) %>%   
 summarise( n = unweighted(n()),  
 prop = survey\_prop(vartype = c("se","ci")))  
prop\_sexoR

## # A tibble: 2 × 6  
## Sex n prop prop\_se prop\_low prop\_upp  
## <chr> <int> <dbl> <dbl> <dbl> <dbl>  
## 1 Female 679 0.516 0.00824 0.500 0.533  
## 2 Male 618 0.484 0.00824 0.467 0.500

el 51.6% de las mujeres y el 48.4% de los hombres viven en la zona rural con intervalos de confianza de (49.9%, 53.2%) y (46,7%, 50%) respectivamente. Los intervalos de confianza anteriores nos reflejan que, inclusive, con una confianza del 95%, la cantidad estimada de mujeres en la zona rural es de 53% y de hombres es de 50%.

Ahora bien, si nos centramos solo en la población de hombres en la base de datos y se desea estimar la proporción de hombres por zona, el código computacional es el siguiente:

prop\_ZonaH <- sub\_Hombre %>% group\_by(Zone) %>%   
 summarise(prop = survey\_prop(vartype = c("se","ci")))  
prop\_ZonaH

## # A tibble: 2 × 5  
## Zone prop prop\_se prop\_low prop\_upp  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Rural 0.491 0.0178 0.455 0.526  
## 2 Urban 0.509 0.0178 0.474 0.545

En la anterior tabla se puede observar que el 49% de los hombres están en la zona rural y el 51% en la zona urbana. Si se observa el intervalo de confianza se puede concluir que, con una confianza del 95%, la población estimada de hombres que viven en la zona rural puede llegar a ser el 52% y en urbana un 54%.

Si se realiza ahora el mismo ejercicio para la mujeres el código computacional es:

prop\_ZonaM <- sub\_Mujer %>% group\_by(Zone) %>%   
 summarise(prop = survey\_prop(vartype = c("se","ci")))  
prop\_ZonaM

## # A tibble: 2 × 5  
## Zone prop prop\_se prop\_low prop\_upp  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Rural 0.470 0.0140 0.443 0.498  
## 2 Urban 0.530 0.0140 0.502 0.557

De la tabla anterior se puede inferir que, el 47% de las mujeres están en la zona rural y el 52% en la zona urbana. Observando también intervalos de confianza al 95% de (44%, 49%) y (50%, 55%) para las zonas rural y urbana respectivamente.

Si se desea estimar por varios niveles de desagregación, con el uso de la función group\_by es posible estimar un mayor número de niveles de agregación al combinar dos o más variables. Por ejemplo, si se desea estimar la proporción de hombres por zona y en estado de pobreza, se realiza de la siguiente manera:

prop\_ZonaH\_Pobreza <- sub\_Hombre %>%  
 group\_by(Zone, Poverty) %>%   
 summarise(  
 prop = survey\_prop(vartype = c("se","ci")))%>%  
 data.frame()  
prop\_ZonaH\_Pobreza

## Zone Poverty prop prop\_se prop\_low prop\_upp  
## 1 Rural NotPoor 0.5488453 0.06264753 0.42434340 0.6675180  
## 2 Rural Extreme 0.1975254 0.06745258 0.09582905 0.3637294  
## 3 Rural Relative 0.2536293 0.03724070 0.18711180 0.3340755  
## 4 Urban NotPoor 0.6599255 0.03662268 0.58415144 0.7283141  
## 5 Urban Extreme 0.1128564 0.02451869 0.07264146 0.1712240  
## 6 Urban Relative 0.2272181 0.02604053 0.17979436 0.2828371

De la salida anterior se puede observar que, en la ruralidad, el 19% de los hombres están en pobreza extrema mientras que en la zona urbana el 11% también están en pobreza extrema. Por otro lado, el 54% de los hombres que viven en la zona rural no están en pobreza mientras que, en la zona urbana el 65% no está en esta condición.

El mismo ejercicio anterior para la población de mujeres sería:

prop\_ZonaM\_Pobreza <- sub\_Mujer %>%   
 group\_by(Zone, Poverty) %>%   
 summarise( prop = survey\_prop(vartype = c("se","ci"))) %>%  
 data.frame()  
prop\_ZonaM\_Pobreza

## Zone Poverty prop prop\_se prop\_low prop\_upp  
## 1 Rural NotPoor 0.5539176 0.05568825 0.44281376 0.6598834  
## 2 Rural Extreme 0.1599702 0.05574533 0.07728197 0.3021593  
## 3 Rural Relative 0.2861122 0.04357612 0.20803909 0.3794466  
## 4 Urban NotPoor 0.6612172 0.03224726 0.59475977 0.7218725  
## 5 Urban Extreme 0.1093753 0.02209821 0.07267359 0.1613865  
## 6 Urban Relative 0.2294075 0.02655874 0.18106582 0.2861459

De la salida anterior se puede observar que, en la ruralidad, el 16% de las mujeres están en pobreza extrema mientras que en la zona urbana el 10% también están en pobreza extrema. Por otro lado, el 55% de las mujeres que viven en la zona rural no están en pobreza mientras que, en la zona urbana el 66% no está en esta condición.

Si lo que se desea ahora es estimar la proporción de hombres empleados o no por zona, se realiza de la siguiente manera:

prop\_ZonaH\_Ocupacion <- sub\_Hombre %>%  
 group\_by(Zone, Employment) %>%   
 summarise(prop = survey\_prop(vartype = c("se","ci")))%>%  
 data.frame()  
prop\_ZonaH\_Ocupacion

## Zone Employment prop prop\_se prop\_low prop\_upp  
## 1 Rural Unemployed 0.05125186 0.015733138 0.02767737 0.09298588  
## 2 Rural Inactive 0.10351629 0.020267044 0.06970747 0.15106011  
## 3 Rural Employed 0.52251375 0.026522751 0.46994089 0.57459249  
## 4 Rural <NA> 0.32271810 0.034987840 0.25763953 0.39547790  
## 5 Urban Unemployed 0.04374724 0.008492664 0.02969659 0.06400729  
## 6 Urban Inactive 0.16331307 0.018093938 0.13056379 0.20236490  
## 7 Urban Employed 0.51337023 0.023637331 0.46658553 0.55992181  
## 8 Urban <NA> 0.27956945 0.022085422 0.23799131 0.32531045

De la salida anterior se puede observar que, el 5% de los hombres que viven en la ruralidad están desempleados mientras que el 4% de los que viven en la zona urbana están en esta misma condición. Ahora bien, el 52% de los hombres que viven en la ruralidad trabajan mientras que el 51% de los que viven en la zona rural también están empleados.

Si se hace este mismo ejercicio para las mujeres se obtiene:

prop\_ZonaM\_Ocupacion <- sub\_Mujer %>%   
 group\_by(Zone, Employment) %>%   
 summarise(prop = survey\_prop(vartype = c("se","ci"))) %>%  
 data.frame()  
prop\_ZonaM\_Ocupacion

## Zone Employment prop prop\_se prop\_low prop\_upp  
## 1 Rural Unemployed 0.01017065 0.005540256 0.003443802 0.02964628  
## 2 Rural Inactive 0.44719272 0.035247218 0.378871481 0.51756811  
## 3 Rural Employed 0.23999716 0.039151859 0.171118101 0.32570701  
## 4 Rural <NA> 0.30263948 0.030765644 0.245379430 0.36676711  
## 5 Urban Unemployed 0.02109678 0.005964137 0.012019202 0.03677508  
## 6 Urban Inactive 0.36445938 0.021442387 0.323143427 0.40787461  
## 7 Urban Employed 0.38455672 0.019452094 0.346831628 0.42372325  
## 8 Urban <NA> 0.22988711 0.013850398 0.203613820 0.25845036

Para las mujeres se puede observar que, el 1% de las mujeres que viven en la ruralidad están desempleados mientras que el 2% de las que viven en la zona urbana están en esta misma condición. Ahora bien, el 24% de las mujeres que viven en la ruralidad trabajan mientras que el 38% de las que viven en la zona rural también están empleados.

Otro parámetro que es de interés es estimar en encuestas de hogares la cantidad de personas menores y mayores de edad en los hogares. A continuación, ejemplificamos la estimación de menores y mayores a 18 años cruzado por pobreza:

diseno %>% group\_by(edad\_18, pobreza) %>%   
 summarise(Prop = survey\_prop(vartype = c("se", "ci"))) %>%  
 data.frame()

## edad\_18 pobreza Prop Prop\_se Prop\_low Prop\_upp  
## 1 < 18 anios 0 0.4984504 0.03729355 0.4251710 0.5717964  
## 2 < 18 anios 1 0.5015496 0.03729355 0.4282036 0.5748290  
## 3 >= 18 anios 0 0.6646140 0.02978353 0.6033275 0.7208132  
## 4 >= 18 anios 1 0.3353860 0.02978353 0.2791868 0.3966725

De la anterior salida se puede observar que, el 50% de los menores de edad y el 33% de los mayores de edad están en estado de pobreza. Al observar los intervalos de confianza para los menores de edad en estado de pobreza se puede observar que, dicha estimación puede llegar, con una confianza del 95% a 57% mientras que a los mayores de edad puede llegar a 39%.

Ahora, si se hace este mismo ejercicio, pero esta vez cruzando con la variable que indica empleo se obtiene:

diseno %>% group\_by(edad\_18, desempleo) %>%   
 summarise(Prop = survey\_prop(vartype = c("se", "ci"))) %>%  
 data.frame()

## edad\_18 desempleo Prop Prop\_se Prop\_low Prop\_upp  
## 1 < 18 anios 0 0.166704172 0.014856561 0.139321648 0.19822898  
## 2 < 18 anios 1 0.003729693 0.001969183 0.001309174 0.01057808  
## 3 < 18 anios NA 0.829566135 0.015009188 0.797760089 0.85726505  
## 4 >= 18 anios 0 0.955234872 0.007552778 0.937660285 0.96802386  
## 5 >= 18 anios 1 0.044765128 0.007552778 0.031976144 0.06233972

De la tabla anterior se puede observar que, el 0.3% de los menores de edad y el 4% de los mayores de edad están desempleados. Adicionalmente, con una confianza del 95% y basados en la muestra se puede observar que el desempleo en menores de edad puede llegar a 0.7% y para los mayores llega a un 5%.

Por otro lado, si el objetivo ahora es estimar la cantidad de menores de edad en la zona rural se realiza de la siguiente manera:

sub\_Rural %>% group\_by(edad\_18) %>%   
 summarise(Prop = survey\_prop(vartype = c("se", "ci"))) %>%  
 data.frame()

## edad\_18 Prop Prop\_se Prop\_low Prop\_upp  
## 1 < 18 anios 0.3711613 0.03021982 0.3128746 0.4334566  
## 2 >= 18 anios 0.6288387 0.03021982 0.5665434 0.6871254

De la anterior tabla se puede observar que, el 37% de las personas que viven en la zona rural de la base de ejemplo son menores de edad con un intervalo de confianza al 95% comprendido entre 31% y 43%.

Como se mencionó al inicio del capítulo, es posible categorizar una variable de tipo cuantitativo como por ejemplo la edad y cruzarla con la variable que categoriza la empleabilidad. A continuación, se estima la edad de las mujeres por rango.

sub\_Mujer %>% mutate(edad\_rango = case\_when(  
 Age>= 18 & Age <=35 ~ "18 - 35", TRUE ~ "Otro")) %>%  
 group\_by(edad\_rango, Employment) %>%   
 summarise(Prop = survey\_prop(vartype = c("se", "ci"))) %>%   
 data.frame()

## edad\_rango Employment Prop Prop\_se Prop\_low Prop\_upp  
## 1 18 - 35 Unemployed 0.02893412 0.009142347 0.015403014 0.05370358  
## 2 18 - 35 Inactive 0.51653851 0.037905184 0.441673039 0.59066889  
## 3 18 - 35 Employed 0.45452737 0.035685710 0.385232560 0.52562948  
## 4 Otro Unemployed 0.01015164 0.004026104 0.004617517 0.02217073  
## 5 Otro Inactive 0.35271022 0.020725430 0.312834830 0.39474850  
## 6 Otro Employed 0.25483870 0.021700305 0.214292062 0.30012671  
## 7 Otro <NA> 0.38229944 0.022313379 0.339191706 0.42734277

De la anterior tabla se puede observar, entre otros que, las mujeres con edades entre 18 y 35 años el 2% están desempleadas y el 45% están empleadas. Análisis similares se pueden hacer para los demás rangos de edades.

Este mismo ejercicio se puede realizar para los hombres y hacer los mismos análisis. A continuación, se muestra el código computacional:

sub\_Hombre %>% mutate(edad\_rango = case\_when(  
 Age>= 18 & Age <=35 ~ "18 - 35",TRUE ~ "Otro")) %>%  
 group\_by(edad\_rango, Employment) %>%   
 summarise(Prop = survey\_prop(vartype = c("se", "ci"))) %>%   
 data.frame()

## edad\_rango Employment Prop Prop\_se Prop\_low Prop\_upp  
## 1 18 - 35 Unemployed 0.09637042 0.018215667 0.06584071 0.13895080  
## 2 18 - 35 Inactive 0.08939940 0.016438321 0.06175556 0.12773290  
## 3 18 - 35 Employed 0.81423018 0.022991735 0.76436394 0.85553799  
## 4 Otro Unemployed 0.02606667 0.007175709 0.01506262 0.04474457  
## 5 Otro Inactive 0.15344056 0.019883462 0.11805657 0.19706023  
## 6 Otro Employed 0.38849664 0.020270309 0.34919327 0.42930563  
## 7 Otro <NA> 0.43199614 0.021111842 0.39076987 0.47418649

## 5.3 Tablas cruzadas

En los levantamientos de encuestas de hogares, es habitual recopilar información sobre variables categóricas, que permiten clasificar a la población en grupos mutuamente excluyentes. Ejemplos comunes son el estado laboral (*ocupado*, *desempleado*, *inactivo*), el nivel educativo alcanzado (*primaria*, *secundaria*, *terciaria*) o el acceso a determinados servicios (*sí*, *no*). Explorar si dos de estas variables están asociadas constituye un elemento central del análisis, pues ofrece información valiosa en distintos ámbitos:

* En política pública, al relacionar educación y empleo para diseñar estrategias laborales.
* En evaluación de programas, al detectar variaciones en el acceso a salud según el nivel de ingresos.
* En investigación social, al estudiar vínculos entre factores demográficos y servicios para comprender dinámicas y tendencias sociales.

### 5.3.1 Definición y notación

El estudio de la asociación entre dos variables categóricas implica verificar si la distribución de una depende de la otra. Para ello se comparan las frecuencias de todas las combinaciones posibles de categorías. Por ejemplo, puede contabilizarse cuántos individuos corresponden simultáneamente a cada nivel educativo y estado laboral. Estas frecuencias pueden transformarse en proporciones, que muestran la participación relativa de cada combinación en la población.

La herramienta más común para organizar esta información es la tabla de contingencia, también conocida como tabla cruzada. En su forma más simple, corresponde a una matriz de doble entrada en la que las filas representan las categorías de una variable y las columnas las de otra. Cada celda contiene la frecuencia o proporción de casos que presentan simultáneamente la combinación .

Formalmente, sean y dos variables categóricas con y categorías, respectivamente. Bajo un modelo de superpoblación, la distribución conjunta puede expresarse como:

con la restricción .

Si se dispusiera de un censo, el número de unidades en cada celda se calcularía como:

y las proporciones poblacionales se definirían como , donde es el tamaño total de la población. En la práctica, al trabajar con encuestas, estas proporciones se estiman mediante los estimadores ponderados explicados en secciones previas.

### 5.3.2 Tablas de doble entrada

Una tabla de contingencia puede representarse en forma general como:

| Variable 2 | Variable 1 | Marginal fila |  |
| --- | --- | --- | --- |
|  | 0 | 1 |  |
| 0 |  |  |  |
| 1 |  |  |  |
| Marginal columna |  |  |  |

Cuando se aplican los pesos muestrales, se obtienen las frecuencias ponderadas:

| Variable 2 | Variable 1 | Marginal fila |  |
| --- | --- | --- | --- |
|  | 0 | 1 |  |
| 0 |  |  |  |
| 1 |  |  |  |
| Marginal columna |  |  |  |

donde, por ejemplo, la frecuencia ponderada en la celda está dada por:

y las proporciones estimadas se calculan como:

### 5.3.3 Extensiones y aplicaciones

Aunque comúnmente se presentan como tablas bidimensionales (), las tablas de contingencia pueden extenderse a más dimensiones, incorporando una tercera variable o más ( subtablas), lo que permite explorar relaciones más complejas.

Gracias a su simplicidad y potencia descriptiva, las tablas cruzadas son ampliamente utilizadas en la investigación aplicada y en el diseño de políticas públicas, pues permiten identificar patrones y asociaciones que no serían evidentes a simple vista. Su interpretación también puede reforzarse mediante representaciones gráficas, como los diagramas de barras apiladas, que facilitan la visualización de tendencias y diferencias (véase la Sección 9.8 para un análisis más detallado).

## 5.4 Estimación de proporciones y tablas cruzadas

En el análisis de encuestas de hogares, una de las tareas más comunes consiste en estimar proporciones poblacionales a partir de variables categóricas. Estas estimaciones permiten cuantificar la prevalencia de una característica en la población, por ejemplo, el porcentaje de hogares con acceso a internet o la proporción de individuos ocupados.

### 5.4.1 Proporciones para variables binarias

La estimación de una sola proporción, , para una variable de respuesta binaria requiere únicamente una extensión directa del estimador de razón. Al recodificar las categorías de respuesta originales en una variable indicadora con valores (presencia) y (ausencia), el estimador de la media de la razón estima la proporción o prevalencia en la población de la siguiente manera (*Heeringa, S. G., 2017*):

La varianza de este estimador puede aproximarse mediante la linealización de Taylor:

Cuando la proporción estimada se aproxima a 0 o 1, los intervalos de confianza estándar pueden producir límites fuera del rango . Para solventar este problema, se proponen intervalos de confianza basados en transformaciones, como el *Wilson modificado* o el uso del *logit* de :

cuyo resultado puede transformarse de nuevo a la escala original de proporciones.

### 5.4.2 Proporciones para variables multinomiales

En encuestas, es común que las variables categóricas tengan más de dos categorías (multinomiales). En este caso, la estimación de la proporción para cada categoría sigue un esquema análogo:

### 5.4.3 Tablas cruzadas o de contingencia

Cuando se analizan dos variables categóricas de manera conjunta, el objetivo es estudiar si existe asociación entre ellas. Para ello, se construyen tablas cruzadas (o tablas de contingencia), que organizan los datos en una matriz de , donde son las categorías de la variable y las de la variable .

A diferencia de un simple conteo de casos, en encuestas se utilizan **frecuencias ponderadas** que reflejan la población total representada. Cada celda contiene la frecuencia estimada:

Los totales marginales se obtienen como:

De estas frecuencias ponderadas se derivan **proporciones conjuntas y marginales**, que permiten analizar la distribución relativa de cada cruce:

Este enfoque ofrece una visión más completa de la relación entre categorías, ya que combina la estimación de proporciones simples con el análisis conjunto de distribuciones. Por ejemplo, se puede explorar cómo se distribuye el nivel educativo (filas) según la condición laboral (columnas), identificando patrones de desigualdad o brechas estructurales.

A continuación, siguiendo con la base de ejemplo, se estima la proporción de hombres y mujeres en pobreza y no pobreza junto con su error estándar e intervalos de confianza.

prop\_sexo\_zona <- diseno %>%   
 group\_by(pobreza,Sex) %>%  
 summarise(prop = survey\_prop(vartype = c("se", "ci"))) %>%   
 data.frame()  
  
prop\_sexo\_zona

## pobreza Sex prop prop\_se prop\_low prop\_upp  
## 1 0 Female 0.5291800 0.01242026 0.5045356 0.5536829  
## 2 0 Male 0.4708200 0.01242026 0.4463171 0.4954644  
## 3 1 Female 0.5236123 0.01586237 0.4921512 0.5548870  
## 4 1 Male 0.4763877 0.01586237 0.4451130 0.5078488

Como se puede observar, el 52.3% de las mujeres y el 47.6% son pobres. Generando intervalos de confianza al 95% de (49.2%, 55.5%) para las mujeres y (44.5%, 50.7%) para los hombres.

En la librería survey existe una alternativa para estimar tablas de contingencias y es utilizando la función svyby como se muestra a continuación:

tab\_Sex\_Pobr <- svyby(formula = ~Sex, by = ~pobreza, design = diseno, FUN = svymean)  
tab\_Sex\_Pobr

## pobreza SexFemale SexMale se.SexFemale se.SexMale  
## 0 0 0.5291800 0.4708200 0.01242026 0.01242026  
## 1 1 0.5236123 0.4763877 0.01586237 0.01586237

Como se pudo observar, los argumentos que requiere la función son definir la variable a la cual se desea estimar (formula), las categorías por la cual se desea estimar (by), el diseño muestral (desing) y el parámetro que se desea estimar (FUN). Para la estimación de los intervalos de confianza se utiliza la función *confint* como sigue:

Para la estimación de los intervalos de confianza utilizar la función confint.

confint(tab\_Sex\_Pobr) %>% as.data.frame()

## 2.5 % 97.5 %  
## 0:SexFemale 0.5048367 0.5535232  
## 1:SexFemale 0.4925226 0.5547019  
## 0:SexMale 0.4464768 0.4951633  
## 1:SexMale 0.4452981 0.5074774

Los cuales coinciden con los generados anteriormente usando la funicón *group\_by*.

Otro análisis de interés relacionado con tablas de doble entrada en encuestas de hogares es estimar el porcentaje de desempleados por sexo.

tab\_Sex\_Ocupa <- svyby(formula = ~Sex, by = ~Employment,  
 design = diseno, FUN = svymean)  
tab\_Sex\_Ocupa

## Employment SexFemale SexMale se.SexFemale se.SexMale  
## Unemployed Unemployed 0.2726730 0.7273270 0.05351318 0.05351318  
## Inactive Inactive 0.7703406 0.2296594 0.02340005 0.02340005  
## Employed Employed 0.4051575 0.5948425 0.01851986 0.01851986

De la anterior salida se puede observar que, el 27.2% de las mujeres y el 72.7% de los hombres están desempleados con errores estándares para estas estimaciones de 5.3% para mujeres y hombres. cuyos intervalos de confianza se calculan a continuación:

confint(tab\_Sex\_Ocupa) %>% as.data.frame()

## 2.5 % 97.5 %  
## Unemployed:SexFemale 0.1677891 0.3775570  
## Inactive:SexFemale 0.7244773 0.8162038  
## Employed:SexFemale 0.3688592 0.4414557  
## Unemployed:SexMale 0.6224430 0.8322109  
## Inactive:SexMale 0.1837962 0.2755227  
## Employed:SexMale 0.5585443 0.6311408

Si ahora el objetivo es estimar la pobreza, pero por las distintas regiones que se tienen en la base de datos. Primero, dado que la variable *pobreza* es de tipo numérica, es necesario convertirla en factor y luego realizar la estimación con la función svyby.

tab\_region\_pobreza <- svyby(formula = ~as.factor(pobreza), by = ~Region,   
 design = diseno, FUN = svymean)  
tab\_region\_pobreza

## Region as.factor(pobreza)0 as.factor(pobreza)1  
## Norte Norte 0.6410318 0.3589682  
## Sur Sur 0.6561536 0.3438464  
## Centro Centro 0.6346152 0.3653848  
## Occidente Occidente 0.5991839 0.4008161  
## Oriente Oriente 0.5482079 0.4517921  
## se.as.factor(pobreza)0 se.as.factor(pobreza)1  
## Norte 0.05547660 0.05547660  
## Sur 0.04348901 0.04348901  
## Centro 0.07858599 0.07858599  
## Occidente 0.04670473 0.04670473  
## Oriente 0.08849644 0.08849644

De lo anterior se puede concluir que, en la región Norte, el 35% de las personas están en estado de pobreza mientras que en el sur es el 34%. La pobreza más alta se tiene en la región oriente con un 45% de pobres. Los errores estándares de las estimaciones.

### 5.4.4 Prueba de independencia

Una **prueba de hipótesis** es un procedimiento estadístico que permite evaluar la validez de una afirmación sobre una población, comparando la evidencia empírica de una muestra con lo que se esperaría si dicha afirmación fuera cierta. En el caso de las **pruebas de independencia**, se plantea como hipótesis nula () que dos variables categóricas son independientes, es decir, que la distribución de una no depende de las categorías de la otra. Matemáticamente, esta hipótesis puede expresarse como:

donde representa la proporción esperada en la celda bajo independencia, y y las proporciones marginales de las filas y columnas, respectivamente.

En consecuencia, la prueba consiste en contrastar las proporciones estimadas con las esperadas . Cuando las diferencias son pequeñas, los datos respaldan la hipótesis nula; en cambio, discrepancias significativas conducen al rechazo de .

#### 5.4.4.1 Prueba clásica

Para una tabla de , la estadística de Pearson se define como:

donde corresponde a las proporciones esperadas bajo independencia.

En términos prácticos, las **tablas de contingencia ponderadas** constituyen el insumo de la prueba, ya que cada celda estima el número de personas en la población en esa combinación de categorías. A partir de ellas se calculan frecuencias relativas y se evalúa la independencia de las variables.

#### 5.4.4.2 Ajustes para encuestas: prueba de Rao-Scott

En encuestas de hogares, el análisis es más complejo porque el diseño muestral (estratificación, conglomerados y ponderación) altera la varianza de los estimadores. Por ello, la prueba debe ajustarse para reflejar estas particularidades.

El **ajuste de Rao-Scott** (Rao y Scott, 1984; Thomas y Rao, 1987) corrige la prueba clásica mediante el **efecto de diseño generalizado (GDEFF)**, lo que produce un estadístico robusto:

Bajo , sigue aproximadamente una distribución con grados de libertad. En casos de **muestras pequeñas**, es común aplicar ajustes basados en la distribución F, que mejoran la precisión de los resultados. Actualmente, las pruebas de Rao-Scott constituyen el estándar en el análisis de datos categóricos en encuestas complejas (Heeringa, West y Berglund, 2017).

#### 5.4.4.3 Ejemplo en R

El paquete **survey** en R implementa estas pruebas a través de la función svychisq(), que aplica automáticamente las correcciones de Rao-Scott. A modo de ejemplo, para evaluar si la pobreza es independiente del sexo, se ejecuta:

svychisq(formula = ~Sex + pobreza, design = diseno, statistic="F")

##   
## Pearson's X^2: Rao & Scott adjustment  
##   
## data: NextMethod()  
## F = 0.056464, ndf = 1, ddf = 119, p-value = 0.8126

* Si el **p-valor** es superior al 5%, no se rechaza , concluyendo que no existe evidencia de asociación entre pobreza y sexo.
* Si el **p-valor** es inferior al 5%, se rechaza , lo que sugiere dependencia entre las variables.

De manera similar, se pueden evaluar otras relaciones, como desempleo y sexo o pobreza y región:

svychisq(formula = ~Sex + Employment, design = diseno, statistic = "F")

##   
## Pearson's X^2: Rao & Scott adjustment  
##   
## data: NextMethod()  
## F = 62.251, ndf = 1.6865, ddf = 200.6978, p-value < 2.2e-16

svychisq(formula = ~Region + pobreza, design = diseno, statistic = "F")

##   
## Pearson's X^2: Rao & Scott adjustment  
##   
## data: NextMethod()  
## F = 0.48794, ndf = 3.0082, ddf = 357.9731, p-value = 0.6914

Para realizar la prueba de independencia en R, se utilizará la función *svychisq* del paquete *srvyr*. Esta función requiere que se definan las variables de interés (formula) y requiere que se le defina el diseño muestral (desing). Ahora, para ejemplificar el uso de esta función tomaremos la base de datos de ejemplo y se probará si la pobreza es independiente del sexo.

**Razón de odds**

Como lo menciona *Monroy, L. G. D. (2018)* La traducción más aproximada del término odds es “la ventaja”, en términos de probabilidades es la posibilidad de que un evento ocurra con relación a que no ocurra, es decir, es un número que expresa cuánto más probable es que se produzca un evento frente a que no se produzca. También se puede utilizar para cuantificar la asociación entre los niveles de una variable y un factor categórico *(Heeringa, S. G. 2017)*.

Suponga que se desea calcular la siguiente razón de odds.

El procedimiento para realizarlo en R sería, primero estimar las proporciones de la tabla cruzada entre las variables sexo y pobreza:

tab\_Sex\_Pobr <- svymean(x = ~interaction (Sex, pobreza), design = diseno,   
 se=T, na.rm=T, ci=T, keep.vars=T)  
  
tab\_Sex\_Pobr %>% as.data.frame()

## mean SE  
## interaction(Sex, pobreza)Female.0 0.3218703 0.01782709  
## interaction(Sex, pobreza)Male.0 0.2863733 0.01768068  
## interaction(Sex, pobreza)Female.1 0.2051285 0.01659697  
## interaction(Sex, pobreza)Male.1 0.1866279 0.01778801

Luego, se realiza el contraste dividiendo cada uno de los elementos de la expresión mostrada anteriormente:

svycontrast(stat = tab\_Sex\_Pobr,   
contrasts = quote((`interaction(Sex, pobreza)Female.0`/`interaction(Sex, pobreza)Female.1`) /(`interaction(Sex, pobreza)Male.0`/ `interaction(Sex, pobreza)Male.1`)))

## nlcon SE  
## contrast 1.0226 0.0961

Obtiendo que, se estima que el odds de las mujeres que no están en estado de pobreza es 1.02 comparandolo con el odds de los hombres. En otras palabras, se estima que las probabilidades de que las mujeres no estén en estado de pobreza sin tener en cuenta ninguna otra variable de la encuesta es cera de 2% mayor que las probabilidades de los hombres.

**Diferencia de proporciones en tablas de contingencias**

Como lo menciona *Heeringa, S. G. (2017)* las estimaciones de las proporciones de las filas en las tablas de doble entrada son, de hecho, estimaciones de subpoblaciones en las que la subpoblación se define por los niveles de la variable factorial. Ahora bien, si el interés se centra en estimar diferencias de las proporciones de las categorías entre dos niveles de una variable factorial, se pueden utilizando contrastes.

A manera de ejemplo, se requiere estimar ahora, el contraste de proporciones de mujeres en estado de pobreza versus los hombres en esta misma condición (). Para ellos, primero, estimemos la proporción de hombres y mujeres en estado de pobreza como se ha mostrado en capítulos anteriores:

(tab\_sex\_pobreza <- svyby(formula = ~pobreza, by = ~Sex,   
 design = diseno , svymean, na.rm=T,  
 covmat = TRUE, vartype = c("se", "ci")))

## Sex pobreza se ci\_l ci\_u  
## Female Female 0.3892389 0.03159581 0.3273123 0.4511656  
## Male Male 0.3945612 0.03662762 0.3227724 0.4663501

Ahora bien, para calcular la estimación de la diferencia de proporciones junto con sus errores estándares, se realizarán los siguientes pasos:

* *Paso 1:* Calcular la diferencia de estimaciones

0.3892 - 0.3946

## [1] -0.0054

Con la función vcov se obtiene la matriz de covarianzas:

library(kableExtra)  
vcov(tab\_sex\_pobreza)%>% data.frame() %>%   
 kable(digits = 10,  
 format.args = list(scientific = FALSE))

|  | Female | Male |
| --- | --- | --- |
| Female | 0.0009982953 | 0.0009182927 |
| Male | 0.0009182927 | 0.0013415823 |

* *Paso 2:* El cálculo del error estándar es:

sqrt(0.0009983 + 0.0013416 - 2\*0.0009183)

## [1] 0.02243435

Ahora bien, aplicando la función svycontrast se puede obtener la estimación de la diferencia de proporciones anterior:

svycontrast(tab\_sex\_pobreza,  
 list(diff\_Sex = c(1, -1))) %>%  
 data.frame()

## contrast diff\_Sex  
## diff\_Sex -0.005322297 0.02243418

De lo que se concluye que, la diferencia entre las proporciones de mujeres y hombres en estado de pobreza es -0.005 (-0.5%) con una desviación estándar de 0.022.

Otro ejercicio de interés en un análisis de encuestas de hogares es verificar la diferencia del desempleo por sexo. Al igual que el ejemplo anterior, se inicia con la estimación del porcentaje de desempleados por sexo:

tab\_sex\_desempleo <- svyby(formula = ~desempleo, by = ~Sex,   
 design = diseno %>% filter(!is.na(desempleo)) ,   
 FUN = svymean, na.rm=T, covmat = TRUE,  
 vartype = c("se", "ci"))  
tab\_sex\_desempleo

## Sex desempleo se ci\_l ci\_u  
## Female Female 0.02168620 0.005580042 0.01074952 0.03262288  
## Male Male 0.06782601 0.012161141 0.04399062 0.09166141

Para calcular la estimación de la diferencia de proporciones junto con sus errores estándares, se realizarán los siguientes pasos:

* *Paso 1*: Diferencia de las estimaciones

0.02169 - 0.06783

## [1] -0.04614

Estimación de la matriz de covarianza:

vcov(tab\_sex\_desempleo) %>% data.frame() %>%   
 kable(digits = 10,  
 format.args = list(scientific = FALSE))

|  | Female | Male |
| --- | --- | --- |
| Female | 0.0000311369 | 0.0000208130 |
| Male | 0.0000208130 | 0.0001478933 |

* *Paso 2*: Estimación del error estándar.

sqrt(0.00003114 + 0.00014789 - 2\*0.00002081)

## [1] 0.0117222

Siguiendo el ejemplo anterior, utilizando la función svycontrast se tiene que:

svycontrast(tab\_sex\_desempleo,  
 list(diff\_Sex = c(-1, 1))) %>%  
 data.frame()

## contrast diff\_Sex  
## diff\_Sex 0.04613982 0.01172195

de lo que se concluye que, la estimación del contraste es 0.04 (4%) con un error estándar de 0.011.

Otro ejercicio que se puede realizar en una encuesta de hogares es ahora estimar la proporción de desempleados por región. Para la realización de este ejercicio, se seguirán los pasos de los dos ejemplos anteriores:

tab\_region\_desempleo <- svyby(formula = ~desempleo, by = ~Region,   
 design = diseno %>% filter(!is.na(desempleo)) ,   
 FUN = svymean, na.rm=T, covmat = TRUE,  
 vartype = c("se", "ci"))  
tab\_region\_desempleo

## Region desempleo se ci\_l ci\_u  
## Norte Norte 0.04877722 0.02002293 0.009532997 0.08802144  
## Sur Sur 0.06563877 0.02375124 0.019087202 0.11219034  
## Centro Centro 0.03873259 0.01240317 0.014422832 0.06304235  
## Occidente Occidente 0.03996523 0.01229650 0.015864529 0.06406592  
## Oriente Oriente 0.02950231 0.01256905 0.004867428 0.05413719

Ahora, el interés es realizar los contrastes siguientes para desempleo:

,

Escrita de forma matricial sería:

La matriz de varianzas y covarianzas es:

vcov(tab\_region\_desempleo)%>%  
 data.frame() %>%   
 kable(digits = 10,  
 format.args = list(scientific = FALSE))

Por tanto, la varianza estimada está dada por:

sqrt(0.0002981 + 0.0002884 - 2\*0)

## [1] 0.02421776

sqrt(0.0001968 + 0.0002884 - 2\*0)

## [1] 0.02202726

sqrt(0.0001267 + 0.0004093 - 2\*0)

## [1] 0.02315167

Usando la función svycontrast, la estimación de los contrastes sería:

svycontrast(tab\_region\_desempleo, list(  
 Norte\_sur = c(1, 0, -1, 0, 0),  
 Sur\_centro = c(0, 1, -1, 0, 0),  
 Occidente\_Oriente = c(0, 0, 0, 1, -1))) %>% data.frame()

## contrast SE  
## Norte\_sur 0.01004463 0.02355327  
## Sur\_centro 0.02690618 0.02679477  
## Occidente\_Oriente 0.01046292 0.01758365

Por último, repitiendo el contraste anterior y los pasos para resolverlo, pero ahora utilizando la variable pobreza se tiene:

tab\_region\_pobreza <- svyby(formula = ~pobreza, by = ~Region,   
 design = diseno %>% filter(!is.na(desempleo)) ,   
 FUN = svymean, na.rm=T, covmat = TRUE,  
 vartype = c("se", "ci"))  
tab\_region\_pobreza

## Region pobreza se ci\_l ci\_u  
## Norte Norte 0.3262813 0.04800361 0.2321959 0.4203666  
## Sur Sur 0.2946736 0.04794292 0.2007072 0.3886400  
## Centro Centro 0.3233923 0.07211854 0.1820426 0.4647421  
## Occidente Occidente 0.3673286 0.04400234 0.2810856 0.4535716  
## Oriente Oriente 0.3870632 0.09160150 0.2075276 0.5665989

El interés se centra en realizar los contrastes siguientes para pobreza:

,

Escrita de forma matricial es:

Y, utilizando la función svycontrast se obtiene:

svycontrast(tab\_region\_pobreza, list(  
 Norte\_sur = c(1, 0, -1, 0, 0),  
 Sur\_centro = c(0, 1, -1, 0, 0),  
 Occidente\_Oriente = c(0, 0, 0, 1, -1))) %>% data.frame()

## contrast SE  
## Norte\_sur 0.002888908 0.08663389  
## Sur\_centro -0.028718759 0.08660027  
## Occidente\_Oriente -0.019734641 0.10162205

## 5.5 Modelos de regresión en encuestas de hogares

La regresión estadística constituye una técnica fundamental para examinar los vínculos entre variables en el marco de los datos muestrales obtenidos mediante encuestas. A través de este procedimiento, es posible determinar la forma en que una o varias variables de respuesta (dependientes) se relacionan con una o varias variables explicativas (independientes). Tal como exponen Nolan y Speed (2000) y Freedman (2005), la validez de los resultados depende de una adecuada formulación del modelo.

Un ejemplo ilustrativo sería la estimación del ingreso de los hogares (variable dependiente) en función del nivel educativo alcanzado y de la situación laboral de sus miembros (variables independientes), empleando datos provenientes de encuestas de hogares. Estos análisis permiten identificar patrones, cuantificar efectos y generar evidencia útil para la formulación de políticas públicas.

No obstante, dado que estas encuestas suelen estar sustentadas en **diseños muestrales complejos**, los enfoques tradicionales de regresión resultan insuficientes. Ignorar los pesos muestrales, la estratificación o la conglomeración puede derivar en sesgos en los coeficientes estimados y, sobre todo, en una subestimación de sus varianzas, lo cual compromete la validez de las inferencias. Por esta razón, los modelos de regresión aplicados a encuestas deben ser modificados y ajustados para garantizar resultados representativos y robustos.

En este sentido, el análisis de datos provenientes de encuestas exige una atención detallada al diseño de muestreo. La incorporación de los **pesos de la encuesta** y de los ajustes correspondientes a la estratificación y a la conglomeración permite obtener inferencias válidas y precisas. Además, en algunos casos se han planteado alternativas simplificadas, como el uso de pesos normalizados o enfoques de ponderación aproximada, que buscan balancear la complejidad metodológica con la factibilidad práctica del análisis.

### 5.5.1 Antecedentes históricos y desarrollos metodológicos

El estudio de la regresión bajo diseños de muestreo complejos tiene una trayectoria bien documentada. De manera empírica, **Kish y Frankel (1974)** fueron de los primeros en discutir el impacto de estos diseños en las inferencias derivadas de modelos de regresión. Posteriormente, **Fuller (1975)** desarrolló un estimador de varianza apoyado en técnicas de linealización para modelos de regresión lineal múltiple con ponderación desigual, e introdujo métodos específicos para diseños estratificados y de dos etapas.

Más adelante, **Sha et al. (1977)** abordaron el problema de las violaciones a los supuestos clásicos de los modelos de regresión al trabajar con datos de encuestas, proponiendo alternativas de inferencia robusta para los parámetros. En paralelo, **Binder (1983)** se enfocó en las distribuciones muestrales de los estimadores de regresión en poblaciones finitas, definiendo procedimientos para estimar varianzas bajo esquemas complejos.

En los años siguientes, **Skinner et al. (1989)** ampliaron estos aportes al trabajar con estimadores de varianza para los coeficientes de regresión que contemplaban la estratificación y la conglomeración, recomendando explícitamente el uso de métodos de linealización o técnicas alternativas para la estimación de la varianza. Avanzando en la línea de tiempo, **Fuller (2002)** realizó un compendio de los métodos de estimación aplicables a modelos de regresión en encuestas complejas, mientras que **Pfeffermann (2011)** discutió enfoques más recientes, como los métodos de ponderación “q-weighted”, mostrando evidencia empírica de su utilidad.

### 5.5.2 Relevancia práctica

En la actualidad, los modelos de regresión bajo diseños de muestreo complejos representan una herramienta esencial para el análisis de encuestas de hogares. Estos modelos permiten ir más allá de las estadísticas descriptivas y aproximarse a explicaciones causales o predictivas, siempre que se reconozcan y se ajusten las particularidades del diseño muestral. Su correcta aplicación abre la posibilidad de analizar cómo las características sociodemográficas y económicas se asocian con distintos resultados de interés, aportando evidencia clave para la formulación de políticas públicas.

En las siguientes secciones se mostrará cómo implementar estos modelos en R, utilizando la librería survey. Se abordará la especificación del diseño muestral, la estimación de modelos lineales y logísticos, así como la obtención de errores estándar y pruebas de hipótesis ajustadas al diseño. De este modo, se integrarán los fundamentos teóricos con ejemplos prácticos en un flujo de trabajo reproducible.

Un primer paso consiste en comprender la estructura básica de los modelos de regresión. El modelo de regresión lineal simple se define como

donde es la variable dependiente, la variable independiente, y los parámetros del modelo, y el error aleatorio, definido como la diferencia entre el valor observado y el valor ajustado del modelo:

Generalizando este planteamiento, los modelos de regresión lineal múltiple incorporan varias covariables:

lo cual puede expresarse en notación matricial como

donde corresponde al vector de covariables del individuo , y es el vector de parámetros.

En este contexto, el valor esperado de la variable respuesta condicionado a las covariables puede escribirse como:

Para que estos modelos sean válidos, es necesario que se cumplan ciertos **supuestos clásicos**, recogidos en la literatura (Heeringa, West y Berglund, 2017), entre los que destacan:

* **Esperanza nula de los residuos**: .
* **Homogeneidad de varianza**: .
* **Normalidad de los errores**: .
* **Independencia de los residuos**: .

Estos supuestos permiten garantizar que los estimadores obtenidos tengan buenas propiedades estadísticas (insesgamiento, eficiencia y consistencia). Sin embargo, al trabajar con encuestas bajo **diseños complejos**, estas condiciones rara vez se cumplen de manera estricta, por lo que se requieren adaptaciones que serán abordadas en las próximas secciones.

Una vez definido el modelo de regresión lineal y sus supuestos, se puede deducir los siguiente:

y Adicionalmente,

### 5.5.3 ¿Aplicar o no aplicar ponderaciones?

Heeringa, West y Berglund (2017) examinan el desafío de determinar cómo utilizar adecuadamente los pesos en modelos de regresión y si conviene emplear factores de expansión al estimar coeficientes de regresión en encuestas con diseños complejos. En este marco, se distinguen dos enfoques principales para incorporar los pesos en los modelos:

* **Enfoque orientado al diseño**: busca realizar inferencias válidas sobre la población total. Los pesos de la encuesta resultan indispensables para obtener estimaciones insesgadas de los coeficientes, ya que corrigen las probabilidades desiguales de selección derivadas del diseño muestral. No obstante, este método no protege frente a la mala especificación del modelo: si la relación planteada no refleja adecuadamente lo que ocurre en la población, los coeficientes estimados, aunque insesgados en el marco del diseño, pueden carecer de utilidad sustantiva.
* **Enfoque orientado al modelo**: sostiene que los pesos no son necesarios siempre que el modelo esté correctamente formulado y el muestreo sea no informativo, es decir, que el modelo válido para la muestra coincida con el de la población. En este escenario, se asume que las relaciones entre variables están bien descritas por el modelo independientemente del diseño muestral, y que la utilización de ponderaciones podría incrementar innecesariamente la variabilidad de las estimaciones, elevando los errores estándar.

La decisión entre utilizar o no ponderaciones en los modelos de regresión depende tanto del contexto como de la sensibilidad de los resultados a su inclusión. Autores como **Skinner, Holt y Smith (1989)** y **Pfeffermann (2011)** han debatido ampliamente sobre la pertinencia de incorporar los pesos muestrales en la estimación de los parámetros de regresión y en sus errores estándar.

Una recomendación metodológica ampliamente aceptada es estimar los modelos con y sin ponderaciones y comparar los resultados. Si al incluir los pesos se observan variaciones significativas en los coeficientes o en las conclusiones, ello indica que el muestreo fue informativo o que el modelo presenta deficiencias de especificación, por lo que conviene utilizar estimaciones ponderadas. En cambio, si los pesos solo aumentan los errores estándar sin modificar sustancialmente los coeficientes, se puede asumir que el modelo está bien planteado y que no es indispensable ponderar.

En términos prácticos, la decisión puede resumirse en dos escenarios:

* **Inferencia descriptiva**: es obligatorio aplicar ponderaciones, ya que el objetivo es reflejar con precisión la estructura de la población.
* **Inferencia analítica**: es posible recurrir a modelos no ponderados o ajustados por pesos. En este caso, si la meta es analizar relaciones o verificar hipótesis, la ponderación no siempre es necesaria, especialmente cuando el modelo incluye variables del diseño muestral (estratos o conglomerados). Sin embargo, el uso de modelos sin ponderar debe justificarse de forma explícita, pues implica supuestos más restrictivos que los modelos ponderados.

El uso de ponderaciones en encuestas permite asegurar que los modelos de regresión sean representativos de la población, ya que corrigen posibles sesgos de sobre o subrepresentación de determinados grupos y garantizan que la distribución poblacional se refleje adecuadamente. Asimismo, las ponderaciones contribuyen a obtener estimaciones de varianza más exactas, pues consideran la estratificación, el agrupamiento y las probabilidades desiguales de selección, lo cual genera errores estándar, intervalos de confianza y pruebas estadísticas más confiables.

Dentro del enfoque basado en el diseño, los coeficientes de regresión se estiman a partir de ecuaciones poblacionales ajustadas con ponderaciones. Esto permite que los resultados ponderados se aproximen a valores insesgados comparables a los que se obtendrían en un censo completo, incluso cuando el modelo estadístico no esté formulado de manera óptima.

Un aspecto que no debe pasarse por alto es que el uso de ponderaciones puede aumentar la varianza de las estimaciones de los parámetros, en especial cuando los pesos presentan gran dispersión. En situaciones donde existen valores extremos o muy variables, las estimaciones tienden a volverse inestables, ya que ciertas observaciones llegan a ejercer una influencia desproporcionada sobre el ajuste del modelo. En este sentido, cuando el propósito es explicativo o analítico (como en el análisis de relaciones entre variables), los modelos sin ponderar pueden, en ocasiones, generar resultados más consistentes y eficientes.

No obstante, cuando el modelo está mal especificado, la regresión sin ponderaciones puede producir estimaciones poco útiles o carentes de validez. Por ello, resulta fundamental que los analistas seleccionen e incorporen las variables pertinentes para lograr una especificación adecuada. Incluso en los casos en que el modelo esté correctamente definido, es indispensable tener en cuenta la estratificación y la conglomeración del diseño muestral al calcular los errores estándar bajo un enfoque no ponderado.

En definitiva, la decisión sobre **aplicar o no ponderaciones** no puede basarse únicamente en una regla rígida, sino que exige un análisis crítico de los objetivos del estudio, del diseño de la encuesta y de la robustez de los modelos estimados. Asimismo, un análisis diagnóstico riguroso resulta esencial para validar las inferencias y garantizar que los resultados reflejen de manera adecuada la realidad poblacional (véase la Subsección 9.6.4).

### 5.5.4 Enfoques inferenciales para el análisis de datos

En el análisis de encuestas, uno de los principales retos consiste en manejar adecuadamente la variabilidad de los datos. Esta proviene de dos fuentes fundamentales: el diseño muestral, que determina cómo se recolecta la información, y el modelo estadístico, que define cómo se interpreta dicha información para inferir propiedades de la población. Ignorar cualquiera de estas dimensiones puede comprometer la validez de los resultados.

Por ello, se han desarrollado metodologías inferenciales avanzadas que permiten integrar ambas fuentes de incertidumbre en un mismo marco analítico. Estas técnicas buscan reflejar tanto la estructura del diseño como los supuestos y limitaciones del modelo. Entre las aproximaciones más relevantes se encuentran la seudo-verosimilitud (Molina & Skinner, 1992) y la inferencia combinada (Binder, 2011).

El método de seudo-verosimilitud extiende las técnicas tradicionales de máxima verosimilitud para ajustarlas a las particularidades de los diseños muestrales complejos. En este enfoque, la distribución de muestreo definida por el diseño tiene un rol central, mientras que la distribución del modelo ocupa un lugar secundario. Si bien, en contextos de modelos bien especificados, los estimadores basados en seudo-verosimilitud tienden a ser insesgados o consistentes, lo crucial es que este procedimiento evita sesgos que se originarían si se ignorara el diseño muestral. En términos prácticos, la seudo-verosimilitud traduce el modelo tradicional en uno que respete la forma en que los datos fueron obtenidos, garantizando inferencias más sólidas.

En contraste, la inferencia combinada propone un marco unificado en el que se integran simultáneamente la variabilidad del muestreo y la incertidumbre del modelo. Al considerar ambas fuentes, este enfoque ofrece una visión más completa de la variabilidad y permite obtener estimaciones más precisas y confiables. Su principal aporte radica en que evita sesgos que pueden aparecer cuando se analiza únicamente desde la perspectiva del diseño o del modelo. De esta manera, la inferencia combinada resulta especialmente útil en aplicaciones donde se requiere un balance entre representatividad poblacional y solidez estadística de los modelos ajustados.

## 5.6 Estimación de los parámetros en un modelo de regresión con muestras complejas

Una vez se establecen los supuestos del modelo y las características distribucionales de los errores, el paso siguiente es el proceso de estimación de los parámetros. A modo ilustrativo, si en lugar de observar una muestra de tamaño de los elementos de la población se hubiera realizado un censo completo, el parámetro de regresión de población finita podría calcularse como sigue **(Téllez et al., 2016)**:

Sin embargo, cuando se desea estimar los parámetros de un modelo de regresión lineal utilizando información proveniente de encuestas con diseños complejos, el enfoque estándar cambia. La principal razón es que los datos no cumplen con los supuestos clásicos de independencia e idéntica distribución, ya que el diseño muestral introduce elementos como estratificación, conglomerados o probabilidades de selección desiguales. En consecuencia, los estimadores tradicionales (como los obtenidos por máxima verosimilitud o mínimos cuadrados) pueden resultar sesgados y sus errores estándar poco confiables.

En este contexto, **Wolter (2007)** propone el uso de métodos no paramétricos robustos, como la linealización de Taylor o los procedimientos de replicación (Jackknife, Bootstrap, Balanced Repeated Replication, entre otros), para estimar varianzas y obtener inferencias válidas.

### 5.6.1 Estimación de parámetros

En el caso de un modelo de regresión lineal simple, la estimación de la pendiente bajo un esquema de encuesta compleja se realiza mediante un estimador ponderado, que puede expresarse de la siguiente forma:

donde representa los pesos de muestreo. La principal diferencia respecto al estimador clásico es la incorporación explícita de dichos pesos, que corrigen las probabilidades desiguales de selección y aseguran la representatividad de las estimaciones.

La varianza del estimador puede expresarse como:

Este enfoque permite cuantificar la precisión del coeficiente considerando tanto los pesos como la estructura del diseño muestral.

En el caso de la regresión múltiple, el cálculo de la varianza de cada coeficiente se realiza considerando su interdependencia con los demás. Esto se refleja en la construcción de una **matriz de varianza-covarianza**, que recoge tanto la variabilidad individual como las covarianzas entre todos los parámetros. De acuerdo con **Kish y Frankel (1974)**, esta estimación requiere utilizar totales ponderados de cuadrados y productos cruzados de todas las combinaciones entre la variable dependiente y el conjunto de predictores . En términos generales:

Este marco permite evaluar la estabilidad y precisión de los parámetros en modelos más complejos, garantizando que las inferencias reflejen adecuadamente tanto el diseño de la encuesta como la relación entre las variables.

Para ejemplificar los conceptos trabajados hasta este momento, se tomará la misma base que se ha venido trabajando durante todo el desarrollo de este libro. Se inicia con el cargue de las librerías, la base de datos y la definición del diseño de muestreo:

knitr::opts\_chunk$set(warning = FALSE, message = FALSE, error = FALSE)  
options(digits = 4)  
options(tinytex.verbose = TRUE)  
library (survey)  
library(srvyr)  
library(convey)  
library(TeachingSampling)  
library(printr)  
library(stargazer)  
library(jtools)  
library(broom)

Cargue de la base y definición del diseño muestral:

data(BigCity, package = "TeachingSampling")  
library(tidyverse)  
  
encuesta <- readRDS("Data/encuesta.rds")  
head(encuesta)

| HHID | Stratum | NIh | nIh | dI | PersonID | PSU | Zone | Sex | Age | MaritalST | Income | Expenditure | Employment | Poverty | dki | dk | wk | Region | CatAge |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| idHH00031 | idStrt001 | 9 | 2 | 4.5 | idPer01 | PSU0003 | Rural | Male | 68 | Married | 409.9 | 346.3 | Employed | NotPoor | 8 | 36 | 34.50 | Norte | Más de 60 |
| idHH00031 | idStrt001 | 9 | 2 | 4.5 | idPer02 | PSU0003 | Rural | Female | 56 | Married | 409.9 | 346.3 | Employed | NotPoor | 8 | 36 | 33.64 | Norte | 46-60 |
| idHH00031 | idStrt001 | 9 | 2 | 4.5 | idPer03 | PSU0003 | Rural | Female | 24 | Married | 409.9 | 346.3 | Employed | NotPoor | 8 | 36 | 33.64 | Norte | 16-30 |
| idHH00031 | idStrt001 | 9 | 2 | 4.5 | idPer04 | PSU0003 | Rural | Male | 26 | Married | 409.9 | 346.3 | Employed | NotPoor | 8 | 36 | 34.50 | Norte | 16-30 |
| idHH00031 | idStrt001 | 9 | 2 | 4.5 | idPer05 | PSU0003 | Rural | Female | 3 | NA | 409.9 | 346.3 | NA | NotPoor | 8 | 36 | 33.64 | Norte | 0-5 |
| idHH00041 | idStrt001 | 9 | 2 | 4.5 | idPer01 | PSU0003 | Rural | Female | 61 | Widowed | 823.8 | 392.2 | Employed | NotPoor | 8 | 36 | 33.64 | Norte | Más de 60 |

library(srvyr)  
diseno <- encuesta %>%  
 as\_survey\_design(  
 strata = Stratum,  
 ids = PSU,  
 weights = wk,  
 nest = T  
 )

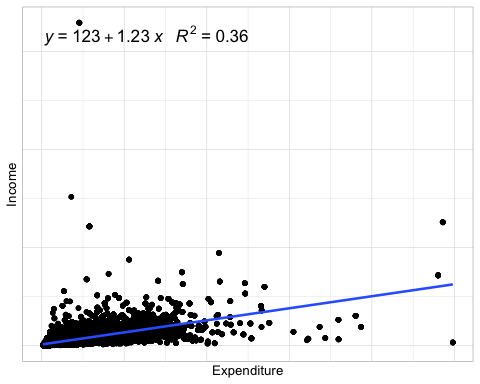
Para efectos de los ejemplos y como se ha hecho en anteriores ocasiones, se divide la muestra en sub-grupos de la encuesta como sigue:

sub\_Urbano <- diseno %>% filter(Zone == "Urban")  
sub\_Rural <- diseno %>% filter(Zone == "Rural")  
sub\_Mujer <- diseno %>% filter(Sex == "Female")  
sub\_Hombre <- diseno %>% filter(Sex == "Male")

En este capítulo se ajustarán los modelos de regresión usando la base de datos de ejemplo que se ha venido trabajando en capítulos anteriores. Puesto que, en modelos de regresión, se utiliza muy frecuente el recurso gráfico. A continuación, se define un tema estándar que la CEPAL tiene para generar sus gráficos el cual se utilizará en este capítulo.

Para observar que existe una correlación entre el ingreso y el gasto, las cuales son las variables que se utilizarán para el ajuste de los modelos, se construye un scatterplot usando la librería ggplot. Cabe resaltar que, como la base de datos encuesta, la cual se usa para ejemplificar es una muestra de la base BigCity, analizaremos de manera gráfica si poblacionalmente las dos variables mencionadas anteriormente tienen correlación como se muestra a continuación:

library(ggplot2); library(ggpmisc)  
plot\_BigCity <- ggplot(data = BigCity,  
 aes(x = Expenditure, y = Income)) +  
 geom\_point() + geom\_smooth(method = "lm",  
 se = FALSE,  
 formula = y ~ x) + theme\_cepal()  
  
plot\_BigCity + stat\_poly\_eq(formula = y~x, aes(label = paste(..eq.label..,  
 ..rr.label.., sep = "~~~"),size = 3), parse = TRUE)



Si bien, existen unas observaciones por fuera de la nube de punto, el comportamiento general de la relación ingresos vs gastos mantiene una tendencia lineal.

Una vez hecho el análisis gráfico de las variables a utilizar en los modelos a trabajar, se realizará primero un ajuste del modelo con los datos poblacionales y con esto poder analizar qué tan bueno serán los ajustes que se realizarán posteriormente. A continuación, se muestra el ajuste del modelo con los datos poblacionales:

fit <- lm(Income ~ Expenditure, data = BigCity)

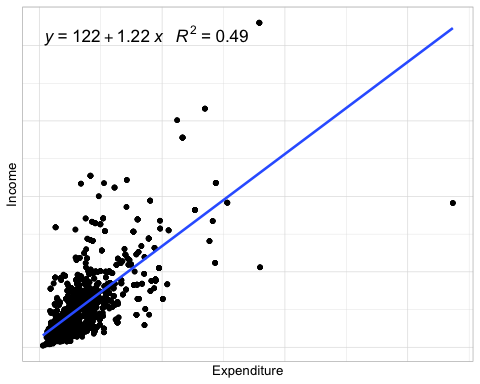
Ahora bien, para observar los parámetros poblacionales del modelo se utilizará la función modelsummary de la librería modelsummary de la siguiente manera:

library(knitr)  
   
tabla <- data.frame(  
 Term = c("(Intercept)", "Expenditure", "Num.Obs.", "R2", "R2 Adj.", "RMSE"),  
 Pob = c(123.337, 1.229, 150266, 0.359, 0.359, 461.74)  
)  
   
kable(tabla,   
 format = "latex",   
 booktabs = TRUE,   
 digits = 3,  
 col.names = c("", "Pob"))

De la anterior salida se puede observar que, el intercepto es igual a 123.337 y el parámetro asociado al gasto es 1.229. La demás información relacionada a esta salida se analizará más adelante.

Una vez revisada la información poblacional, se utilizará la información obtenida de la muestra para estimar los parámetros y con ello analizar qué tan buenas son las estimaciones. A continuación, se presenta una sintaxis similar a la anterior que permite construir el scatterplot pero para los datos de la muestra.

plot\_sin <- ggplot(data = encuesta,  
 aes(x = Expenditure, y = Income)) +  
 geom\_point() +  
 geom\_smooth(method = "lm",  
 se = FALSE, formula = y ~ x) + theme\_cepal()  
  
plot\_sin + stat\_poly\_eq(formula = y~x, aes(label = paste(..eq.label..,  
 ..rr.label.., sep = "~~~"), size = 5), parse = TRUE)



Como se puede observar, los datos de la muestra tienen una tendencia lineal aunque un poco dispersa a medida que crecen los gastos en las familias.

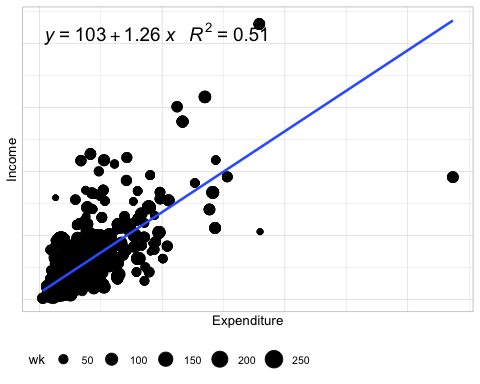
Una vez hecho los análisis gráficos se procede a ajustar los modelos de regresión lineal. A modo de comparar el efecto que tiene hacer un correcto uso de los factores de expansión del diseño, primero, se ajustará un modelo sin tener encuesta dichos factores como se muestra a continuación:

fit\_sinP <- lm(Income ~ Expenditure, data = encuesta)  
stargazer(fit\_sinP, type = "latex",  
 title = "Modelo sin factores de expansion",  
 label = "tab:sin\_factores",  
 header = FALSE)

Para el modelo ajustado sin factores de expansión, el es 121.52 y el asociado a la variable gastos es 1.22.

Ahora, haciendo un Scatterplot con los datos encuesta pero utilizando los factores de expansión del diseño se debe agregar mapping = aes(weight = wk) en la función geom\_smoothcomo sigue:

plot\_Ponde <- ggplot(data = encuesta,  
 aes(x = Expenditure, y = Income)) +  
 geom\_point(aes(size = wk)) +  
 geom\_smooth(method = "lm", se = FALSE, formula = y ~ x, mapping = aes(weight = wk)) + theme\_cepal()  
  
plot\_Ponde + stat\_poly\_eq(formula = y~x, aes(weight = wk,  
label = paste(..eq.label..,..rr.label.., sep = "~~~")), parse = TRUE,size = 5)



En este sentido, para ajustar modelos teniendo en cuenta los factores de expansión existen 2 formas, la primera es usando la función lm y la segunda es usando la función svyglm de la librería survey. A continuación. se ajusta el modelo usando la función lm:

fit\_Ponde <- lm(Income ~ Expenditure, data = encuesta, weights = wk)  
stargazer(fit\_Ponde, header = FALSE,  
 title = "Modelo encuesta ponderada",  
 type = "latex")

Para el modelo ajustado con factores de expansión usando la función lm, el es 103.14 y el asociado a la variable gastos es 1.26. Ahora, haciendo el mismo ajuste pero usando la función svyglm:

fit\_svy <- svyglm(Income ~ Expenditure,  
 design = diseno, family=stats::gaussian())  
fit\_svy

## Stratified 1 - level Cluster Sampling design (with replacement)  
## With (238) clusters.  
## Called via srvyr  
## Sampling variables:  
## - ids: PSU   
## - strata: Stratum   
## - weights: wk   
##   
## Call: svyglm(formula = Income ~ Expenditure, design = diseno, family = stats::gaussian())  
##   
## Coefficients:  
## (Intercept) Expenditure   
## 103.14 1.26   
##   
## Degrees of Freedom: 2604 Total (i.e. Null); 118 Residual  
## Null Deviance: 6.35e+08   
## Residual Deviance: 3.11e+08 AIC: 38300

Obteniendo estimaciones para el es 103.14 y el asociado a la variable gastos es 1.26. Siendo exactamente las mismas que con la función lm ya que, como se definió en los argumentos de la función, la función de enlace es Gausiana.

Por último y a modo de resumen se muestra un gráfico donde se encuentran depositados todos los modelos estimados anteriormente y así poder comparar de manera gráfica su ajuste:

df\_model <- data.frame(  
 intercept = c(coefficients(fit)[1],  
 coefficients(fit\_sinP)[1],  
 coefficients(fit\_Ponde)[1],  
 coefficients(fit\_svy)[1]),  
 slope = c(coefficients(fit)[2],  
 coefficients(fit\_sinP)[2],  
 coefficients(fit\_Ponde)[2],  
 coefficients(fit\_svy)[2]),  
 Modelo = c("Población", "Sin ponderar",  
 "Ponderado(lm)", "Ponderado(svyglm)"))  
plot\_BigCity + geom\_abline( data = df\_model,  
 mapping = aes( slope = slope,  
 intercept = intercept, linetype = Modelo,  
 color = Modelo ), size = 2  
 )



### 5.6.2 Uso de ponderaciones

En el análisis de datos provenientes de encuestas con diseños complejos surge un interrogante clave: **¿cómo deben incorporarse los pesos muestrales en los modelos de regresión?** La respuesta no es trivial, ya que los pesos reflejan tanto la probabilidad de selección como los ajustes por no respuesta y calibración, pero su uso directo en el modelado puede generar problemas de eficiencia y aumentar la varianza de los estimadores.

Para enfrentar estas limitaciones, se han propuesto diferentes estrategias de ajuste que buscan lograr un balance entre precisión y eficiencia en la estimación. Entre los procedimientos más utilizados destacan los siguientes:

**Pesos tipo Senado** Este procedimiento ajusta los pesos de manera que su suma coincida con el tamaño de la muestra, en lugar del tamaño de la población. El objetivo es mantener la representatividad relativa de las unidades, pero reduciendo la dispersión de los pesos originales, lo que resulta particularmente ventajoso en encuestas donde existe alta variabilidad entre los factores de expansión:

**Pesos normalizados** En este enfoque los pesos originales se reescalan para que su suma sea igual a uno, lo que evita un incremento innecesario de la varianza en los modelos. Esta técnica es especialmente útil cuando se trabaja con diferentes subconjuntos de datos (por ejemplo, modelos estimados en subpoblaciones) o cuando se busca minimizar la inflación de la varianza:

Es importante resaltar que, en ambos métodos, los pesos ajustados se obtienen mediante transformaciones multiplicativas directas de los pesos muestrales originales. Por esta razón, no deben emplearse para calcular **totales poblacionales**, ni modifican los coeficientes de variación asociados a ellos. Asimismo, estos procedimientos no afectan las estimaciones de **razones**, como medias o proporciones, ya que en dichos casos los pesos se cancelan en el cociente.

En la práctica, el uso de estos ajustes puede considerarse una solución pragmática en contextos donde no se dispone de software especializado para encuestas. Sin embargo, cuando se cuenta con programas estadísticos que permiten la incorporación directa de pesos y del diseño muestral —como los descritos en la Subsección 9.3.4— el reescalamiento de los pesos deja de ser necesario, ya que dichos programas realizan el tratamiento adecuado para preservar tanto la representatividad como las propiedades inferenciales del modelo.

## 5.7 Diagnóstico del modelo

En el análisis de encuestas de hogares, cuando se ajusta un modelo estadístico, es crucial realizar **verificaciones de calidad** que garanticen la validez de las conclusiones. La literatura metodológica destaca que un modelo bien especificado no solo depende de la elección de las covariables, sino también de que se cumplan los supuestos básicos que aseguran la coherencia de los resultados *(Téllez, 2016)*.

Entre los elementos que deben revisarse al aplicar un modelo de regresión lineal en encuestas complejas se encuentran los siguientes:

* **Adecuación del ajuste**: comprobar si el modelo logra explicar una proporción significativa de la variabilidad de la variable de interés y si las predicciones se ajustan razonablemente a los datos observados.
* **Normalidad de los errores**: verificar si los errores se distribuyen aproximadamente de manera normal, lo que garantiza la validez de las pruebas de significancia.
* **Homogeneidad de la varianza (homocedasticidad)**: confirmar que la variabilidad de los errores se mantenga constante a lo largo de los valores de las covariables. La presencia de heterocedasticidad puede sesgar las inferencias.
* **Independencia de los errores**: examinar si los errores son independientes entre sí, evitando correlaciones que comprometan la validez de las pruebas estadísticas.
* **Casos influyentes**: identificar observaciones que ejercen un efecto desproporcionado en la estimación del modelo, lo que podría distorsionar los resultados.
* **Datos atípicos (outliers)**: detectar unidades que se apartan significativamente de la tendencia general de la muestra y que pueden afectar el ajuste del modelo.

En el contexto de encuestas complejas, estas verificaciones adquieren una relevancia particular. Problemas como la multicolinealidad, la falta de independencia entre observaciones o la presencia de valores extremos pueden acentuarse debido al diseño muestral (estratificación, conglomeración y ponderación). Por ello, los procedimientos de diagnóstico no deben limitarse a la revisión de los supuestos clásicos, sino también considerar las especificidades del diseño.

La aplicación sistemática de estas pruebas diagnósticas permite evaluar la solidez del modelo, incrementar la confianza en las inferencias y garantizar que los resultados derivados sean representativos y útiles para el análisis de políticas públicas o para la investigación social aplicada.

### 5.7.1 Coeficiente de determinación

Una medida clásica para evaluar el ajuste de un modelo de regresión es el **coeficiente de determinación** (), también conocido como coeficiente de correlación múltiple. Este indicador estima la proporción de la varianza de la variable dependiente que es explicada por el modelo, y sus valores oscilan entre 0 y 1. Cuanto más próximo esté de 1, mayor será la proporción de variabilidad explicada; por el contrario, un valor cercano a 0 refleja que el modelo aporta poca capacidad explicativa.

No obstante, la interpretación de varía según el campo disciplinar. En ciencias físicas, es común obtener valores superiores al 0.98 o 0.99, mientras que en ciencias químicas suelen alcanzarse niveles por encima de 0.90. En contraste, en ciencias sociales y, en general, en estudios con poblaciones humanas, incluso los mejores modelos explicativos rara vez superan un rango del 20 % al 40 % de la variabilidad de la variable de interés (*Heringa*). Este contraste resalta que la magnitud de no debe interpretarse de manera absoluta, sino en función del contexto y la naturaleza de los datos analizados.

El coeficiente de determinación se calcula a partir de las sumas de cuadrados totales y de error, de la siguiente manera:

donde representa la suma de cuadrados totales y la suma de cuadrados del error.

En encuestas con **diseños de muestreo complejos**, es necesario ajustar esta medida para reflejar la estructura del diseño y los pesos muestrales. En este caso, el estimador ponderado se define como:

donde corresponde a la **suma ponderada de errores al cuadrado**, calculada como:

y representa la **suma total ponderada de cuadrados**, definida por:

Finalmente, dado que tiende a incrementarse a medida que se incluyen más variables en el modelo, se recomienda emplear también el **coeficiente de determinación ajustado** (), que incorpora una corrección en función del número de covariables y del tamaño de la muestra:

donde es el tamaño muestral efectivo y el número de parámetros estimados.

Este ajuste permite una comparación más justa entre modelos con diferente número de predictores y es particularmente útil en el análisis de encuestas, donde la complejidad del diseño y el uso de ponderaciones pueden influir notablemente en la magnitud de .

Para continuar con los modelos ajustados en la sección anterior, se procede a estimar los utilizando R. Inicialmente, se procede a estimar los parámetros del modelo utilizando la función svyglm de survey como se mostró anteriormente y también, se ajusta un modelo solo con el intercepto para obtener la estimación de la SST:

fit\_svy <- svyglm(Income ~ Expenditure,  
 design = diseno)  
  
modNul <- svyglm(Income ~ 1, design = diseno)  
  
s1 <- summary(fit\_svy)  
s0 <-summary(modNul)  
  
WSST<- s0$dispersion  
WSSE<- s1$dispersion

Por tanto, la estimación del es:

R2 = 1- WSSE/WSST  
R2

## variance SE  
## [1,] 0.509 19005

y, para estimar el se requiere definir el diseño muestral pero incluyendo los q-weigthed **(Pffeferman, 2011)**. A continuación, se muestra los pasos para encontrar los q-weigthed:

* Ajustar un modelo de regresión a los pesos finales de la encuesta utilizando las variables predictoras en el modelo de regresión de interés.

fit\_Nul <- lm(wk ~ 1, data = encuesta)

* Obtener las predicciones de los pesos de la encuesta para cada caso como una función de las variables predictoras en el conjunto de datos

qw <- predict(fit\_Nul)

* Dividir los pesos finales de la encuesta por los valores predichos en el paso anterior:

encuesta %<>% mutate(wk1 = wk/qw)

* Usar los nuevos pesos obtenidos para el ajuste de los modelos de regresión:

diseno\_qwgt <- encuesta %>%  
 as\_survey\_design(  
 strata = Stratum,  
 ids = PSU,  
 weights = wk1,  
 nest = T)

Ahora bien, una vez definido el diseño muestral con los nuevos pesos q-weigthed, se procede a calcular el como sigue:

n = sum(diseno\_qwgt$variables$wk)  
p<- 2  
R2Adj = 1-( ( (n-1)/(n-p) )\*R2 )  
R2Adj

## variance SE  
## [1,] 0.491 19005

Como se puede observar, el es un poco más bajo que el y cercanos al 50% que como se comentó anteriormente, dependiendo del contexto del problema se podrá concluir si es grande o pequeño.

Después de realizar la comparación entre las diferentes formas de estimar los coeficientes del modelo se opta por la metodología consolidadas en svyglm:

diseno\_qwgt %<>% mutate(Age2 = Age^2)  
mod\_svy <- svyglm( Income ~ Expenditure + Zone + Sex + Age2 ,  
 design = diseno\_qwgt)  
s1 <- summary(mod\_svy)  
s0 <- summary(modNul)  
  
mod\_svy

Stratified 1 - level Cluster Sampling design (with replacement) With (238) clusters. Called via srvyr Sampling variables: - ids: PSU - strata: Stratum - weights: wk1

Call: svyglm(formula = Income ~ Expenditure + Zone + Sex + Age2, design = diseno\_qwgt)

Coefficients: (Intercept) Expenditure ZoneUrban SexMale Age2  
62.18419 1.22548 63.46000 21.73256 0.00852

Degrees of Freedom: 2604 Total (i.e. Null); 115 Residual Null Deviance: 6.35e+08 Residual Deviance: 3.08e+08 AIC: 38300

stargazer(mod\_svy, header = FALSE,single.row = T,  
 title = "Modelo propuesto",  
 type = "latex", omit.stat=c("bic", "ll"))

### 5.7.2 Diagnóstico de los residuales

En el diagnóstico de los modelos, el análisis de los residuales constituye una herramienta fundamental. Bajo el supuesto de que el modelo ajustado es adecuado, los residuales proporcionan una estimación de los errores y, en consecuencia, permiten evaluar la validez de los supuestos del modelo. Un examen cuidadoso de los mismos ayuda al investigador a determinar si el procedimiento de ajuste ha respetado dichos supuestos o, por el contrario, si alguno de ellos ha sido violado, en cuyo caso sería necesario revisar la especificación del modelo o incluso replantear el método de ajuste.

En encuestas con diseños muestrales complejos, los **residuales de Pearson** son una forma habitual de evaluar discrepancias entre los valores observados y los esperados. Se definen como:

donde representa el valor esperado de bajo el modelo ajustado, es el peso muestral correspondiente al individuo y es la función de varianza del resultado.

#### 5.7.2.1 Residuos estandarizados

En términos generales, los residuos corresponden a la diferencia entre los valores observados y los estimados por el modelo. El análisis de estos residuos es esencial para verificar el cumplimiento de los supuestos de la regresión. Una práctica común consiste en graficar los residuos frente a los valores predichos o frente a las variables independientes. En un modelo correctamente especificado, la nube de puntos resultante debería mostrar un patrón aleatorio; la presencia de formas sistemáticas puede indicar problemas como **heterocedasticidad** (varianza no constante) o **relaciones no lineales** no captadas por el modelo.

El análisis gráfico es un procedimiento ampliamente utilizado para identificar posibles deficiencias en el modelo. En particular, los **gráficos de residuos frente a valores predichos** son una de las herramientas más informativas para evaluar la adecuación del ajuste. La inspección visual de estos gráficos ayuda a determinar si el modelo cumple con los supuestos de normalidad e independencia de los errores.

En el contexto de encuestas complejas, los residuos pueden expresarse de la siguiente manera:

donde es el valor predicho de , es el peso muestral de la unidad y corresponde a la función de varianza asociada. Estos residuos ponderados se emplean para evaluar tanto la **normalidad** como la **homogeneidad de la varianza** en los errores.

#### 5.7.2.2 Evaluación de la homocedasticidad

Uno de los supuestos más relevantes en los modelos de regresión es la constancia de la varianza de los errores (homocedasticidad). Si este supuesto se viola, los estimadores de los parámetros del modelo permanecen insesgados y consistentes, pero pierden eficiencia, es decir, ya no alcanzan la menor varianza posible entre todos los estimadores insesgados.

Para evaluar este aspecto, se recomienda representar los residuos frente a los valores predichos o frente a alguna covariable . La aparición de un patrón sistemático (por ejemplo, forma de embudo o curvaturas) es un indicio de **heterocedasticidad**. En tales casos, pueden considerarse estrategias de corrección, como transformaciones de la variable dependiente, inclusión de términos adicionales en el modelo o el uso de estimadores robustos de varianza.

Otra definición que se debe tener en consideración para el análisis de los residuales es el de la matriz hat, la cual se estima como:

donde,

es una matriz diagonal de y es la función de enlace del modelo lineal generalizado.

### 5.7.3 Observaciones influyentes

En el análisis diagnóstico de modelos, una técnica fundamental consiste en la identificación de observaciones influyentes. Estas son unidades muestrales cuyo impacto sobre el ajuste del modelo es desproporcionado en comparación con el resto de la muestra. Es importante destacar que una observación influyente no necesariamente corresponde a un valor atípico: mientras que un atípico puede estar alejado del patrón general de los datos, su efecto sobre el ajuste puede ser mínimo. Por el contrario, una observación influyente puede alterar de manera significativa las estimaciones de los parámetros, incluso si no luce atípica.

Una observación se considera influyente si su exclusión provoca cambios sustanciales en el ajuste global del modelo o en parámetros específicos. Para detectar este tipo de observaciones es esencial precisar el tipo de influencia que se desea evaluar, ya que una unidad puede ser influyente sobre la estimación de los parámetros pero no sobre la varianza del error, o viceversa.

En el caso de encuestas complejas, este análisis requiere especial atención, pues los pesos muestrales, las estratificaciones y las unidades primarias de muestreo (PSU) amplifican o reducen la influencia de cada observación en comparación con los modelos ajustados bajo supuestos de muestreo simple aleatorio. Para este propósito, la literatura recomienda el uso de herramientas adaptadas a diseños muestrales complejos, tales como el paquete svydiags en R (véase Valliant, 2024), que implementa diagnósticos extendidos compatibles con datos de encuestas.

A continuación, se describen los principales estadísticos utilizados para la detección de observaciones influyentes en modelos de regresión, con sus respectivas adaptaciones al contexto de encuestas complejas:

#### 5.7.3.1 Distancia de Cook

La **distancia de Cook** mide el efecto de eliminar la observación *i* sobre el ajuste global del modelo. Evalúa simultáneamente el tamaño del residual, la varianza estimada y el apalancamiento de la observación. En el contexto de encuestas complejas, su cálculo se adapta incorporando los pesos muestrales:

donde:

* son los pesos de la encuesta,
* es el residual de la observación *i*,
* es el número de parámetros del modelo,
* es el parámetro de dispersión en el modelo lineal generalizado,
* corresponde al apalancamiento de la observación *i*,
* es la varianza linealizada de la ecuación de puntuación.

Para evaluar su magnitud, se compara con puntos de referencia. Una aproximación es el estadístico:

donde son los grados de libertad basados en el diseño. En la práctica, la literatura (Heeringa; Téllez, 2016) suele considerar como observaciones influyentes aquellas cuyo excede valores críticos como 2 o 3.

#### 5.7.3.2

El estadístico cuantifica el cambio en los coeficientes de regresión cuando la observación *i* es eliminada:

donde y es el vector de parámetros estimados sin la observación *i*.

En su forma estandarizada:

La interpretación es directa: una observación es influyente sobre el coeficiente si , con o , o alternativamente si supera el umbral .

#### 5.7.3.3

Finalmente, el estadístico mide la influencia de una observación sobre el ajuste total del modelo. Se calcula como:

La observación *i* se considera influyente si:

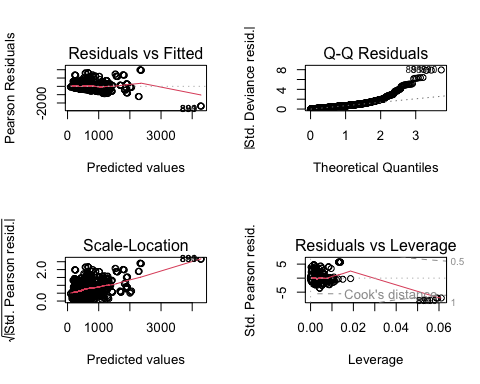
Por otro lado, un análisis que es de vital importancia en el ajuste de modelos de regresión más específicamente en el análisis de residuales es el de varianza constante en los errores. La principal consecuencia de no tener en cuenta la violación de este supuesto es que los estimadores pierden eficiencia. Si el supuesto de varianza constante no se cumple, los estimadores siguen siendo insesgados y consistentes, pero dejan de ser eficientes, es decir, dejan de ser los mejores en cuanto a que ya no tienen la menor varianza entre todos los estimadores insesgados. Como consecuencia de lo anterior, los intervalos de confianza serán más amplios y las pruebas t y F darán resultados imprecisos *(Tellez, 2016)*.

Una de las formas de analizar el supuesto de varianzas constantes en los errores es hacerlo de manera gráfica. Para ello, se grafica los residuos del modelo contra o los residuos del modelo contra . Si al realizar estos gráficos se logra evidenciar un patrón (funciones cuadráticas, cúbicas, logarítmicas, etc), se puede decir que la varianza de los errores no es constante.

Otro supuesto que se debe revisar en los errores al momento de realizar ajustes es la normalidad en lo errores. Una forma muy común para hacer dicha evaluación es realizar un gráfico cuantil-cuantil normal o QQplot. El QQplot es una gráfica de cuantiles para los residuos observados frente a los calculados a partir de una distribución normal teórica que tiene la misma media y varianza que la distribución de los residuos observados. Por lo tanto, una línea recta de 45° en este gráfico sugeriría que la normalidad es una suposición razonable para los errores aleatorios en el modelo.

A manera de ejemplificar los conceptos vistos, se van a utilizar los modelos previamente ajustados. En primero instancia, el análisis del modelo se centrará en los supuestos de normalidad y varianza constante en los errores. Primero, se realizará el análisis de la normalidad en los errores de manera gráfica como se muestra a continuación:

par(mfrow = c(2,2))  
plot(mod\_svy)



Como se puedo observar en el QQplot, hay evidencia gráfica de que los errores no se distribuyen según una distribución normal.

La librería svydiags está pensada en ayudar en el diagnostico de modelos de regresión lineal, siendo una extensión más para complementar el paquete survey. Con las librerías svydiags se extraen los residuales estandarizados como sigue:

library(svydiags)  
stdresids = as.numeric(svystdres(mod\_svy)$stdresids)  
diseno\_qwgt$variables %<>% mutate(stdresids = stdresids)

Podemos hacer el análisis de normalidad también por medio del histograma de los residuales estandarizados como sigue:

ggplot(data = diseno\_qwgt$variables,  
 aes(x = stdresids)) +  
 geom\_histogram(aes(y = ..density..),  
 colour = "black",  
 fill = "blue", alpha = 0.3) +  
 geom\_density(size = 2, colour = "blue") +  
 geom\_function(fun = dnorm, colour = "red",  
 size = 2) +  
 theme\_cepal()+labs(y = "")

y como se puede observar gráficamente los errores no siguen una distribución normal.

Por otro lado, el otro análisis que se realiza de manera gráfica es el de varianzas constantes el cual se realizará a continuación:

Primero, agreguemos las predicciones a la base de datos para poder realizar las gráficas.

library(patchwork)  
diseno\_qwgt$variables %<>%  
 mutate(pred = predict(mod\_svy))  
g2 <- ggplot(data = diseno\_qwgt$variables,  
 aes(x = Expenditure, y = stdresids))+  
 geom\_point() +  
 geom\_hline(yintercept = 0) + theme\_cepal()  
g3 <- ggplot(data = diseno\_qwgt$variables,  
 aes(x = Age2, y = stdresids))+  
 geom\_point() +  
 geom\_hline(yintercept = 0) + theme\_cepal()

g4 <- ggplot(data = diseno\_qwgt$variables,  
 aes(x = Zone, y = stdresids))+  
 geom\_point() +  
 geom\_hline(yintercept = 0) + theme\_cepal()  
g5 <- ggplot(data = diseno\_qwgt$variables,  
 aes(x = Sex, y = stdresids))+  
 geom\_point() + geom\_hline(yintercept = 0) +  
 theme\_cepal()  
  
(g2|g3)/(g4|g5)

Como se puede observar en las gráficas de gastos y edad, ambas muestran tendencias y no un comportamiento aleatorio. Por lo anterior, se puede decir que las varianzas no son constantes.

Otros de os análisis a realizar es revisar si existen datos influyentes en la base de datos. Para ejemplificar los conceptos definidos, se seguirán con los modelos ajustados en la sección anterior. Una vez ajustados estos modelos y verificados los supuestos, se procede a hacer el cálculo de la distancia de Cook’s usando la función svyCooksDdel paquete svydiags como sigue:

library(svydiags)  
d\_cook = data.frame(  
 cook = svyCooksD(mod\_svy),  
 id = 1:length(svyCooksD(mod\_svy)))  
  
table(d\_cook$cook>3)  
  
  
ggplot(d\_cook, aes(y = cook, x = id)) +  
 geom\_point() +  
 theme\_bw(20)

Como se puede observar, ninguna de las distancias de Cook’s es mayor a 3 por lo que, podemos decir que no existen observaciones influyentes.

Ahora bien, se desea observar si hay observaciones influyentes pero utilizando se realiza con la función svydfbetas como se muestra a continuación:

d\_dfbetas = data.frame(t(svydfbetas(mod\_svy)$Dfbetas))  
colnames(d\_dfbetas) <- paste0("Beta\_", 1:5)  
d\_dfbetas %>% slice(1:10L)

| Beta\_1 | Beta\_2 | Beta\_3 | Beta\_4 | Beta\_5 |
| --- | --- | --- | --- | --- |
| 0.0006 | -2e-04 | 0.0021 | -0.0045 | -0.0077 |
| -0.0006 | -1e-04 | 0.0014 | 0.0026 | -0.0031 |
| -0.0009 | -1e-04 | 0.0009 | 0.0022 | 0.0008 |
| -0.0004 | -1e-04 | 0.0012 | -0.0031 | 0.0007 |
| -0.0009 | 0e+00 | 0.0008 | 0.0021 | 0.0014 |
| 0.0009 | 6e-04 | -0.0036 | -0.0063 | 0.0098 |
| 0.0027 | 4e-04 | -0.0031 | -0.0076 | -0.0028 |
| 0.0011 | 3e-04 | -0.0028 | 0.0077 | -0.0043 |
| 0.0030 | 4e-04 | -0.0030 | -0.0078 | -0.0051 |
| -0.0003 | 4e-04 | 0.0012 | -0.0037 | -0.0040 |

Una vez calculado los se procede a acomodar la salida con para verificar cuáles observaciones son influyentes. Para esto, de calcula el umbral (cutoff) para definir si es o no influyente la observación. Ese umbral es tomado de las salidas de la función svydfbetas. Por último, se genera una variable dicotómica que indique si la observación es o no influyente como se muestra a continuación:

d\_dfbetas$id <- 1:nrow(d\_dfbetas)  
d\_dfbetas <- reshape2::melt(d\_dfbetas, id.vars = "id")  
cutoff <- svydfbetas(mod\_svy)$cutoff  
d\_dfbetas %<>% mutate( Criterio = ifelse(abs(value) > cutoff, "Si", "No"))  
  
tex\_label <- d\_dfbetas %>%  
 filter(Criterio == "Si") %>%  
 arrange(desc(abs(value))) %>%  
 slice(1:10L)  
tex\_label

| id | variable | value | Criterio |
| --- | --- | --- | --- |
| 889 | Beta\_1 | 0.2781 | Si |
| 890 | Beta\_2 | -0.2593 | Si |
| 891 | Beta\_1 | 0.2559 | Si |
| 889 | Beta\_2 | -0.2537 | Si |
| 891 | Beta\_2 | -0.2491 | Si |
| 890 | Beta\_1 | 0.2456 | Si |
| 2311 | Beta\_5 | 0.2056 | Si |
| 889 | Beta\_5 | -0.1993 | Si |
| 890 | Beta\_5 | -0.1788 | Si |
| 890 | Beta\_4 | 0.1616 | Si |

Como se pudo observar en la salida anterior hay varias observaciones que resultan influyentes dado el criterio del . A continuación, y de manera ilustrativa, se grafican los y el umbral con el fin de ver de manera gráfica aquellas observaciones influyentes, teniendo en cuenta que, aquellos puntos rojos en la gráfica representan observaciones influyentes.

ggplot(d\_dfbetas, aes(y = abs(value), x = id)) +  
 geom\_point(aes(col = Criterio)) +  
 geom\_text(data = tex\_label,  
 angle = 45,  
 vjust = -1,  
 aes(label = id)) +  
 geom\_hline(aes(yintercept = cutoff)) +  
 facet\_wrap(. ~ variable, nrow = 2) +  
 scale\_color\_manual(  
 values = c("Si" = "red", "No" = "black")) +  
 theme\_cepal()

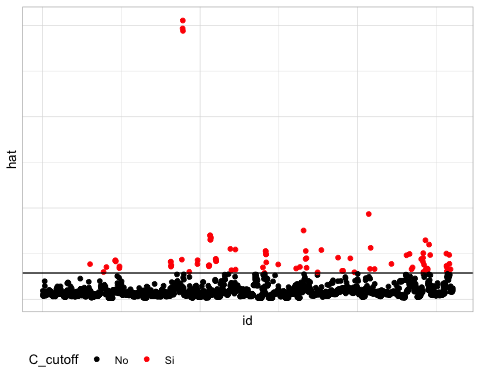
Si el objetivo ahora es detectar observaciones influyentes pero considerando ahora la estadística , se utiliza la función svydffits y se siguen los mismos pasos mostrados para el estadístico :

d\_dffits = data.frame( dffits = svydffits(mod\_svy)$Dffits,  
 id = 1:length(svydffits(mod\_svy)$Dffits))  
  
cutoff <- svydffits(mod\_svy)$cutoff  
  
d\_dffits %<>% mutate(C\_cutoff = ifelse(abs(dffits) > cutoff, "Si", "No"))  
ggplot(d\_dffits, aes(y = abs(dffits), x = id)) +  
 geom\_point(aes(col = C\_cutoff)) +  
 geom\_hline(yintercept = cutoff) +  
 scale\_color\_manual(  
 values = c("Si" = "red", "No" = "black"))+  
 theme\_cepal()

Como se puede observar en el gráfico anterior, también hay observaciones influyentes utilizando , las cuales se muestran en rojo en el gráfico.

Un último acercamiento que se trabajará en este texto para la detección de datos influyentes está encaminado al uso de la matriz *H*. En este sentido, la matriz asociada al Estimador de Pseudo Máxima Verosimilitud (PMLE) de es cuya diagonal esta dado por . Utilizando la matriz *H*, una observación puede ser grande y, como resultado, influir en las predicciones, cuando un es considerablemente diferente del promedio ponderado . Según *(Tellez, 2016)* una observación es considerada grande si es mayor a tres veces el promedio de los . A continuación, se muestra el procedimiento en R cuya función a utilizar es svyhat:

vec\_hat <- svyhat(mod\_svy, doplot = FALSE)  
d\_hat = data.frame(hat = vec\_hat, id = 1:length(vec\_hat))  
d\_hat %<>% mutate(C\_cutoff = ifelse(hat > (3 \* mean(hat)),"Si", "No"))  
  
ggplot(d\_hat, aes(y = hat, x = id)) +  
 geom\_point(aes(col = C\_cutoff)) +  
 geom\_hline(yintercept = (3 \* mean(d\_hat$hat))) +  
 scale\_color\_manual(  
 values = c("Si" = "red", "No" = "black"))+  
 theme\_cepal()



Dado que esta última técnica es empírica, se puede observar en el gráfico anterior que hay varias observaciones posiblemente influyentes en el conjunto de datos de la muestra de hogares.

## 5.8 Inferencia sobre los parámetros del modelo

Una vez evaluado el correcto ajuste del modelo utilizando las metodologías vistas anteriormente y corroboradas las propiedades distribucionales de los errores —y, en consecuencia, de la variable respuesta —, el siguiente paso consiste en verificar si los parámetros estimados son estadísticamente significativos. Esto permite determinar si las covariables incluidas en el modelo aportan de manera sustantiva a la explicación o predicción de la variable de estudio.

En los modelos de regresión, el enfoque clásico para contrastar hipótesis sobre los parámetros se basa en las propiedades distribucionales de sus estimadores. Un estadístico de prueba natural para evaluar la significancia de un coeficiente se construye a partir de la distribución *t-Student* y se define como:

donde es el número de parámetros del modelo y el tamaño muestral. Este estadístico permite contrastar la hipótesis nula frente a la alternativa .

En consecuencia, cuando es lo suficientemente grande, se rechaza la hipótesis nula, concluyendo que la covariable asociada al parámetro tiene un efecto significativo en la variable respuesta.

De igual manera, las propiedades distribucionales de los permiten construir intervalos de confianza al nivel , definidos como:

donde es el error estándar del estimador y corresponde al percentil de la distribución *t-Student* con grados de libertad.

En el caso particular de las encuestas de hogares con **diseños muestrales complejos**, los grados de libertad no se calculan simplemente como (como en el enfoque clásico), sino que deben ajustarse para reflejar la estructura del muestreo. De acuerdo con la teoría de diseño, se establece que:

donde es el número total de conglomerados finales de la primera etapa y el número de estratos de dicha etapa.

Este ajuste es fundamental porque asegura que la inferencia refleje de manera adecuada la variabilidad introducida por la estratificación, la conglomeración y las probabilidades de selección desiguales, evitando conclusiones erróneas sobre la significancia de los parámetros en el análisis de encuestas complejas.

Para la aplicación de las temáticas vistas, es decir, realizar la prueba de hipótesis y los intervalos de confianza para los parámetros utilizaremos el modelo que se ha venido trabajando y aplicaremos las funciones summary.svyglm para las pruebas t y confint.svyglm para los intervalos de confianza como sigue:

survey:::summary.svyglm(mod\_svy)

##   
## Call:  
## svyglm(formula = Income ~ Expenditure + Zone + Sex + Age2, design = diseno\_qwgt)  
##   
## Survey design:  
## Called via srvyr  
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 62.18419 62.98928 0.99 0.33   
## Expenditure 1.22548 0.19798 6.19 9.5e-09 \*\*\*  
## ZoneUrban 63.46000 40.09025 1.58 0.12   
## SexMale 21.73256 15.78081 1.38 0.17   
## Age2 0.00852 0.00557 1.53 0.13   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 118297)  
##   
## Number of Fisher Scoring iterations: 2

survey:::confint.svyglm(mod\_svy)

|  | 2.5 % | 97.5 % |
| --- | --- | --- |
| (Intercept) | -62.5855 | 186.9538 |
| Expenditure | 0.8333 | 1.6176 |
| ZoneUrban | -15.9511 | 142.8711 |
| SexMale | -9.5262 | 52.9913 |
| Age2 | -0.0025 | 0.0196 |

De lo anterior se puede observar que, con una confianza del 95% el único parámetro significativo del modelo es Expenditure y ese mismo resultado lo reflejan los intervalos de confianza.

*Estimación de una observación*

Los modelos de regresión lineales, según *(Neter et al., 1996).*, son utilizado esencialmente con 2 fines, el primero es tratar de explicar la variable respuesta en términos de covariables que pueden encontrarse en la encuesta o en registros administrativos, censos, etc. Adicionalmente, también son usados para predecir valores de la variable en estudio ya sea dentro del intervalo de valores recogidos en la muestra o por fuera de dicho intervalo. Lo primero se ha abordado a lo largo de todo el capítulo y lo segundo se obtiene de la siguiente manera:

De manera explícita, si se ajusta un modelo con 4 covariables la expresión sería:

La varianza de la estimación se calcula de la siguiente manera:

A continuación, se presenta cómo se realiza la estimación del valor esperado, primero se estiman los parámetros del modelo:

| term | estimate | std.error | statistic | p.value |
| --- | --- | --- | --- | --- |
| (Intercept) | 62.1842 | 62.9893 | 0.9872 | 0.3256 |
| Expenditure | 1.2255 | 0.1980 | 6.1901 | 0.0000 |
| ZoneUrban | 63.4600 | 40.0902 | 1.5829 | 0.1162 |
| SexMale | 21.7326 | 15.7808 | 1.3772 | 0.1711 |
| Age2 | 0.0085 | 0.0056 | 1.5297 | 0.1288 |

Por lo anterior, la estimación del valor esperado o predicción queda:

Para calcular la varianza de la estimación, primero se deben obtener las varianzas de la estimación de los parámetros:

vcov(mod\_svy)

|  | (Intercept) | Expenditure | ZoneUrban | SexMale | Age2 |
| --- | --- | --- | --- | --- | --- |
| (Intercept) | 3967.6496 | -11.3101 | 275.9858 | 312.4533 | -0.1852 |
| Expenditure | -11.3101 | 0.0392 | -3.5526 | -0.5004 | 0.0005 |
| ZoneUrban | 275.9858 | -3.5526 | 1607.2280 | -130.7108 | -0.0559 |
| SexMale | 312.4533 | -0.5004 | -130.7108 | 249.0339 | -0.0043 |
| Age2 | -0.1852 | 0.0005 | -0.0559 | -0.0043 | 0.0000 |

Ahora bien, se procede a realizar los cálculos como lo indica la expresión mostrada anteriormente:

xobs <- model.matrix(mod\_svy) %>%  
 data.frame() %>% slice(1) %>% as.matrix()  
  
cov\_beta <- vcov(mod\_svy) %>% as.matrix()  
  
as.numeric(xobs %\*% cov\_beta %\*% t(xobs))

## [1] 1921

Si el objetivo ahora es calcular el intervalo de confianza para la predicción se utiliza la siguiente ecuación:

Para realizar los cálculos en R, se utiliza la función confint y predict como sigue:

pred <- data.frame(predict(mod\_svy, type = "link"))  
pred\_IC <- data.frame(confint(predict(mod\_svy, type = "link")))  
colnames(pred\_IC) <- c("Lim\_Inf", "Lim\_Sup")  
pred\_IC

Ahora, de manera gráfica las predicciones e intervalos se vería de la siguiente manera:

pred <- bind\_cols(pred, pred\_IC)  
pred$Expenditure <- encuesta$Expenditure  
pred %>% slice(1:6L)  
pd <- position\_dodge(width = 0.2)  
ggplot(pred %>% slice(1:100L),  
 aes(x = Expenditure , y = link)) +  
 geom\_errorbar(aes(ymin = Lim\_Inf,  
 ymax = Lim\_Sup),  
 width = .1,  
 linetype = 1) +  
 geom\_point(size = 2, position = pd) +  
 theme\_bw()

Por último, si el interés es hacer una predicción fuera del rango de valores que fue capturado en la muestra. Para esto, supongamos que se desea predecir:

datos\_nuevos <- data.frame(Expenditure = 1600,  
 Age2 = 40^2, Sex = "Male",  
 Zone = "Urban")

La varianza para la predicción se hace siguiendo la siguiente ecuación:

Por tanto, se construye la matriz de observaciones y se calcula la varianza como sigue:

x\_noObs = matrix(c(1,1600,1,1,40^2),nrow = 1)  
as.numeric(sqrt(x\_noObs%\*%cov\_beta%\*%t(x\_noObs)))

## [1] 244.6

Por último, el intervalo de confianza sigue la siguiente ecuación:

En R se hace la predicción de la siguiente manera:

predict(mod\_svy, newdata = datos\_nuevos, type = "link")

## link SE  
## 1 2122 245

y el intervalo:

confint(predict(mod\_svy,newdata = datos\_nuevos))

| 2.5 % | 97.5 % |
| --- | --- |
| 1642 | 2601 |

# 6 Visualización de datos

En esta sección se aborda la manera de presentar datos y estimaciones provenientes de encuestas de hogares mediante representaciones gráficas. Los gráficos, cuando se diseñan de forma adecuada, permiten identificar patrones, tendencias y relaciones en los datos, lo que facilita la interpretación de los hallazgos y su comunicación a públicos diversos.

Si bien es posible utilizar gráficos estándar para mostrar distribuciones y asociaciones a partir de datos muestrales sin ponderar, estos pueden generar interpretaciones erróneas respecto a las distribuciones y asociaciones poblacionales. Por ello, se recomienda emplear gráficos modificados que incorporen los factores de expansión o pesos muestrales.

Por ejemplo, un gráfico de barras que represente la distribución del ingreso debe incorporar los pesos muestrales para reflejar de manera adecuada la distribución estimada de la población total. De igual forma, los diagramas de dispersión que analizan asociaciones entre variables deben utilizar marcadores ponderados o ajustes de densidad con el fin de asegurar representaciones más precisas.

Además, al mostrar estimaciones derivadas de encuestas —que están sujetas a error de muestreo— es esencial reflejar esta condición presentando tanto los estimadores puntuales como los errores estándar o los intervalos de confianza.

Incorporar las características del diseño en las visualizaciones contribuye a que los lectores comprendan la incertidumbre inherente a las estimaciones de encuestas, favoreciendo interpretaciones más fundamentadas. Cuando las unidades de la encuesta poseen diferentes pesos de muestreo, estos deben ser considerados en los gráficos, ya que cada peso refleja la cantidad de unidades poblacionales representadas por la observación muestral.

Si los gráficos se elaboran sin considerar los pesos, la representación visual reflejará la muestra en lugar de la población. Esta discrepancia puede distorsionar las distribuciones, proporciones o relaciones entre variables. Incorporar los pesos garantiza que los gráficos ofrezcan una representación precisa de la población.

## 6.1 Gráficas en R

El objetivo de esta sección es mostrar cómo generar gráficos a partir de encuestas complejas en R. En todo análisis de encuestas, el componente gráfico cumple dos funciones principales:

1. **Análisis exploratorio:** permite revisar tendencias en variables de interés, distribuciones y asociaciones.
2. **Diagnóstico de modelos:** ayuda a verificar supuestos en el ajuste de modelos, tales como la homogeneidad de varianzas o la normalidad de los errores.

Uno de los paquetes más utilizados en R para graficar es ggplot2, desarrollado por Hadley Wickham. Este paquete implementa el enfoque **Grammar of Graphics**, el cual plantea que cualquier gráfico puede describirse mediante una gramática compuesta de capas (datos, estéticas, geometrías, estadísticas, escalas y temas).

En combinación con paquetes como **survey** y **srvyr**, es posible generar gráficos que consideren los pesos de la encuesta y que representen correctamente tanto estimaciones como errores de muestreo.

Como es costumbre, iniciamos cargando las librerías necesarias y configurando opciones generales:

# Configuración general  
# knitr::opts\_chunk$set(cache = TRUE, warning = FALSE, message = FALSE, error = FALSE)  
options(digits = 4)  
  
# Paquetes para manejo de encuestas y visualización  
library(tidyverse)  
library(survey) # Diseños muestrales complejos  
library(srvyr) # Interfaz amigable con dplyr para encuestas  
library(ggplot2) # Gramática de gráficos  
library(patchwork) # Composición de gráficos

El cargue de la base de datos se hace a continuación,

data(BigCity, package = "TeachingSampling")  
encuesta <- readRDS("Data/encuesta.rds")

A continuación, se define el diseño de muestreo:

library(srvyr)  
diseno <- encuesta %>%  
 as\_survey\_design(  
 strata = Stratum,  
 ids = PSU,  
 weights = wk,  
 nest = T  
 )

A partir de las variables de la encuesta, para efectos de los ejemplos, se definen las siguientes variables:

diseno <- diseno %>% mutate(  
 pobreza = ifelse(Poverty != "NotPoor", 1, 0),  
 desempleo = ifelse(Employment == "Unemployed", 1, 0),  
 edad\_18 = case\_when(  
 Age < 18 ~ "< 18 años",  
 TRUE ~ ">= 18 años"  
 )  
)

Como se mostró en capítulos anteriores, se divide la muestra en sub grupos para ejemplificar los conceptos que se mostrarán en este capítulo:

sub\_Urbano <- diseno %>% filter(Zone == "Urban")  
sub\_Rural <- diseno %>% filter(Zone == "Rural")  
sub\_Mujer <- diseno %>% filter(Sex == "Female")  
sub\_Hombre <- diseno %>% filter(Sex == "Male")

Para crear las gráficas en este texto se utilizará por defecto el tema que la CEPAL tiene asignado por defecto. El tema se define a continuación:

theme\_cepal <- function(...) {  
 theme\_light(10) +  
 theme(  
 axis.text.x = element\_blank(),  
 axis.ticks.x = element\_blank(),  
 axis.text.y = element\_blank(),  
 axis.ticks.y = element\_blank(),  
 legend.position = "bottom",  
 legend.justification = "left",  
 legend.direction = "horizontal",  
 plot.title = element\_text(size = 20, hjust = 0.5),  
 ...  
 )  
}

## 6.2 Histogramas para graficar variables continuas

Un histograma es una representación gráfica de la distribución de los datos de una variable numérica continua. Está compuesto por rectángulos (barras) cuya altura es proporcional a la frecuencia de los valores representados y cuyo ancho es proporcional a la amplitud de los intervalos de clase (bins).

Como se mencionó anteriormente, para el análisis de encuestas se recomienda incorporar los factores de expansión o pesos muestrales, de modo que los histogramas reflejen la distribución poblacional estimada y no solo la de la muestra observada.

En esta sección, se construyen histogramas en R utilizando principalmente el paquete ggplot2, apoyados en patchwork para organizar comparaciones visuales.

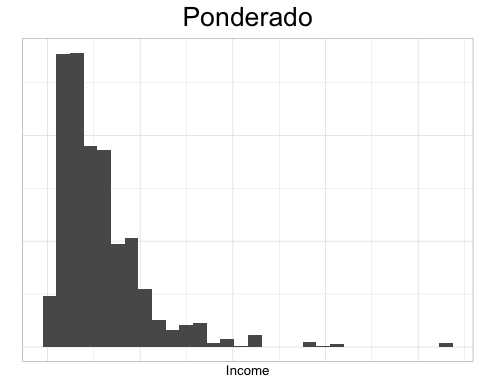
### 6.2.1 Histogramas ponderados y no ponderados

En primera instancia se presenta un histograma de la variable Income, incluyendo los pesos muestrales (wk) de la encuesta. Para ello se definen:

* la fuente de información (data),
* la variable a graficar (x),
* y los pesos de muestreo (weight).

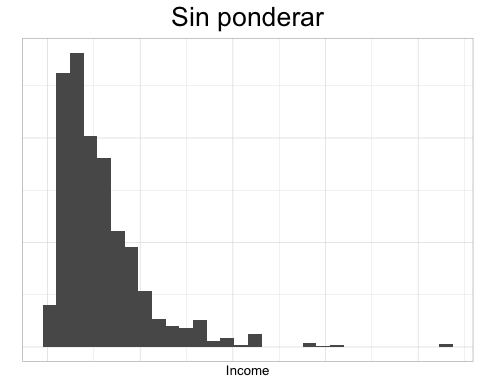
Luego se especifica el tipo de gráfico con geom\_histogram, se ajustan los títulos y se aplica un tema de estilo (en este caso, theme\_cepal).

plot1\_Ponde <- ggplot(  
 data = encuesta,   
 aes(x = Income, weight = wk)) +  
 geom\_histogram(   
 aes(y = ..density..)) +   
 ylab("") +   
 ggtitle("Ponderado") +   
 theme\_cepal()   
plot1\_Ponde



De manera análoga, se construye un histograma **sin aplicar pesos**, es decir, tomando la muestra sin expandir:

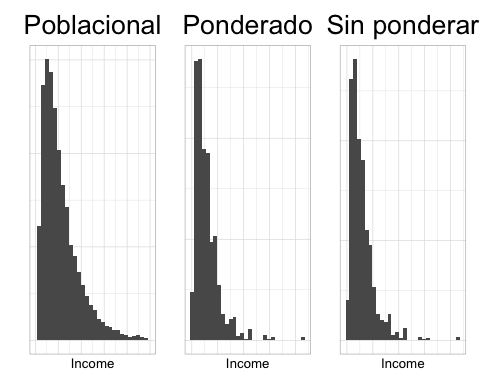
plot1\_SinPonde <-  
 ggplot(encuesta, aes(x = Income)) +  
 geom\_histogram(aes(y = ..density..)) +  
 ylab("") +  
 ggtitle("Sin ponderar") +  
 theme\_cepal()  
plot1\_SinPonde



#### Comparación con la población

Para efectos de contraste, se incluye también el histograma de la población completa (BigCity). Así es posible visualizar las diferencias entre la distribución poblacional, la estimación muestral ponderada y la no ponderada:

plot1\_censo <- ggplot(BigCity, aes(x = Income)) +  
 geom\_histogram(aes(y = ..density..)) +  
 ylab("") +  
 ggtitle("Poblacional") +  
 theme\_cepal() +  
 xlim(0, 2500)  
  
plot1\_censo | plot1\_Ponde | plot1\_SinPonde



Los histogramas muestran cómo se distribuyen los valores de una variable numérica continua en la encuesta. El eje horizontal se divide en intervalos (bins), y cada barra representa la frecuencia de observaciones en dicho rango.

A diferencia de un gráfico de barras, en un histograma **el área de cada barra tiene un significado analítico**: corresponde a la proporción de observaciones en el intervalo, mientras que la altura se obtiene dividiendo esa frecuencia por el ancho del bin.

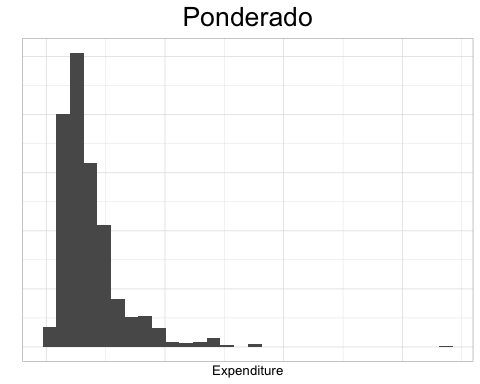
En el caso de encuestas, los histogramas deben construirse incorporando los **pesos de muestreo**. Así, la distribución gráfica refleja la cantidad de unidades poblacionales representadas, en lugar de limitarse a la frecuencia de la muestra. Una aplicación común es graficar la **distribución del ingreso o del gasto**, permitiendo observar su forma, dispersión y tendencia poblacional.

Además, los histogramas pueden emplearse para **comparar subgrupos de la población**, como zonas geográficas (urbana y rural) o características demográficas (sexo, edad, etnia, etc.). Esta estrategia permite detectar diferencias relevantes, por ejemplo, en la forma como se distribuye el ingreso o el gasto en distintos segmentos de la población.

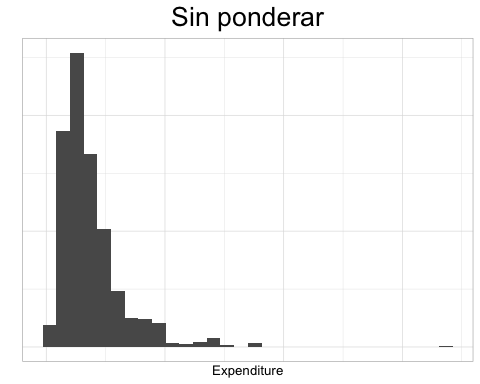
Cuando se combinan histogramas con **curvas de densidad suavizadas**, se obtiene una visión más detallada de las distribuciones poblacionales, lo que contribuye a un análisis más claro y robusto.

Por otro lado, repetimos ahora la secuencia de gráficos pero en este caso para la variable *Expenditure*:

plot2\_Ponde <- ggplot(  
 data = encuesta,  
 aes(x = Expenditure, weight = wk)  
) +  
 geom\_histogram(aes(y = ..density..)) +  
 ylab("") +  
 ggtitle("Ponderado") +  
 theme\_cepal()  
plot2\_Ponde



plot2\_SinPonde <- ggplot(data = encuesta,  
 aes(x = Expenditure)) +  
 geom\_histogram(aes(y = ..density..)) +  
 ylab("") +  
 ggtitle("Sin ponderar") +  
 theme\_cepal()  
plot2\_SinPonde

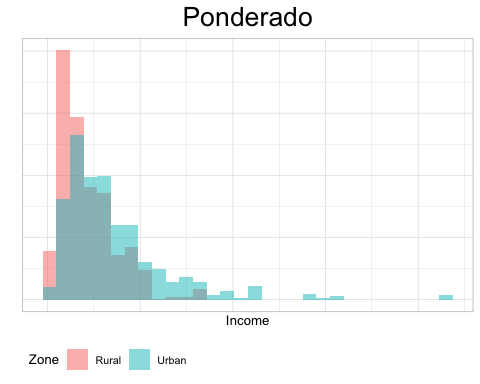


plot2\_censo <- ggplot(BigCity, aes(x = Expenditure)) +  
 geom\_histogram(aes(y = ..density..)) +  
 ylab("") +  
 ggtitle("Poblacional") +  
 theme\_cepal() +  
 xlim(0, 1500)  
  
plot2\_censo | plot2\_Ponde | plot2\_SinPonde

Como conclusión, de ambos ejercicios, se puede observar que el histograma que mejor se aproxima al poblacional es aquel que utiliza los pesos de muestreo, aunque el gráfico que no los utiliza se aproxima bien y esto debido a la correcta selección de la muestra.

Por otro lado, cuando el interés ahora es realizar comparaciones entre dos o más agrupaciones, es posible hacer uso del parámetro fill, el cual “rellena” las barras del histograma con diferentes colores según sea el grupo. Para este ejemplo, se van a graficar subgrupos por zonas:

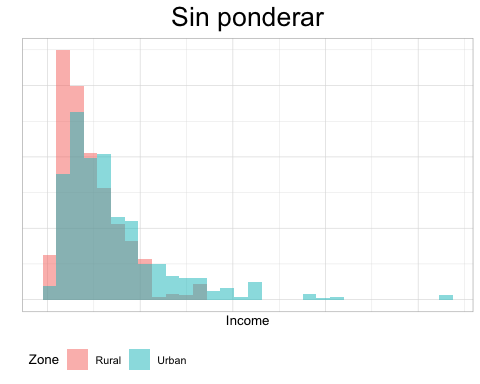
plot3\_Ponde <- ggplot(  
 encuesta,  
 aes(x = Income, weight = wk)) +  
 geom\_histogram(  
 aes(y = ..density.., fill = Zone),  
 alpha = 0.5,  
 position = "identity"   
 ) +  
 ylab("") +  
 ggtitle("Ponderado") +  
 theme\_cepal()  
plot3\_Ponde



Como se pudo observar en la generación del histograma, se utilizó el parámetro position el cual permite que las barras del gráfico sean distingibles.

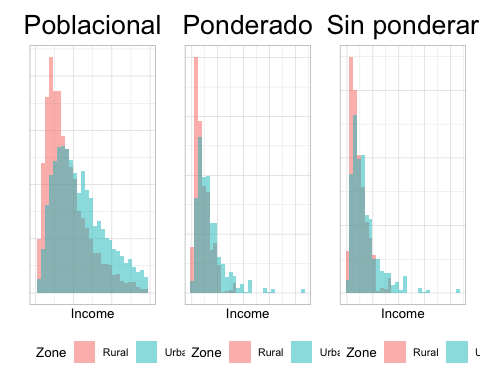
Ahora se graficará la misma variable pero esta vez sin los pesos de muestreo:

plot3\_SinPonde <- ggplot(encuesta, aes(x = Income)) +  
 geom\_histogram(aes(y = ..density.., fill = Zone),  
 alpha = 0.5, position = "identity"  
 ) +  
 ggtitle("Sin ponderar") +  
 theme\_cepal() +  
 ylab("")  
plot3\_SinPonde



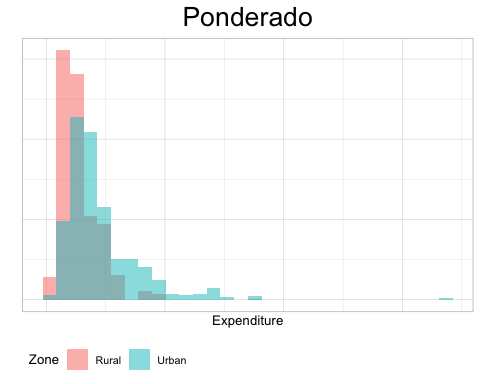
Ahora, siguiendo el esquema de comparación anterior, se graficará la variable ingreso usando la información de la población y los subgrupos de zonas definidos anteriormente y por último, se muestran los 3 histogramas para poder compararlos:

plot3\_censo <- ggplot(BigCity, aes(x = Income)) +  
 geom\_histogram(aes(y = ..density.., fill = Zone),  
 alpha = 0.5, position = "identity"  
 ) +  
 ggtitle("Poblacional") +  
 theme\_cepal() +  
 xlim(0, 1500) +  
 ylab("")  
plot3\_censo | plot3\_Ponde | plot3\_SinPonde

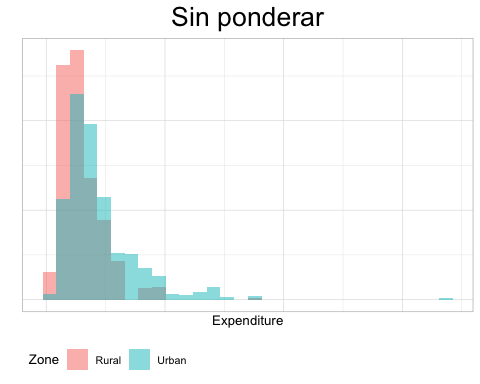


Ahora, repetimos la secuencia de gráficos anteriores pero, para la variable *Expenditure*:

plot4\_Ponde <- ggplot(  
 encuesta,  
 aes(x = Expenditure, weight = wk)  
) +  
 geom\_histogram(aes(y = ..density.., fill = Zone),  
 alpha = 0.5, position = "identity"  
 ) +  
 ylab("") +  
 ggtitle("Ponderado") +  
 theme\_cepal()  
plot4\_Ponde

 Sin ponderar,

plot4\_SinPonde <- ggplot(  
 encuesta,  
 aes(x = Expenditure)  
) +  
 geom\_histogram(aes(y = ..density.., fill = Zone),  
 alpha = 0.5, position = "identity"  
 ) +  
 ggtitle("Sin ponderar") +  
 theme\_cepal() +  
 ylab("")  
plot4\_SinPonde

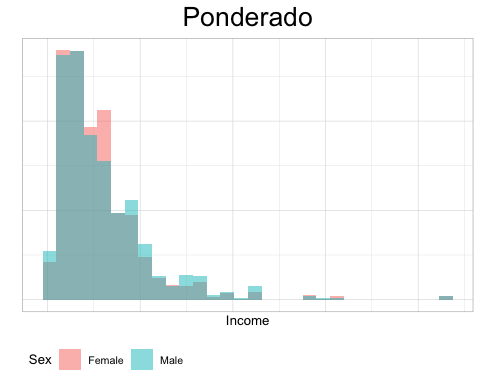


Poblacional,

plot4\_censo <- ggplot(BigCity, aes(x = Expenditure)) +  
 geom\_histogram(aes(y = ..density.., fill = Zone),  
 alpha = 0.5, position = "identity"  
 ) +  
 ggtitle("Poblacional") +  
 theme\_cepal() +  
 xlim(0, 1500) +  
 ylab("")  
plot4\_censo | plot4\_Ponde | plot4\_SinPonde

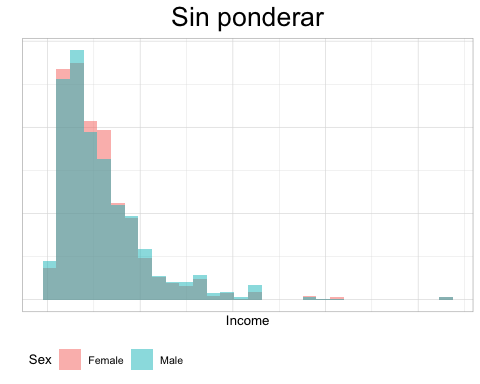
Ahora, repetimos la secuencia de gráficos para la variable *Income*, pero hacemos las particiones por la variable *sexo*, Primero, hagamos el histogramas ponderado:

plot5\_Ponde <- ggplot(  
 encuesta,  
 aes(x = Income, weight = wk)  
) +  
 geom\_histogram(aes(y = ..density.., fill = Sex),  
 alpha = 0.5, position = "identity"  
 ) +  
 ylab("") +  
 ggtitle("Ponderado") +  
 theme\_cepal()  
plot5\_Ponde



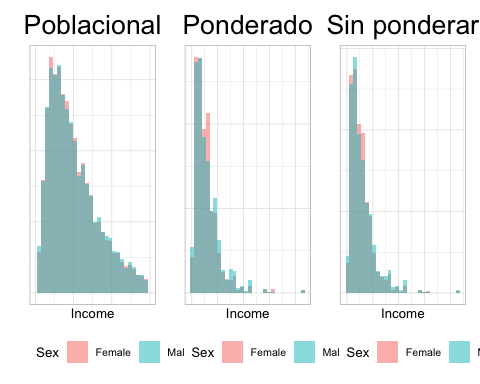
Sin ponderar,

plot5\_SinPonde <- ggplot(encuesta, aes(x = Income)) +  
 geom\_histogram(aes(y = ..density.., fill = Sex),  
 alpha = 0.5, position = "identity"  
 ) +  
 ggtitle("Sin ponderar") +  
 theme\_cepal() +  
 ylab("")  
plot5\_SinPonde



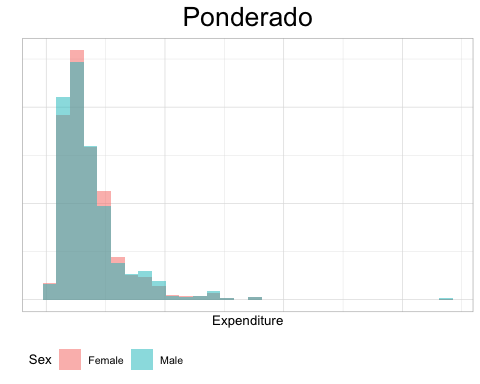
Poblacional,

plot5\_censo <- ggplot(BigCity, aes(x = Income)) +  
 geom\_histogram(aes(y = ..density.., fill = Sex),  
 alpha = 0.5, position = "identity"  
 ) +  
 ggtitle("Poblacional") +  
 theme\_cepal() +  
 xlim(0, 1500) +  
 ylab("")  
plot5\_censo | plot5\_Ponde | plot5\_SinPonde



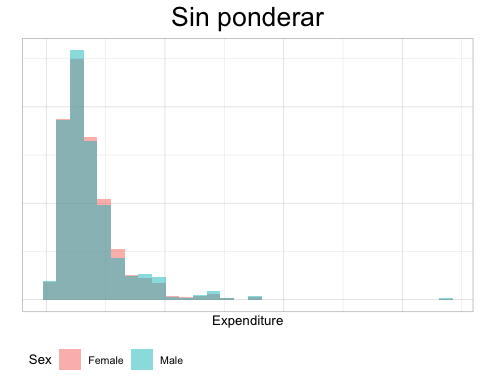
Ahora, repetimos la secuencia de gráficos para la variable *Expenditure* desagregada por la variable *sexo*, primero, ponderado:

plot6\_Ponde <- ggplot(  
 encuesta,  
 aes(x = Expenditure, weight = wk)  
) +  
 geom\_histogram(aes(y = ..density.., fill = Sex),  
 alpha = 0.5, position = "identity"  
 ) +  
 ylab("") +  
 ggtitle("Ponderado") +  
 theme\_cepal()  
plot6\_Ponde



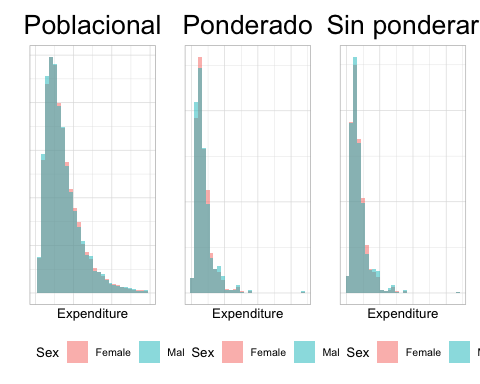
Sin ponderar,

plot6\_SinPonde <- ggplot(encuesta, aes(x = Expenditure)) +  
 geom\_histogram(aes(y = ..density.., fill = Sex),  
 alpha = 0.5, position = "identity"  
 ) +  
 ggtitle("Sin ponderar") +  
 theme\_cepal() +  
 ylab("")  
plot6\_SinPonde



Poblacional,

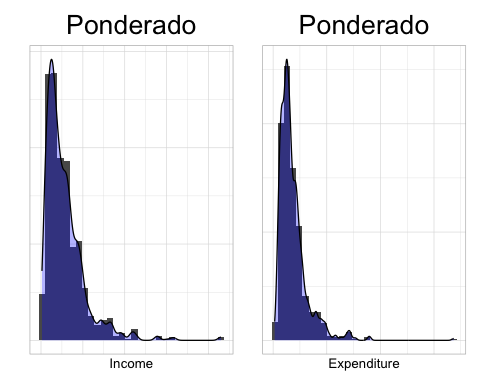
plot6\_censo <- ggplot(BigCity, aes(x = Expenditure)) +  
 geom\_histogram(aes(y = ..density.., fill = Sex),  
 alpha = 0.5, position = "identity"  
 ) +  
 ggtitle("Poblacional") +  
 theme\_cepal() +  
 xlim(0, 1500) +  
 ylab("")  
plot6\_censo | plot6\_Ponde | plot6\_SinPonde



## 6.3 Agregando densidades y graficando Boxplot

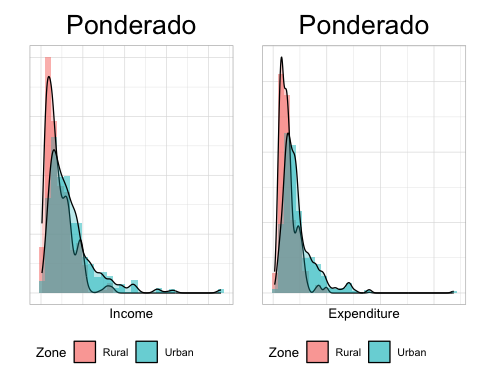
Dadas las cualidades de la librería ggplot2, se pueden agregar nuevas capas a los gráficos, particularmente, a los histogramas antes realizados. La densidad se agrega con el argumento geom\_density y se incorpora el parámetro alpha que regula la transparencia del relleno. A continuacuón, se muestra cómo se agregan las densidades:

plot1\_Ponde + geom\_density(fill = "blue", alpha = 0.3) |  
 plot2\_Ponde + geom\_density(fill = "blue", alpha = 0.3)



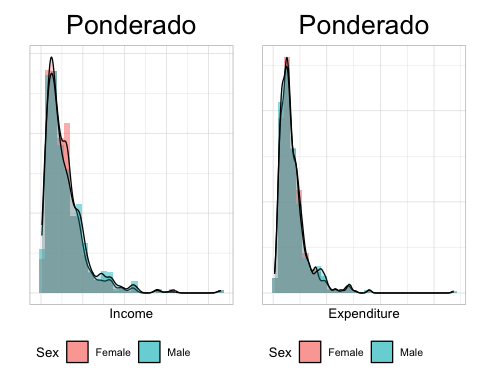
Ahora bien, al aplicar aes(fill = Zone) permite que la densidad sea agregada para cada una de las agrupaciones como se muestra a continución,

plot3\_Ponde + geom\_density(aes(fill = Zone), alpha = 0.3) |  
 plot4\_Ponde + geom\_density(aes(fill = Zone), alpha = 0.3)



En está oportunidad se agrega la desnidad por sexo,

plot5\_Ponde + geom\_density(aes(fill = Sex), alpha = 0.3) |  
 plot6\_Ponde + geom\_density(aes(fill = Sex), alpha = 0.3)



*Boxplot*

El boxplot, diagrama de caja y bigotes, es un gráfico resumen presentado por John Tukey en 1977 que en la actualidad es ampliamente utilizado en la práctica estadística. En este diagrama se visualiza de forma general un conjunto de datos empleando el resumen de cinco números. La forma generada por este gráfico compuesto por un rectángulo (“caja”) y dos brazos (“bigotes”) suministra información sobre la relación ente los cuartiles (Q1, Q2 o mediana y Q3) y los valores mínimo y máximo, sobre la existencia de valores atípicos y la simetría de la distribución.

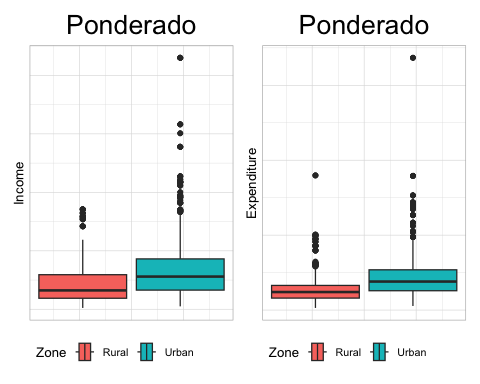
Para realizar este gráfico en ggplot2 se utiliza la función geom\_boxplot. A continuación, se presentan los Boxplot para las variables ingresos y gastos respectivamente:

plot7\_Ponde <- ggplot(  
 data = encuesta, aes(x = Income, weight = wk)  
) +  
 geom\_boxplot() +  
 ggtitle("Ponderado") +  
 coord\_flip() +  
 theme\_cepal()  
  
  
plot8\_Ponde <- ggplot(  
 encuesta,  
 aes(x = Expenditure, weight = wk)  
) +  
 geom\_boxplot() +  
 ggtitle("Ponderado") +  
 coord\_flip() +  
 theme\_cepal()  
  
plot7\_Ponde | plot8\_Ponde

En los gráficos anteriores se puede observar que la variable ingresos tiene más variabilidad que la variable gastos. En ambos gráficos se observan datos atípicos.

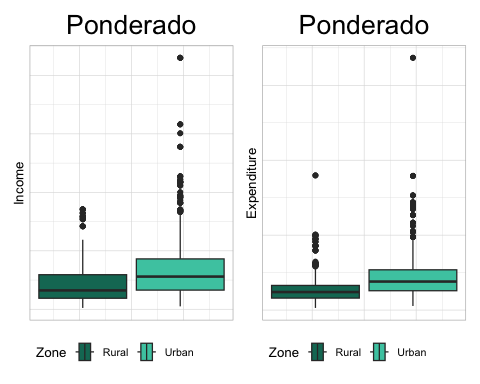
Estos diagramas también permiten la comparación entre dos o más niveles de agrupamiento, por ejemplo, por zonas para las variables ingresos y gastos como se muestra a continuación,

plot9\_Ponde <- ggplot(  
 encuesta,  
 aes(x = Income, weight = wk)  
) +  
 geom\_boxplot(aes(fill = Zone)) +  
 ggtitle("Ponderado") +  
 coord\_flip() +  
 theme\_cepal()  
  
plot10\_Ponde <- ggplot(  
 encuesta,  
 aes(x = Expenditure, weight = wk)  
) +  
 geom\_boxplot(aes(fill = Zone)) +  
 ggtitle("Ponderado") +  
 coord\_flip() +  
 theme\_cepal()  
  
plot9\_Ponde | plot10\_Ponde

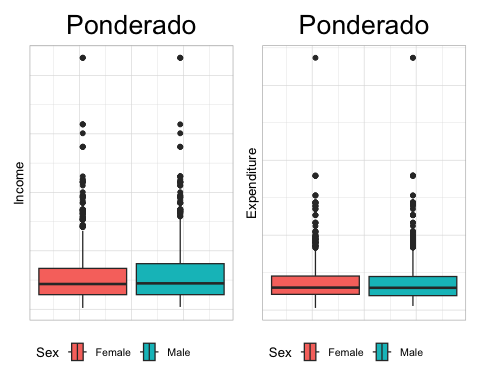
 Observándose, entre otros que, para la variable gasto en la zona rural es donde más datos atípico hay.

Ahora, si se desea personalizar los colores del relleno debe hacer uso de la función scale\_fill\_manualcomo se muestra a continuación:

colorZona <- c(Urban = "#48C9B0", Rural = "#117864")  
plot9\_Ponde + scale\_fill\_manual(values = colorZona) |  
 plot10\_Ponde + scale\_fill\_manual(values = colorZona)

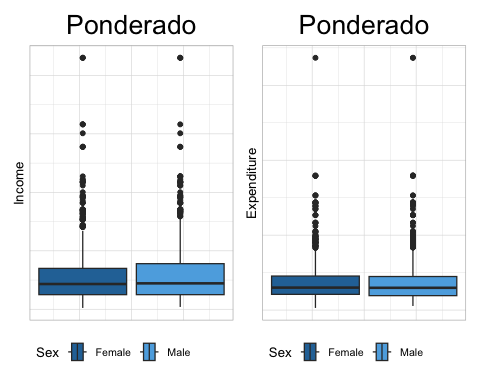
 Para mayores colores, ver la ayuda de la librería. Ahora, si se desea comparar los ingresos y gastos por sexo se procede de la siguiente manera:

plot11\_Ponde <- ggplot(  
 data = encuesta,  
 aes(x = Income, weight = wk)) +  
 geom\_boxplot(aes(fill = Sex)) +  
 ggtitle("Ponderado") +  
 coord\_flip() +  
 theme\_cepal()  
  
plot12\_Ponde <- ggplot(  
 encuesta,  
 aes(x = Expenditure, weight = wk)  
) +  
 geom\_boxplot(aes(fill = Sex)) +  
 ggtitle("Ponderado") +  
 coord\_flip() +  
 theme\_cepal()  
  
plot11\_Ponde | plot12\_Ponde

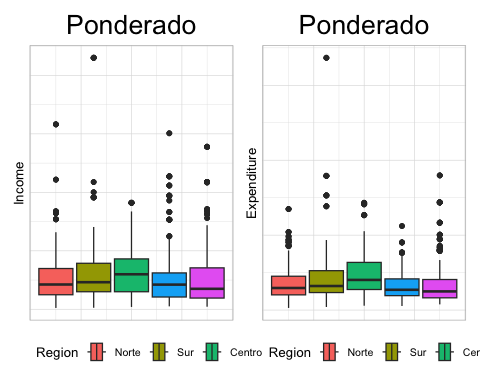


Definiendo el color del relleno para hombres y mujeres:

colorSex <- c(Male = "#5DADE2", Female = "#2874A6")  
plot11\_Ponde + scale\_fill\_manual(values = colorSex) |  
 plot12\_Ponde + scale\_fill\_manual(values = colorSex)

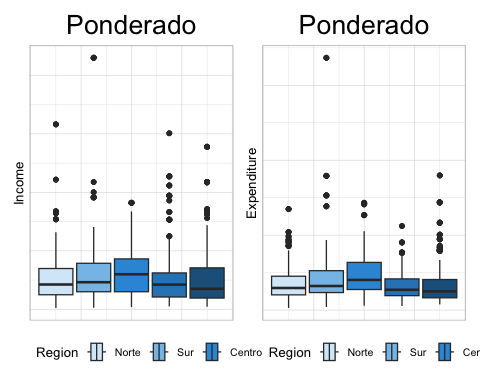
 Realizando la comparación para más de dos categorías, por ejemplo región, se procede como:

plot13\_Ponde <- ggplot(  
 data = encuesta,  
 aes(x = Income, weight = wk)) +  
 geom\_boxplot(aes(fill = Region)) +  
 ggtitle("Ponderado") +  
 coord\_flip() +  
 theme\_cepal()  
  
plot14\_Ponde <- ggplot(  
 data = encuesta,  
 aes(x = Expenditure, weight = wk)) +  
 geom\_boxplot(aes(fill = Region)) +  
 ggtitle("Ponderado") +  
 coord\_flip() +  
 theme\_cepal()  
  
plot13\_Ponde | plot14\_Ponde



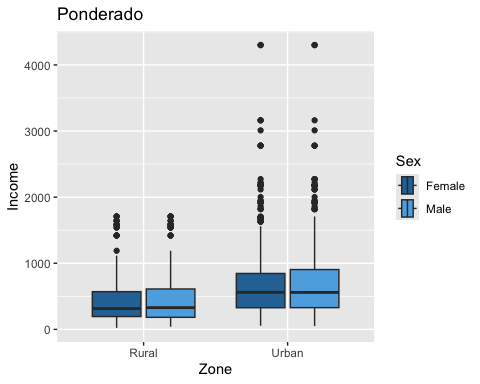
Personalizando los coles cuando hay más de dos categorías, se realiza como se muestra a continuación:

colorRegion <- c(  
 Norte = "#D6EAF8", Sur = "#85C1E9",  
 Centro = "#3498DB", Occidente = "#2E86C1", Oriente = "#21618C"  
)  
plot13\_Ponde + scale\_fill\_manual(values = colorRegion) |  
plot14\_Ponde + scale\_fill\_manual(values = colorRegion)



La función geom\_boxplotpermite realizar comparaciones con más de dos variables al tiempo. A continuación se compara los ingresos por sexo en las diferentes zonas.

plot15\_Ponde <-ggplot(data = encuesta,  
 aes(x = Income, y = Zone, weight = wk)) +  
 geom\_boxplot(aes(fill = Sex)) +  
 ggtitle("Ponderado") +  
 scale\_fill\_manual(values = colorSex) +  
 coord\_flip()  
plot15\_Ponde

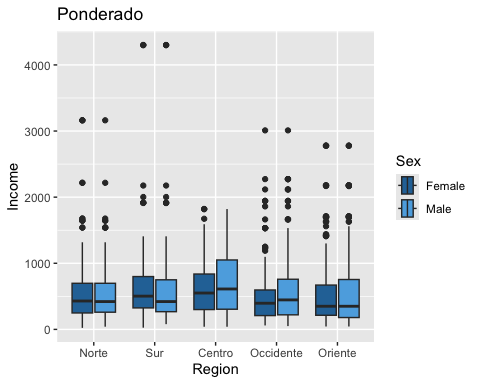


De forma análoga podemos realizar la comparación de los gastos por sexo en las diferentes zonas:

plot16\_Ponde <- ggplot(data = encuesta,  
 aes(x = Expenditure, y = Zone, weight = wk)) +  
 geom\_boxplot(aes(fill = Sex)) +  
 ggtitle("Ponderado") +  
 scale\_fill\_manual(values = colorSex) +  
 coord\_flip()  
plot15\_Ponde / plot16\_Ponde

Se puede extender las comparaciones a variables que tienen más de dos categorías.

plot17\_Ponde <- ggplot(data = encuesta,  
 aes(x = Income, y = Region, weight = wk)) +  
 geom\_boxplot(aes(fill = Sex)) +  
 ggtitle("Ponderado") +  
 scale\_fill\_manual(values = colorSex) +  
 coord\_flip()  
plot17\_Ponde



plot18\_Ponde <- ggplot(data = encuesta,  
 aes(x = Expenditure,  
 y = Region, weight = wk)) +  
 geom\_boxplot(aes(fill = Sex)) +  
 ggtitle("Ponderado") +  
 scale\_fill\_manual(values = colorSex) +  
 coord\_flip()  
  
plot17\_Ponde / plot18\_Ponde

## 6.4 Diagramas de dispersión (Scatterplots)

Los diagramas de dispersión son ideales para explorar relaciones entre dos variables continuas. Permiten detectar patrones, tendencias y posibles asociaciones en los datos. En el contexto de encuestas con diseño muestral complejo, es importante reflejar que las distintas observaciones pueden tener diferentes pesos de muestreo. Para tamaños de muestra pequeños o moderados, esto se puede mostrar mediante puntos de tamaños variables, donde el tamaño del símbolo corresponde al peso de cada observación.

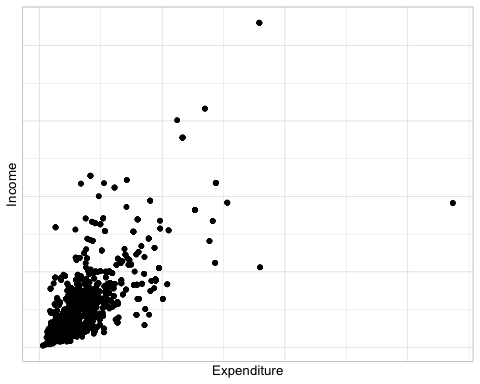
Según Lumley (2010), cuando se trabaja con conjuntos de datos grandes, graficar todos los puntos puede resultar confuso. Existen varias estrategias para manejar este problema:

1. **Submuestreo:** Seleccionar una submuestra aleatoria proporcional a los pesos de muestreo. Esto mantiene la representatividad y facilita la visualización sin perder precisión.
2. **Diagramas de dispersión con hexágonos (hexbin plots):** Dividir el área del gráfico en una cuadrícula de hexágonos y representar cada hexágono con sombreado o tamaño proporcional a la suma de pesos de las observaciones dentro de él. Esta técnica condensa la información de manera clara.
3. **Diagramas de dispersión suavizados:** En lugar de graficar cada punto, se estima una tendencia (por ejemplo, cuantiles de la variable y condicionados a x) y se suaviza a lo largo del eje x. Esto destaca patrones generales y reduce el desorden visual.

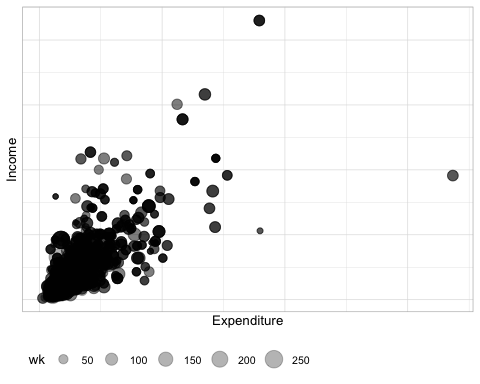
Los diagramas de dispersión ponderados permiten observar relaciones a nivel poblacional. Al incorporar los pesos de la encuesta y aplicar estrategias de visualización para conjuntos de datos grandes, estos gráficos ofrecen información clara y significativa sobre la asociación entre variables continuas. Ya sea usando puntos ponderados, hexbin plots o técnicas de suavizado, los scatterplots siguen siendo fundamentales para la exploración visual de datos en encuestas.

Para realizar este tipo de gráfico se usará la función geom\_point. Para ejemplificar el uso de esta función, se graficarán las variables ingresos y gastos como se muestra a continuación:

plot19\_Ponde <- ggplot(   
 data = encuesta,  
 aes(  
 y = Income,  
 x = Expenditure,  
 weight = wk)) +  
 geom\_point() +  
 theme\_cepal()  
plot19\_Ponde

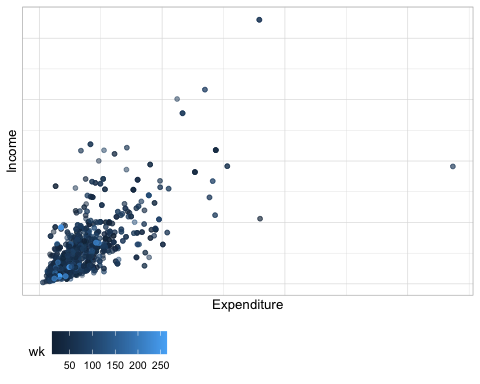
 Note, que en este caso el parámetro weight no está aportando información visual al gráfico. El parámetro weight se puede usar para controlar el tamaño de los puntos, y así, tener un mejor panorama del comportamiento de la muestra:

plot20\_Ponde <- ggplot(  
 data = encuesta,  
 aes(y = Income, x = Expenditure)) +  
 geom\_point(aes(size = wk), alpha = 0.3) +  
 theme\_cepal()  
plot20\_Ponde



Otra forma de usar la variable wk, es asignar la intensidad del color según el valor de la variable:

plot21\_Ponde <- ggplot(  
 data = encuesta,  
 aes(y = Income, x = Expenditure)) +  
 geom\_point(aes(col = wk), alpha = 0.3) +  
 theme\_cepal()  
plot21\_Ponde



Se puede extender las bondades de los gráfico de ggplot2 para obtener mayor información de las muestra. Por ejemplo, agrupar los datos por Zona. Para lograr esto se introduce el parámetro shape:

plot22\_Ponde <- ggplot(  
 data = encuesta,  
 aes(y = Income,   
 x = Expenditure,  
 shape = Zone)) +   
 geom\_point(aes(size = wk, color = Zone), alpha = 0.3) +  
 labs(size = "Peso") + scale\_color\_manual(values = colorZona) +  
 theme\_cepal()  
plot22\_Ponde



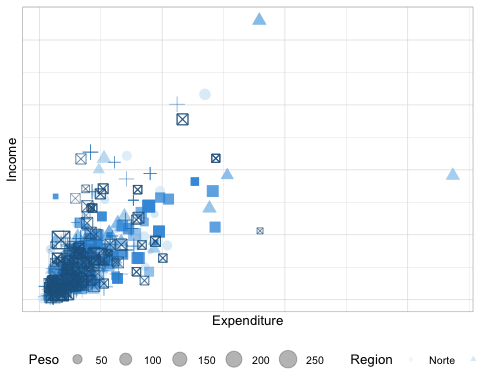
De forma similar se puede obtener el resultado por sexo:

plot23\_Ponde <- ggplot(  
 data = encuesta,  
 aes(  
 y = Income,  
 x = Expenditure,  
 shape = Sex)) +  
 geom\_point(aes(  
 size = wk,  
 color = Sex),  
 alpha = 0.3) +  
 labs(size = "Peso") +  
 scale\_color\_manual(values = colorSex) +  
 theme\_cepal()  
plot23\_Ponde



Un resultado equivalente se obtiene por región:

plot24\_Ponde <- ggplot(  
 data = encuesta,  
 aes(  
 y = Income,  
 x = Expenditure,  
 shape = Region)) +  
 geom\_point(aes(  
 size = wk,  
 color = Region),  
 alpha = 0.3) +  
 labs(size = "Peso") +  
 scale\_color\_manual(values = colorRegion) +  
 theme\_cepal()  
plot24\_Ponde



## 6.5 Diagrama de barras para variables categoricas

Los gráficos de barras son una de las formas más comunes de representar información categórica en el análisis de encuestas. Permiten mostrar la magnitud de las estimaciones asociadas a distintas categorías, ya sea en términos de conteos o proporciones, incorporando los factores de expansión para que los resultados representen de manera adecuada a la población. A diferencia de los gráficos de barras construidos con datos no ponderados, el uso de pesos garantiza que la visualización corresponda a una estimación válida a nivel poblacional.

Es recomendable acompañar las barras con líneas de error que indiquen los intervalos de confianza de las estimaciones puntuales. Esta práctica no solo comunica el valor central de interés, sino que también transmite la incertidumbre asociada al proceso de estimación muestral. De esta forma, los gráficos de barras no solo son útiles para describir patrones generales, sino también para evaluar la precisión y robustez de las conclusiones.

La construcción de gráficos de barras en R requiere en primera instancia calcular las estimaciones que se desean graficar. Para ello, se utilizan funciones como survey\_total() dentro de la librería srvyr, que permiten obtener tanto los estimadores puntuales como sus errores estándar e intervalos de confianza. A continuación, estas estimaciones se representan gráficamente mediante funciones del paquete ggplot2, lo que ofrece gran flexibilidad para definir colores, títulos, escalas y otros elementos visuales.

Una de las ventajas de los gráficos de barras es que pueden extenderse fácilmente a comparaciones entre varias variables categóricas. Por ejemplo, es posible analizar el tamaño poblacional de distintos grupos de pobreza cruzado con la situación de empleo, distinguiendo además subcategorías como desempleo o población infantil. Este tipo de análisis gráfico resulta especialmente útil para identificar intersecciones de vulnerabilidad, al mostrar de manera simultánea dos o más características poblacionales.

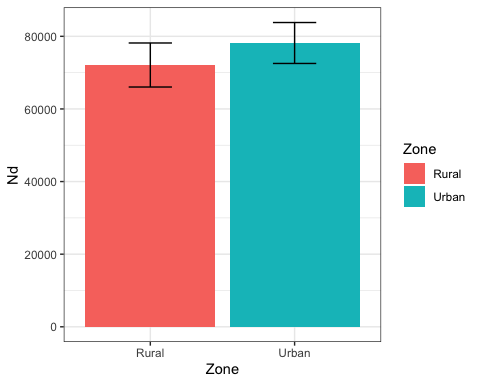
En síntesis, los gráficos de barras permiten no solo una descripción clara de los patrones de consumo, empleo o pobreza en la población, sino también una visualización intuitiva de la incertidumbre estadística. Cuando se aplican correctamente, constituyen una herramienta poderosa para comunicar hallazgos de encuestas de hogares a tomadores de decisiones, investigadores y público en general, manteniendo siempre el rigor estadístico en la representación de los resultados.

tamano\_zona <- diseno %>%  
 group\_by(Zone) %>%  
 summarise( Nd = survey\_total(vartype = c("se", "ci")))  
tamano\_zona

## # A tibble: 2 × 5  
## Zone Nd Nd\_se Nd\_low Nd\_upp  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Rural 72102. 3062. 66039. 78165.  
## 2 Urban 78164. 2847. 72526. 83802.

Ahora, se procede a hacer el gráfico como se mostró en las secciones anteriores:

plot25\_Ponde <- ggplot(  
 data = tamano\_zona,   
 aes(  
 x = Zone,   
 y = Nd,   
 ymax = Nd\_upp,   
 ymin = Nd\_low,   
 fill = Zone)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 geom\_errorbar(position = position\_dodge(width = 0.9),  
 width = 0.3) +  
 theme\_bw()  
plot25\_Ponde



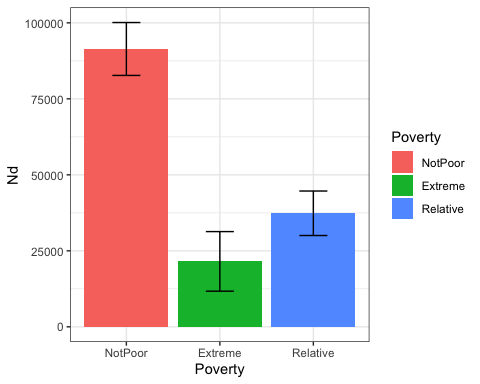
Como se ha visto en los gráficos anteriores, se pueden extender a variables con muchas más categorías:

tamano\_pobreza <- diseno %>%  
 group\_by(Poverty) %>%  
 summarise(Nd = survey\_total(vartype = c("se", "ci")))  
tamano\_pobreza

## # A tibble: 3 × 5  
## Poverty Nd Nd\_se Nd\_low Nd\_upp  
## <fct> <dbl> <dbl> <dbl> <dbl>  
## 1 NotPoor 91398. 4395. 82696. 100101.  
## 2 Extreme 21519. 4949. 11719. 31319.  
## 3 Relative 37349. 3695. 30032. 44666.

El gráfico se obtiene con una sintaxis homologa a la anterior:

plot26\_Ponde <- ggplot(  
 data = tamano\_pobreza,  
 aes(  
 x = Poverty,  
 y = Nd,  
 ymax = Nd\_upp,  
 ymin = Nd\_low,  
 fill = Poverty)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 geom\_errorbar(  
 position = position\_dodge(width = 0.9),  
 width = 0.3) +  
 theme\_bw()  
plot26\_Ponde



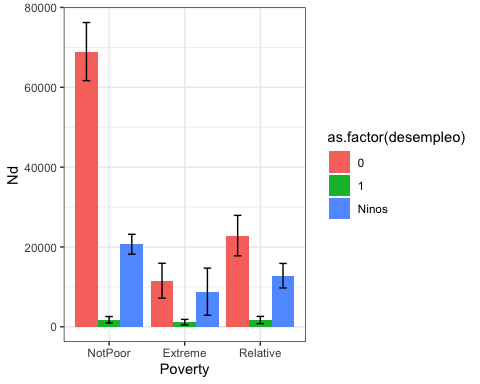
De forma similar a los gráficos Boxplot, es posible realizar comparaciones entre más dos variables.

tamano\_ocupacion\_pobreza <- diseno %>%  
 group\_by(desempleo, Poverty) %>%  
 summarise(Nd = survey\_total(vartype = c("se", "ci"))) %>% as.data.frame() %>%   
 mutate(desempleo = ifelse(is.na(desempleo),"Ninos",desempleo))  
tamano\_ocupacion\_pobreza

## desempleo Poverty Nd Nd\_se Nd\_low Nd\_upp  
## 1 0 NotPoor 68946 3676.3 61666.8 76226  
## 2 0 Extreme 11549 2208.8 7175.8 15923  
## 3 0 Relative 22847 2558.5 17780.5 27913  
## 4 1 NotPoor 1768 405.4 965.7 2571  
## 5 1 Extreme 1169 348.1 479.9 1859  
## 6 1 Relative 1697 457.8 790.7 2604  
## 7 Ninos NotPoor 20684 1256.6 18195.4 23172  
## 8 Ninos Extreme 8800 2979.9 2899.7 14701  
## 9 Ninos Relative 12805 1551.0 9733.9 15876

El gráfico para la tabla anterior queda de la siguiente manera:

plot27\_Ponde <- ggplot(  
 data = tamano\_ocupacion\_pobreza,  
 aes(  
 x = Poverty,  
 y = Nd,  
 ymax = Nd\_upp,  
 ymin = Nd\_low,  
 fill = as.factor(desempleo))) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 geom\_errorbar(  
 position = position\_dodge(width = 0.9),  
 width = 0.3) +  
 theme\_bw()  
plot27\_Ponde



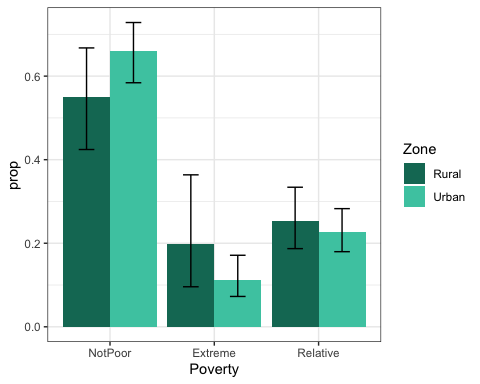
En estos gráficos también se pueden presentar proporciones, como se muestra a continuación:

prop\_ZonaH\_Pobreza <- sub\_Hombre %>%  
 group\_by(Zone, Poverty) %>%  
 summarise(prop = survey\_prop(vartype = c("se", "ci")))  
prop\_ZonaH\_Pobreza

## # A tibble: 6 × 6  
## # Groups: Zone [2]  
## Zone Poverty prop prop\_se prop\_low prop\_upp  
## <chr> <fct> <dbl> <dbl> <dbl> <dbl>  
## 1 Rural NotPoor 0.549 0.0626 0.424 0.668  
## 2 Rural Extreme 0.198 0.0675 0.0958 0.364  
## 3 Rural Relative 0.254 0.0372 0.187 0.334  
## 4 Urban NotPoor 0.660 0.0366 0.584 0.728  
## 5 Urban Extreme 0.113 0.0245 0.0726 0.171  
## 6 Urban Relative 0.227 0.0260 0.180 0.283

Después de tener la tabla con los valores a presentar en el gráfico, los códigos computacionales para realizar el gráfico es el siguiente:

plot28\_Ponde <- ggplot(  
 data = prop\_ZonaH\_Pobreza,  
 aes(  
 x = Poverty, y = prop,  
 ymax = prop\_upp, ymin = prop\_low,  
 fill = Zone)) +   
 geom\_bar(stat = "identity", position = "dodge") +  
 geom\_errorbar(  
 position = position\_dodge(width = 0.9),  
 width = 0.3  
 ) + scale\_fill\_manual(values = colorZona) +  
 theme\_bw()  
plot28\_Ponde



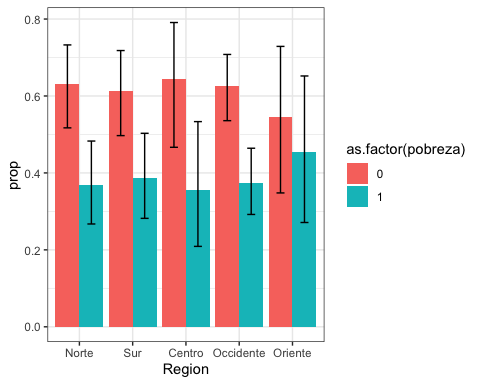
Ahora bien, grafiquemos la proporción de hombres en condición de pobreza por región:

prop\_RegionH\_Pobreza <- sub\_Hombre %>%  
 group\_by(Region, pobreza) %>%  
 summarise(  
 prop = survey\_prop(vartype = c("se", "ci"))) %>%  
 data.frame()  
prop\_RegionH\_Pobreza

## Region pobreza prop prop\_se prop\_low prop\_upp  
## 1 Norte 0 0.6315 0.05523 0.5171 0.7327  
## 2 Norte 1 0.3685 0.05523 0.2673 0.4829  
## 3 Sur 0 0.6134 0.05672 0.4970 0.7181  
## 4 Sur 1 0.3866 0.05672 0.2819 0.5030  
## 5 Centro 0 0.6453 0.08464 0.4666 0.7910  
## 6 Centro 1 0.3547 0.08464 0.2090 0.5334  
## 7 Occidente 0 0.6259 0.04387 0.5358 0.7080  
## 8 Occidente 1 0.3741 0.04387 0.2920 0.4642  
## 9 Oriente 0 0.5450 0.10125 0.3480 0.7289  
## 10 Oriente 1 0.4550 0.10125 0.2711 0.6520

El gráfico de barras es el siguiente:

plot29\_Ponde <- ggplot(  
 data = prop\_RegionH\_Pobreza,  
 aes(  
 x = Region, y = prop,  
 ymax = prop\_upp, ymin = prop\_low,  
 fill = as.factor(pobreza))) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 geom\_errorbar(  
 position = position\_dodge(width = 0.9),  
 width = 0.3  
 ) +  
 theme\_bw()  
plot29\_Ponde



## 6.6 Creando mapas

Los mapas son una herramienta gráfica poderosa para la visualización de datos. Particularmente, para indicadores sociales-demográficos estos son una gran referencia visual para desagregaciones a nivel País, región, departamento, provincia, distrito, municipio, comuna, etc. R posee un sin fin de métodos de programación para representar dichos mapas.

Para graficar mapas es necesario contar con información geoespacial, datos que contienen las coordenadas o delimitaciones geográficas de determinado país o región. Sitios web como <http://www.diva-gis.org/gdata> ofrecen de manera gratuita bases de datos o shapes que contienen los vectores asociados a las geografías correspondientes.

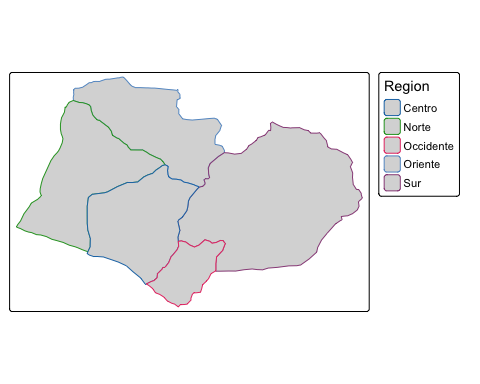
Dichos conjuntos de datos poseen observaciones sobre la longitud y latitud lo cuál permite graficar en R un conjunto de puntos cuya unión en el gráfico formarán las formas los polígonos que dan forma a las áreas geográficas. Entre las distintas librería para realizar mapas en R están tmap y ggplot2. A continuación, se ilustra cómo se generan mapas, inicalmente con la librería tmap:

Inicialmente, para realizar el mapa hay que contar con el archivo de *shepefile* el cual se carga de la siguiente manera::

library(sf)  
library(tmap)  
shapeBigCity <- read\_sf("Data/shapeBigCity/BigCity.shp")

Una vez cargado el shape, el mapa se genera usando las funciones tm\_shape y lo que se desea graficar en el mapa se incluye con la función tm\_polygons. Para este ejemplo, solo grafiquemos las regiones en el mapa:

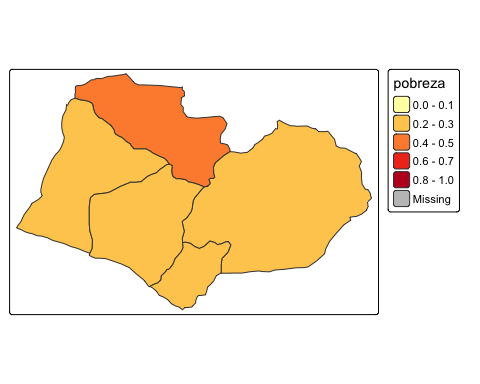
tm\_shape(shapeBigCity) +   
 tm\_polygons(col = "Region")

 Si ahora el objetivo es graficar en las regiones el procentaje de probreza para hombres, inicialmente se debe agregar esa información a la base de datos con la que se graficará el mapa como sigue:

shape\_temp <- tm\_shape(  
 shapeBigCity %>% # shapefile  
 left\_join( # Agregando una variable  
 prop\_RegionH\_Pobreza %>%  
 filter(pobreza == 1), # Filtrando el nivel de interés.   
 by = "Region"))

Una vez generado la base de datos, se procede a crear el mapa. En este ejemplo, agregarán unos puntos de corte en el mapas que son definidos en el argumento brks como se muestra a continuación:

brks <- c(0, .2, .4, .6, 0.8, 1)  
shape\_temp + tm\_polygons(  
 col = "prop",   
 breaks = brks,   
 title = "pobreza",   
 palette = "YlOrRd")

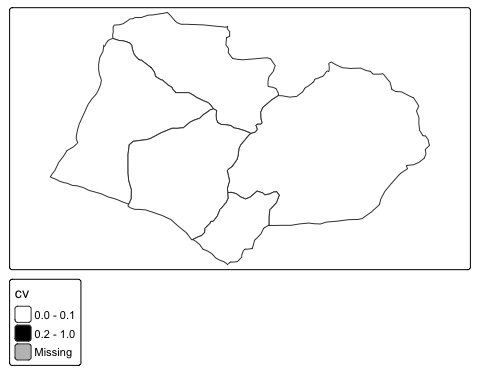


A modo de otro ejemplo, se desea graficar ahora los coeficientes de variación de las estimaciones de los ingresos medios obtenidas por el diseño a nivel de región:

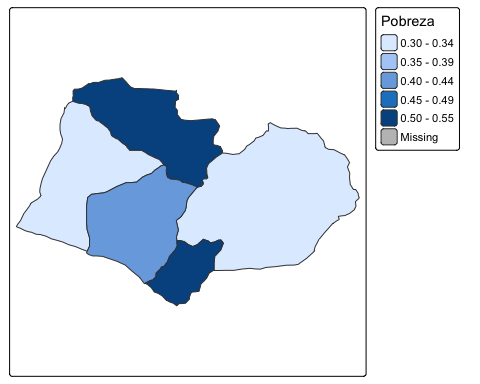
prom\_region <- svyby(~Income, ~Region, diseno,  
 svymean,  
 na.rm = T, covmat = TRUE,  
 vartype = c("cv"))  
prom\_region

## Region Income cv  
## Norte Norte 552.4 0.10022  
## Sur Sur 625.8 0.09973  
## Centro Centro 650.8 0.09445  
## Occidente Occidente 517.0 0.08940  
## Oriente Oriente 541.8 0.13228

brks <- c(0, 0.2, 1)  
shape\_temp <- tm\_shape(  
 shapeBigCity %>%  
 left\_join(  
 prom\_region,  
 by = "Region"))  
  
shape\_temp + tm\_polygons(  
 "cv",  
 breaks = brks,  
 title = "cv",  
 palette = c("#FFFFFF", "#000000"),  
) + tm\_layout(asp = 0)

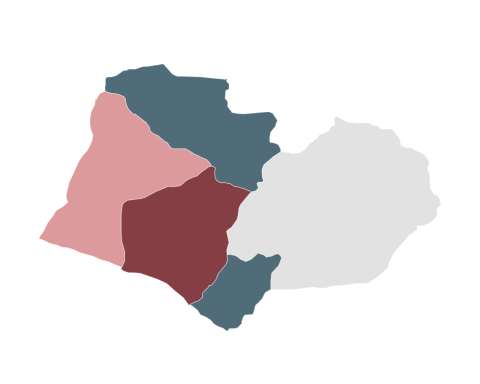
 Ahora, realizar el mismo ejercicio anterior pero por zona y sexo:

prom\_region\_Sex <- diseno %>%  
 group\_by(Region, Zone, Sex, pobreza) %>%  
 summarise(prop = survey\_mean(vartype = "cv")) %>%  
 filter(pobreza == 1, Zone == "Rural", Sex == "Female")  
  
shape\_temp <- tm\_shape(  
 shapeBigCity %>%  
 left\_join(  
 prom\_region\_Sex,  
 by = "Region"))  
  
shape\_temp + tm\_polygons(  
 "prop",  
 title = "Pobreza",  
) + tm\_layout(asp = 0)



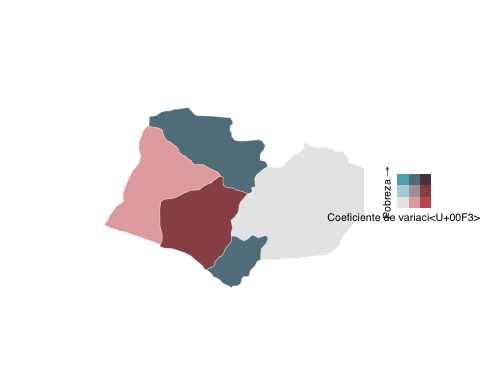
Como se comentó en la introducción de esta sección, los gráficos también se pueden hacer usando la librería ggplot2. Esta librería se apoya en las librerías biscale y cowplot. El procedimiento en R para hacer los mapas es muy similar al mostrado con la librería tmap y se realiza de la siguiente manera:

library(biscale)  
library(cowplot)  
temp\_shape <- shapeBigCity %>%  
 left\_join(  
 prom\_region\_Sex,  
 by = "Region")  
  
k <- 3  
datos.RM.bi <- bi\_class(temp\_shape,  
 y = prop, x = prop\_cv, dim = k,  
 style = "fisher")  
  
map.RM <- ggplot() +  
 geom\_sf(  
 data = datos.RM.bi,  
 aes(fill = bi\_class, geometry = geometry),  
 colour = "white", size = 0.1) +  
 bi\_scale\_fill(pal = "GrPink", dim = k) +  
 bi\_theme() +  
 theme(legend.position = "none")  
map.RM



Ahora, para crear la leyenda del mapa se hace de la siguiente manera:

legend1 <- bi\_legend(  
 pal = "GrPink", dim = k,  
 xlab = "Coeficiente de variaci<U+00F3>n",  
 ylab = "Pobreza", size = 8)  
  
mapa1 <- ggdraw() +  
 draw\_plot(map.RM, 0, 0, 1, scale = 0.7) +  
 draw\_plot(legend1, 0.75, 0.4, 0.2, 0.2, scale = 1) +  
 draw\_text("Estimaciones directas de la pobreza en la mujer rural",  
 vjust = -13, size = 18)  
  
mapa1



# 7 Modelos lineales generalizados en encuestas de hogares

Los modelos lineales generalizados (MLGs) proporcionan una aproximación unificada a la mayoría de los procedimientos usados en estadística aplicada.

El nombre se debe a que ellos generalizan los modelos lineales basados en el supuesto de distribución normal para la variable respuesta. Al igual que los modelos lineales clásicos, tratados en capítulos anteriores, los MLG tienen aplicación en todas las disciplinas del saber. *Nelder & Wedderburn (1972)* presentaron por primera vez el término en un artículo que, sin lugar a dudas, es uno de los más importantes publicados en el área de estadística, por su gran impacto en la forma como se aplica esta disciplina. Desde entonces, poco a poco los modelos lineales generalizados se han ido conociendo y usando ampliamente.

La genialidad de *Nelder & Wedderburn (1972)* consistió en darse cuenta (y demostrar) que muchos de los métodos estadísticos ampliamente usados en la época, aparentemente desligados unos de otros, tales como la regresión lineal múltiple, el análisis probit, el análisis de datos provenientes de ensayos de dilución usando la distribución binomial (realizados por Fisher), los modelos logit para proporciones, los modelos log-lineales para conteos, los modelos de regresión para datos de sobrevivencia, entre otros, se podían tratar con un marco teórico unificado y que las estimaciones de máxima verosimilitud para los parámetros de esos modelos podían obtenerse por el mismo algoritmo conocido como mínimos cuadrados ponderados iterativos (MCPI).

Los desarrollos teóricos en modelos lineales clásicos parten del supuesto que la variable respuesta tiene distribución normal, cuando un fenómeno en estudio genera datos para los cuales no es razonable la suposición de normalidad, como por ejemplo cuando la respuesta es categórica, una proporción o un conteo, obviamente la respuesta no es normal y no es recomendable analizar los datos suponiendo normalidad.

Otro supuesto de los modelos lineales clásicos es el de homogeneidad de la varianza, situación que no se verifica cuando la respuesta es, por ejemplo, una variable aleatoria de poisson, distribución donde la media y la varianza son iguales, es decir, en este modelo un cambio en la media necesariamente implica cambio en la varianza.

Los modelos lineales generalizados son excelentes para modelar datos en condiciones de no normalidad y varianza no constante. Específicamente, se debería considerar usar los MLGs cuando la variable respuesta es: conteos expresados como proporciones, conteos que no son proporciones, respuestas binarias, tiempos de sobrevida donde la varianza se incrementa con la media.

Sin lugar a dudas, en las encuestas de hogares existen variables de tipo conteo, binomiales, etc que meritan su análisis usando modelos lineales generalizados. Es por esto que, este capítulo es de relevancia en este texto.

Para ejemplificar los conceptos, inicialmente se cargan las librerías y la base de datos como sigue:

options(digits = 4)  
options(tinytex.verbose = TRUE)  
library (survey)  
library(srvyr)  
library(convey)  
library(TeachingSampling)  
library(printr)  
library(stargazer)  
library(broom)  
library(jtools)  
library(modelsummary)  
library(patchwork)

Cargue de las bases de datos,

encuesta <- readRDS("Data/encuesta.rds")  
data("BigCity", package = "TeachingSampling")

Por último, se define el diseño muestral,

diseno <- encuesta %>%  
 as\_survey\_design(  
 strata = Stratum,  
 ids = PSU,  
 weights = wk,  
 nest = T  
 )

Se generan nuevas variables en el diseño para ser utilizadas en los ejemplos,

diseno <- diseno %>% mutate(  
 pobreza = ifelse(Poverty != "NotPoor", 1, 0),  
 desempleo = ifelse(Employment == "Unemployed", 1, 0))

Como se ha definido en secciones y capítulos anteriores, con variables dicotómicas se pueden generar tablas de frecuencias teniendo en cuenta los factores de expansión del diseño. En R se hace usando la función svyby de la siguiente manera. Primero, se define la variable a la que se le requiere hacer la tabla (formula), luego se le indica cuál es la variable clasificadora (by). En este caso se quiere hacer una tabla de pobreza claificada por sexo. En tercer lugar se define la función que se quiere aplicar (FUN), en este caso, se quieren calcular totales por celda, por último, se define el diseño de muestreo (design)

tab\_pobreza\_sexo <- svyby(formula = ~factor(pobreza),   
 by = ~Sex,  
 FUN = svytotal,   
 design = as.svrepdesign(diseno),   
 se=F, na.rm=T, ci=T, keep.var=TRUE)  
tab\_pobreza\_sexo

|  | Sex | factor(pobreza)0 | factor(pobreza)1 | se1 | se2 |
| --- | --- | --- | --- | --- | --- |
| Female | Female | 48366 | 30824 | 2411 | 2916 |
| Male | Male | 43032 | 28044 | 2522 | 3095 |

Sin embargo para la estimación de tamaños, se puede emplear también la función svytable como sigue:

tab <- svytable(formula = ~pobreza + Sex, design = diseno)  
kable(tab)

|  | Female | Male |
| --- | --- | --- |
| 0 | 48366 | 43032 |
| 1 | 30824 | 28044 |

Al hacer uso de la función svyby pero usando en el argumento FUN= svymean es posible estimar proporciones como se muestra a continuación:

tab\_pobreza\_sexo <- svyby(formula = ~factor(pobreza),   
 by = ~Sex,  
 FUN = svymean,   
 design = as.svrepdesign(diseno),   
 se=F, na.rm=T, ci=T, keep.var=TRUE)  
tab\_pobreza\_sexo

|  | Sex | factor(pobreza)0 | factor(pobreza)1 | se1 | se2 |
| --- | --- | --- | --- | --- | --- |
| Female | Female | 0.6108 | 0.3892 | 0.0316 | 0.0316 |
| Male | Male | 0.6054 | 0.3946 | 0.0366 | 0.0366 |

También se pueden hacer tablas de doble entrada para la proporción. En forma alternativa es posible usar la función prop.table del paquete base.

kable(prop.table(tab, margin = 2))

|  | Female | Male |
| --- | --- | --- |
| 0 | 0.6108 | 0.6054 |
| 1 | 0.3892 | 0.3946 |

Estas diferentes formas de proceder son de mucha importancia al momento de hacer uso de pruebas de independencia en tablas cruzadas.

## 7.1 Prueba de independencia F

La prueba de independencia F de Fisher permite analizar si dos variables dicotómicas están asociadas cuando la muestra a estudiar es demasiado pequeña y no se cumplen las condiciones para aplicar la prueba . Para utilizar esta técnica, tengamos en cuenta que la probabilidad estimada se escribe como:

Teniendo en cuenta esta expresión, la estadística de Pearson se define de la siguiente manera:

y la estadística de razón de verosimilitud se define como:

donde, es el número de filas y representa el número de columnas, la prueba tiene grados de libertad.

Como lo menciona *Heeringa*, *Fay (1979, 1985)* y *Fellegi (1980)* fueron de los primeros en proponer la corrección del estadístico chi-cuadrado de Pearson basada en un efecto de diseño generalizado (GDEFF, por sus siglas en inglés). *Rao y Scott (1984)* y más tarde *Thomas y Rao (1987)* ampliaron la teoría de las correcciones del efecto de diseño generalizado para estas pruebas estadísticas. El método de Rao-Scott requiere el cálculo de efectos de diseño generalizados que son analíticamente más complicados que el enfoque de Fellegi. Las correcciones de Rao-Scott son ahora el estándar en los procedimientos para el análisis de datos de encuestas categóricas en sistemas de software como Stata y SAS. Los estadísticos de prueba Rao-Scott Pearson ajustados por diseño y razón de verosimilitud chi-cuadrado se calculan de la siguiente manera:

y, para la estadística basada en la razón de verosimilitud se calcula como:

donde el efecto generalizado del diseño () de Rao–Scott, está dado por

Por tanto, la estadística F para independencia basada en la chi-cuadrado de Pearson se calcula como sigue:

y, la estadística F para independencia basada en la razón de verosimilitudes se calcula como sigue:

donde es el número de columnas de la tabla cruzada.

En R, el cálculo de las estadísticas chi-cuadrado y F se camculan usando la función summary como se muestra a continuación:

summary(tab, statistic = "Chisq")

## Sex  
## pobreza Female Male  
## 0 48366 43032  
## 1 30824 28044  
##   
## Pearson's X^2: Rao & Scott adjustment  
##   
## data: NextMethod()  
## X-squared = 0.077, df = 1, p-value = 0.8

Basado en la estadística de Pearson, se puede concluir que el estado de pobreza y el sexo no están relacionados con una confianza del 95%.

summary(tab, statistic = "F")

## Sex  
## pobreza Female Male  
## 0 48366 43032  
## 1 30824 28044  
##   
## Pearson's X^2: Rao & Scott adjustment  
##   
## data: NextMethod()  
## F = 0.056, ndf = 1, ddf = 119, p-value = 0.8

Resultados similares se obtienen con la prueba F de independencia.

## 7.2 Estadístico de Wald

Este estadístico se aplica cuando ya se ha elegido un modelo estadístico ( regresión lineal simple, regresión logística, entre otros).El estadístico de prueba de Wald para la hipótesis nula de independencia de filas y columnas en una tabla de doble entrada se define de la siguiente manera:

donde,

es un vector de de diferencias entre los recuentos de celdas observadas y esperadas, esto es, . La matriz , representa la matriz de varianza y covarianza estimada para el vector de diferencias. En el caso de un diseño de muestra complejo, la matriz de varianza-covarianza de los conteos de frecuencia ponderada, , se estima utilizando métodos de remuestreo o aproximación de Taylor. La matriz es la inversa de la matriz dada por:

Bajo la hipótesis nula de independencia, el estadístico de wald se distribuye chi cuadrado con grados de libertad,

La transformación F del estadístico de Wald es:

En R, para calcular el estadístico de Wald se hace similarmente al cálculo de los estadísticos anteriores usando la función summary como sigue:

summary(tab, statistic = "Wald")

## Sex  
## pobreza Female Male  
## 0 48366 43032  
## 1 30824 28044  
##   
## Design-based Wald test of association  
##   
## data: NextMethod()  
## F = 0.056, ndf = 1, ddf = 119, p-value = 0.8

Se puede concluir que, con una confianza del 95% y basado en la muestra no hay relación entre el estado de pobreza y el sexo.

En este mismo sentido, el estadístco de Wald ajustado en R se se calcula similarmente al anterior y los resultados fueron similares:

summary(tab, statistic = "adjWald")

## Sex  
## pobreza Female Male  
## 0 48366 43032  
## 1 30824 28044  
##   
## Design-based Wald test of association  
##   
## data: NextMethod()  
## F = 0.056, ndf = 1, ddf = 119, p-value = 0.8

## 7.3 Modelo log lineal para tablas de contingencia

El término modelo log-lineal, que básicamente describe el papel de la función de enlace que se utiliza en los modelos lineales generalizados. Iniciaremos esta sección con los modelos log-lineales en tablas de contingencia. El modelo estadístico es el siguiente:

donde:

* la proporción esperada en la celda bajo el modelo.

El modelo log-lineal en R se ajusta utilizando la función svyloglin como sigue:

mod1 <- svyloglin(formula = ~pobreza+Sex + pobreza:Sex,   
 design = diseno)  
s1 <- summary(mod1)  
s1

## Loglinear model: svyloglin(formula = ~pobreza + Sex + pobreza:Sex, design = diseno)  
## coef se p  
## pobreza1 0.219673 0.06778 0.001192  
## Sex1 0.052843 0.01625 0.001145  
## pobreza1:Sex1 0.005583 0.02350 0.812175

En la salida anterior se puede observar que, con una confianza del 95% el estado de pobreza es independiente del sexo, como se ha mostrado con las pruebas anteriores.

Ahora bien, puesto que en la salida anterior se pudo observar que la interacción es no significativa, entonces, ajustemos ahora el modelo sin interacción:

mod2 <- svyloglin(formula = ~pobreza+Sex,   
 design = diseno)  
s2 <- summary(mod2)  
s2

## Loglinear model: svyloglin(formula = ~pobreza + Sex, design = diseno)  
## coef se p  
## pobreza1 0.21997 0.06752 0.0011230  
## Sex1 0.05405 0.01577 0.0006076

Por último, mediante un análisis de varianza es posible comparar los dos modelos como sigue:

anova(mod1, mod2)

## Analysis of Deviance Table  
## Model 1: y ~ pobreza + Sex  
## Model 2: y ~ pobreza + Sex + pobreza:Sex   
## Deviance= 0.07719 p= 0.8827   
## Score= 0.07719 p= 0.8827

De la anterior salida se puede concluir que, con una confianza del 95%, la interacción no es significativa en el modelo log-lineal ajustado.

*Modelo de regresión logistica*

Un modelo de regresion logística es un modelo matemático que puede ser utilizado para describir la relacion entre un conjunto de variables independientes y una variable dicotomica Y. El modelo logístico se describe a continuación:

De aquí,

Por tanto, la probabilidad estimada utilizando el modelo logístico es la siguiente:

La varianza de los parámetros estimados se calcula como sigue:

con,

y,

donde

*Prueba de Wald para los parámetros del modelo*

Para utilizar el estadístico de Wald en en la significancia de los parámetros del modelo se utiliza la razón de verosimilitud. En este caso se contrastan el modelo con todos los parámetros (modelo full) versus el modelo reducido, es decir, el modelo con menos parámetros (modelo reduced),

Dado que el modelo tiene enlace logaritmo, para construir los intervalos de confianza se debe aplicar el función exponencial a cada parámetro,

por ende, el intervalo de confianza es:

A continuación, se muestra el ajuste de un modelo logístico teniendo e cuenta el diseño muestral:

mod\_loglin <- svyglm(formula = pobreza ~ Sex + Zone + Region,  
 family = binomial,   
 design=diseno)  
 tidy(mod\_loglin)

| term | estimate | std.error | statistic | p.value |
| --- | --- | --- | --- | --- |
| (Intercept) | -0.4082 | 0.2640 | -1.5464 | 0.1248 |
| SexMale | 0.0086 | 0.0915 | 0.0945 | 0.9249 |
| ZoneUrban | -0.4378 | 0.2418 | -1.8106 | 0.0729 |
| RegionSur | 0.0063 | 0.3140 | 0.0201 | 0.9840 |
| RegionCentro | 0.1915 | 0.4279 | 0.4476 | 0.6553 |
| RegionOccidente | 0.2319 | 0.3144 | 0.7377 | 0.4622 |
| RegionOriente | 0.3699 | 0.4259 | 0.8686 | 0.3869 |

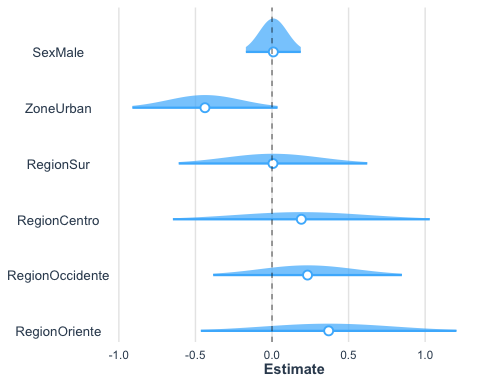
La función *tidy* muestra que ninguna de las covariables son significativas con una confianza del 95%. A continuación, se presentan los intervalos de confianza en los cuales se pueden concluir que en todos los parámetros el cero se encuentra dentro del intrevalo:

confint(mod\_loglin, level = 0.95)

|  | 2.5 % | 97.5 % |
| --- | --- | --- |
| (Intercept) | -0.9312 | 0.1148 |
| SexMale | -0.1727 | 0.1900 |
| ZoneUrban | -0.9169 | 0.0413 |
| RegionSur | -0.6159 | 0.6285 |
| RegionCentro | -0.6562 | 1.0392 |
| RegionOccidente | -0.3910 | 0.8549 |
| RegionOriente | -0.4738 | 1.2136 |

Para verificar de manera gráfica la distribución de los parámetros del modelo, se realizará un gráfico de estos usando la función *plot\_summs* como se muestra a continuación,

library(ggstance)  
plot\_summs(mod\_loglin,   
 scale = TRUE, plot.distributions = TRUE)



Se puede observar en el gráfico que el número cero se encuentra dentro del intervalo de confianza de cada uno de los parámetros, lo que confirma la no significancia al 95% de los parámetros del modelo.

Por otro parte, el estadístico de Wald para el cada una de las variables del modelo se calcula a continuación con la función *regTermTest* para las variables del modelo:

regTermTest(model = mod\_loglin, ~Sex)

## Wald test for Sex  
## in svyglm(formula = pobreza ~ Sex + Zone + Region, design = diseno,   
## family = binomial)  
## F = 0.00893 on 1 and 113 df: p= 0.92

regTermTest(model = mod\_loglin, ~Zone)

## Wald test for Zone  
## in svyglm(formula = pobreza ~ Sex + Zone + Region, design = diseno,   
## family = binomial)  
## F = 3.278 on 1 and 113 df: p= 0.073

regTermTest(model = mod\_loglin, ~Region)

## Wald test for Region  
## in svyglm(formula = pobreza ~ Sex + Zone + Region, design = diseno,   
## family = binomial)  
## F = 0.3654 on 4 and 113 df: p= 0.83

Concluyendo que con una confianza del 95% no son significativas en el modelo como se había mencionado anteriormente.

Como es tradicional en el ejuste de modelos de regresión ya sea, clásico o generalizado, se pueden realizar ajustes con interacciones. A continuación, se present cómo se ajustan modelos loglineales con interacción:

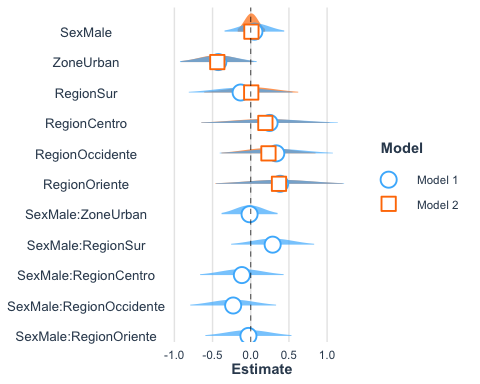
mod\_loglin\_int <- svyglm(formula = pobreza ~ Sex + Zone + Region +   
 Sex:Zone + Sex:Region,  
 family=binomial,   
 design=diseno)  
tab\_mod <- tidy(mod\_loglin\_int) %>% arrange(p.value)  
tab\_mod

| term | estimate | std.error | statistic | p.value |
| --- | --- | --- | --- | --- |
| ZoneUrban | -0.4248 | 0.2562 | -1.6580 | 0.1002 |
| (Intercept) | -0.4289 | 0.2849 | -1.5055 | 0.1351 |
| SexMale:RegionSur | 0.2871 | 0.2774 | 1.0348 | 0.3031 |
| RegionOriente | 0.3843 | 0.4279 | 0.8980 | 0.3712 |
| RegionOccidente | 0.3342 | 0.3783 | 0.8835 | 0.3790 |
| SexMale:RegionOccidente | -0.2302 | 0.2868 | -0.8026 | 0.4240 |
| RegionCentro | 0.2466 | 0.4560 | 0.5408 | 0.5897 |
| SexMale:RegionCentro | -0.1162 | 0.2791 | -0.4162 | 0.6781 |
| RegionSur | -0.1325 | 0.3464 | -0.3825 | 0.7028 |
| SexMale | 0.0478 | 0.1994 | 0.2399 | 0.8109 |
| SexMale:RegionOriente | -0.0304 | 0.2878 | -0.1057 | 0.9161 |
| SexMale:ZoneUrban | -0.0154 | 0.1872 | -0.0824 | 0.9345 |

Observando que la interacción tampoco es significativa con una confianza del 95%.

El gráfico de la distribución de los parámetros del modelo con intercepto y sin intercepto se presenta a continuación:

plot\_summs(mod\_loglin\_int, mod\_loglin, scale = TRUE, plot.distributions = TRUE)



El estadístico de Wald sobre los parámetros del modelo con intercepto son:

regTermTest(model = mod\_loglin\_int, ~Sex)

## Wald test for Sex  
## in svyglm(formula = pobreza ~ Sex + Zone + Region + Sex:Zone + Sex:Region,   
## design = diseno, family = binomial)  
## F = 0.05753 on 1 and 108 df: p= 0.81

regTermTest(model = mod\_loglin\_int, ~Zone)

## Wald test for Zone  
## in svyglm(formula = pobreza ~ Sex + Zone + Region + Sex:Zone + Sex:Region,   
## design = diseno, family = binomial)  
## F = 2.749 on 1 and 108 df: p= 0.1

regTermTest(model = mod\_loglin\_int, ~Region)

## Wald test for Region  
## in svyglm(formula = pobreza ~ Sex + Zone + Region + Sex:Zone + Sex:Region,   
## design = diseno, family = binomial)  
## F = 0.8999 on 4 and 108 df: p= 0.47

regTermTest(model = mod\_loglin\_int, ~Sex:Zone)

## Wald test for Sex:Zone  
## in svyglm(formula = pobreza ~ Sex + Zone + Region + Sex:Zone + Sex:Region,   
## design = diseno, family = binomial)  
## F = 0.006789 on 1 and 108 df: p= 0.93

regTermTest(model = mod\_loglin\_int, ~Sex:Region)

## Wald test for Sex:Region  
## in svyglm(formula = pobreza ~ Sex + Zone + Region + Sex:Zone + Sex:Region,   
## design = diseno, family = binomial)  
## F = 1.058 on 4 and 108 df: p= 0.38

Observándose que con una confianza del 95% ninguno de los parámetros del modelo es significativo.

Ahora bien, como se ha explicado a los largo de los capítulos relacionado con modelos, se pueden ajustar modelos usando *Q\_Weighting*. A continuación, se presenta cómo se ajusta el modelo usando estos pesos:

fit\_wgt <- lm(wk ~ Sex + Zone + Region ,  
 data = encuesta)  
 wgt\_hat <- predict(fit\_wgt)  
 encuesta %<>% mutate(wk2 = wk/wgt\_hat)  
   
 diseno\_qwgt <- encuesta %>%  
 as\_survey\_design(  
 strata = Stratum,  
 ids = PSU,  
 weights = wk2,  
 nest = T  
 )

Defiendo la variable pobreza dentro de la base de datos.

diseno\_qwgt <- diseno\_qwgt %>% mutate(  
pobreza = ifelse(Poverty != "NotPoor", 1, 0))

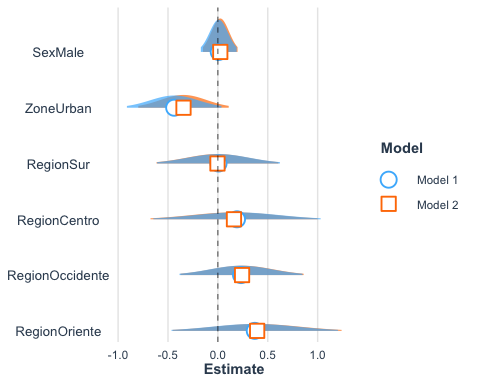
Ajustando el modelo se tiene:

mod\_loglin\_qwgt <- svyglm(formula = pobreza ~ Sex + Zone + Region,  
 family=quasibinomial,  
 design=diseno\_qwgt)  
tab\_mod <- tidy(mod\_loglin\_qwgt)  
tab\_mod

| term | estimate | std.error | statistic | p.value |
| --- | --- | --- | --- | --- |
| (Intercept) | -0.4644 | 0.2630 | -1.7656 | 0.0802 |
| SexMale | 0.0241 | 0.0883 | 0.2726 | 0.7857 |
| ZoneUrban | -0.3445 | 0.2311 | -1.4903 | 0.1389 |
| RegionSur | -0.0041 | 0.3116 | -0.0130 | 0.9896 |
| RegionCentro | 0.1613 | 0.4270 | 0.3778 | 0.7063 |
| RegionOccidente | 0.2424 | 0.3147 | 0.7705 | 0.4426 |
| RegionOriente | 0.3937 | 0.4319 | 0.9115 | 0.3639 |

Concluyendo que, con una confianza del 95% y basado en la muestra, ninguno de los parámetros del modelo es significativo. Lo cual se puede corroborar con el gráfico de la distribución de los parámetros del modelo con los pesos ajustados y sin los pesos:

plot\_summs(mod\_loglin, mod\_loglin\_qwgt,   
 scale = TRUE, plot.distributions = TRUE)



El Estadístico de Wald sobre los parámetros del modelo se obtiene de manera similar a lo visto anteriormente:

regTermTest(model = mod\_loglin\_qwgt, ~Sex)

## Wald test for Sex  
## in svyglm(formula = pobreza ~ Sex + Zone + Region, design = diseno\_qwgt,   
## family = quasibinomial)  
## F = 0.0743 on 1 and 113 df: p= 0.79

regTermTest(model = mod\_loglin\_qwgt, ~Zone)

## Wald test for Zone  
## in svyglm(formula = pobreza ~ Sex + Zone + Region, design = diseno\_qwgt,   
## family = quasibinomial)  
## F = 2.221 on 1 and 113 df: p= 0.14

regTermTest(model = mod\_loglin\_qwgt, ~Region)

## Wald test for Region  
## in svyglm(formula = pobreza ~ Sex + Zone + Region, design = diseno\_qwgt,   
## family = quasibinomial)  
## F = 0.4156 on 4 and 113 df: p= 0.8

Concluyendo también que ninguna de las variables son significativas.

# 8 Modelos lineales generalizados: Introducción a la pseudo máxima verosimilitud

En la introducción de su excelente libro, *Statistical Design for Researches*, Leslie Kish afirma que el enunciado de la mayoría de libros de inferencia estadística abren con el siguiente enunciado: *Dadas variables aleatorias, seleccionadas de una población, independientes e idénticamente distribuidas* y que cada palabra en el anterior enunciado es engañosa. ¿Quién le da a uno las muestras? ¿Existe algún sitio en dónde las repartan? Las muestras no son dadas, las muestras deben ser seleccionadas, asignadas o capturadas. El tamaño de la muestra no siempre es un número fijo, en la mayoría de casos prácticos es una variable aleatoria. Los datos no siguen el supuesto de independencia ni de idéntica distribución; es más, en muchas ocasiones no existe una sola población, sino que la muestra seleccionada es el resultado de una selección de sub-poblaciones para las cuales se deben producir, no solo una sino muchas estimaciones.

En la teoría de muestreo, se considera que las características de interés son parámetros y no constituyen realizaciones de variables aleatorias. Para reforzar esta idea haga lo siguiente: examine una moneda y obsérvela. Suponga que usted está observando la cara (o sello, da igual) de la moneda. Esa cara (o sello) no constituye una realización de una variable aleatoria. Para que se pueda hablar de una variable aleatoria, es necesario realizar un experimento, el cual induce el conjunto de todos los posibles resultados, el cual a su vez induce una sigma-álgebra que define a la variable aleatoria. Sería muy diferente si se crease un experimento con esa moneda. El más sencillo de todos sería lanzarla al aire y observar si la moneda cayó en cara o sello. De forma similar, es muy válido afirmar que, por ejemplo, el estado de la naturaleza de un individuo que está desempleado no constituye una realización de una variable aleatoria.

Un ejemplo práctico se presenta a la hora de estimar la tasa de desempleo, se considera que, si un individuo está desempleado, pues está desempleado y punto. En otras palabras, el estado de la naturaleza del individuo al momento de la medición es “desempleado” y esta caracterización no corresponde a ninguna realización de algún evento aleatorio. Es por esto que, una vertiente de la inferencia en poblaciones finitas considera que el parámetro de interés será el número total de personas desempleadas dividido por el número total de personas en la fuerza laboral. Si se tuviese la oportunidad de medir a todos los integrantes de la fuerza laboral, mediante la realización de un censo, pues esa división correspondería al parámetro poblacional con el cual se tomarían decisiones y/o se cambiarían o reforzarían las políticas públicas de un país.

El propósito de este capítulo es llevar a los lectores al correcto análisis de sus datos, preguntándose acerca del proceso de selección de la muestra. Más aún, en términos de muestreo, solo hay un único caso para el cual la teoría de la inferencia estadística es aplicable y se trata del muestreo aleatorio simple con reemplazo en donde si se tienen las propiedades de independencia y de idéntica distribución. Note que, en términos de selección de muestras, solo hay dos posibles escenarios generales. La selección con reemplazo y la selección sin reemplazo.

## 8.1 Acerca de las muestras aleatorias y su análisis

Hablemos primero de la selección sin reemplazo, en donde una muestra seleccionada está conformada por algunos elementos de la población que no se repiten. Para seleccionar una muestra sin reemplazo de tamaño , de una población de tamaño , el proceso de selección puede ser de la siguiente manera. Se escoge una unidad de las cinco posibles, luego se selecciona una unidad de las cuatro restantes, y por último, una unidad de las tres restantes. Esto hace que el proceso de selección de la muestra no se lleve a cabo de forma independiente. Por ejemplo, si el muestreo es aleatorio simple, la probabilidad de selección de la primera unidad es 1/5, la probabilidad de selección de la segunda unidad es 1/4 y así sucesivamente. Por otro lado, cuando el muestreo es con reemplazo, la selección se realiza de forma independiente puesto que se trata de realizar el mismo ensayo (seleccionar una unidad de cinco posibles) tres veces, sin importar que las unidades tengan diferentes probabilidades de selección.

Por otra parte, es bien sabido que la teoría de muestreo establece que el valor de la característica de interés, , es eso, un valor; por tanto, no es aleatorio. Luego, es incorrecto decir que es una variable aleatoria asociada con alguna distribución de probabilidad. Recuerde que en el muestreo lo único aleatorio en la inferencia es la muestra. Ahora, no significa que no podamos construir variables aleatorias en muestreo. Por ejemplo, construyamos la siguiente variable aleatoria () definida como el valor de la característica de interés en el individuo -ésimo, seleccionado en la -ésima extracción. En este caso, existen tres variables aleatorias, puesto que la muestra es de tamaño tres.

Si consideramos un muestreo aleatorio sin reemplazo, la primera variable aleatoria , podrá tomar cualquiera de los siguiente cinco valores: . La segunda variable aleatoria , solo podrá tomar cuatro valores, puesto que ya fue realizada, y la tercera variable aleatoria solo podrá tomar tres valores, puesto que y ya fueron realizadas. Esto hace que , y no constituya una sucesión de variables aleatorias independientes (puesto que la selección sin reemplazo no es un proceso independiente) ni idénticamente distribuidas (puesto que ni siquiera su espacio muestral es el mismo: puede tomar cinco valores, solo cuatro y solo tres). Lo cual quiere decir que a partir de un muestreo sin reemplazo (ni siquiera el tan mencionado muestreo aleatorio simple) no es posible construir una muestra aleatoria, como las que aparecen en los libros de teoría estadística.

Sin embargo, algo muy distinto sucede con el muestreo con reemplazo. Cuando construimos las variables aleatorias , y , resulta ser que ellas sí conforman una sucesión de variables aleatorias independientes (puesto que el muestreo con reemplazo sí define un proceso de extracciones independientes) e idénticamente distribuidas (puesto que conservan el mismo espacio muestral y mantienen la probabilidad de selección). Es decir, puede tomar los valores . La probabilidad de que es , la probabilidad de selección del primer elemento; la probabilidad de que es , la probabilidad de selección del segundo elemento y así sucesivamente hasta obtener que la probabilidad de que es , la probabilidad de selección del primer elemento primer elemento. La misma distribución la tienen y . Por lo tanto, , y conforman una muestra aleatoria, como las que aparecen en los libros clásicos de inferencia estadística.

Entonces, hemos llegado a un punto sin retorno, en donde la conclusión es que, si la muestra fue seleccionada con reemplazo, entonces podemos inducir una muestra aleatoria. Sin embargo, existen muchas variantes en el muestreo con reemplazo. A continuación, vamos a dilucidar cuál de ellas es la indicada para analizar la muestra de acuerdo con la teoría de los libros de inferencia.

En primera instancia, veamos que para que la esperanza (bajo el diseño de muestreo ) de cualquier variable aleatoria sea igual a la media poblacional, es necesario que, para todos los individuos en la población, la probabilidad de selección sea idéntica e igual a , como se muestra a continuación:

De la misma manera, para que la varianza de cualquier variable aleatoria sea igual a la varianza poblacional, se requiere la misma condición, puesto que:

Por lo tanto, la esperanza y la varianza de un estimador clásico como solo coincidierón con los bien conocidos resultados de la inferencia clásica cuando el muestreo haya sido aleatorio simple con reemplazo. De otra forma, no se tienen las, bien conocidas, propiedades de esta estadística que implican que su esperanza es y su varianza es .

Este razonamiento de aplicarse de la misma forma para pruebas de hipótesis, construcción de intervalos de confianza, modelos de regresión, y hasta diseño de experimentos. Ahora, para una encuesta cuyos datos no fueron extraídos de manera aleatoria simple con reemplazo, la manera correcta de analizarla confiadamente es incluir los pesos de muestreo en todas las técnicas y metodologías estadísticas, ya sean regresiones simples y logísticas o simples varianzas del promedio.

*Modelos de superpoblación*

Suponga que la estimación de máxima verosimilitud es apropiada para muestras aleatorias simples. Por ejemplo, modelos de regresión simple, múltiple, regresión logística, entre otros. Bajo este esquema, se asume que la función de densidad poblacional es donde es el parámetro de interés. Con una réplica del ejemplo que David Binder utiliza en un artículo del año 2011 (una excelente lectura para quienes ha seguido el trabajo de Ken Brewer), se introducen algunos conceptos que son de utilidad. Finalmente, todos los resultados se van a plasmar en simulaciones de Monte Carlo, algunas veces anidadas.

Suponga que se generaron realizaciones de variables aleatorias independientes distribuidas Bernoulli con parámetro de interés . Los datos que se obtienen se muestran a continuación:

1 1 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1 0 0 0 1 1 0 0 0 1 1 0 0 1 0 0 0 1 0 0 1 1 1 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0

En esta población finita, que fue generada a partir de un modelo probabilístico (llamado modelo de superpoblación), hay 28 éxitos.

*Primer proceso inferencial: el modelo*

En este apartado, es notable que la medida de probabilidad que rige la inferencia hasta el momento sea la inducida por la distribución binomial con parámetro 0.3. De esta manera, el estimador insesgado de mínima varianza (todas estas propiedades obtenidas con base en la distribución binomial) está dado por el promedio poblacional. Nótese que la inferencia utiliza todos los datos de la población. Ahora, para reproducirlo computacionalmente, basta con simular muchas poblaciones de 100 variables aleatorias independientes distribuidas Bernoulli con parámetro desconocido =0.3.

Como es bien sabido, bajo la perspectiva de los modelos poblacionales y la inferencia estadística clásica, el estimador es insesgado. Para corroborarlo, es posible introducir la siguiente simulación de Monte Carlo.

N = 100  
theta = 0.3  
nsim1 = 1000  
Est0=rep(NA,nsim1)  
  
for(i in 1:nsim1){  
y=rbinom(N, 1, theta)  
Est0[i]=mean(y)  
}  
  
Esp0 = mean(Est0)  
  
cbind(theta, Esp0)

| theta | Esp0 |
| --- | --- |
| 0.3 | 0.2992 |

*Segundo proceso inferencial: el muestreo*

En el primer proceso inferencial, se asume que las variables de estudio son realizaciones de variables aleatorias gobernadas por un modelo probabilístico. Sin embargo, un razonamiento muy válido es que en cualquier población finita en particular, los valores de la medición son fijos aunque desconocidos y no siguen ningún modelo probabilístico; es decir, no corresponden a realizaciones de variables aleatorias. Por ejemplo, suponga que para esa misma población del ejemplo anterior el dato uno corresponde a un individuo desempleado y el dato cero corresponde a un individuo empleado.

Por otra parte, asuma que la población está subdividida en conglomerados, que pueden ser llamados hogares. De esta forma, nuestra población finita toma la siguiente caracterización, mediante una partición de hogares:

(1 1 0) (1 0) (0 0 0 0 0 0 1) (1 0) (0 0 0 0 0 0 1) (0 0 1) (0 0 0 0 0 0 0 1) (0 0 1) (0 0 0 1) (0 0 0 0 1) (0 0 0 0 0 0 0 1) (1 0) (1 0) (0 0 1) (1 0) (0 0 1) (1 0) (0 1) (0 0 0 1) (0 0 1) (1 1 0) (0 0 0 0 1) (0 1) (0 1) (0 0 0 0 0 0 0 0 0 1) (0 1) (0)

El proceso de aglomeración en hogares es obviamente artificioso en este ejemplo, pero ilustra que en la vida real las poblaciones finitas siempre están aglomeradas. Suponga por otra parte que tomamos una muestra de hogares y en cada hogar seleccionado realizamos un censo; además la selección de los hogares se hará aleatoriamente, sin reemplazo y con probabilidades de inclusión proporcionales al tamaño del hogar . Siendo la característica de interés , el estado del individuo en la fuerza laboral (1, si está desempleado y 0, en otro caso); entonces es bien sabido que bajo este esquema de muestreo un estimador insesgado para la proporción de desempleados es el siguiente:

En donde es la proporción de desempleados en el hogar -ésimo, es el total de desempleados en el hogar -ésimo, es el número de individuos en el hogar y es el número de hogares seleccionados. Por otro lado, un estimador ingenuo, correspondiente a la proporción de desempleados en la muestra, que asume que el agrupamiento de los valores no interfiere en el proceso de inferencia e ignora el diseño de muestreo es el siguiente:

En términos generales el siguiente esquema trata de reproducir gráficamente este proceso de inferencia, en donde un gran número de muestras podrían haber sido extraídas siguiendo el diseño de muestreo. Con la siguiente simulación de Monte Carlo se comprueba fácilmente que es insesgado, mientras que es sesgado:

library(TeachingSampling)  
N=100  
theta=0.3  
y=rbinom(N, 1, theta)  
theta\_N=mean(y)  
nsim2=1000  
Est1=Est2=rep(NA,nsim2)  
  
#-----Creación de los clusters---------  
  
clus=c(0,which((y[-N]-y[-1])!=0)+1)  
NI=(length(clus)-1)  
Ind=matrix(0, nrow=N, ncol=NI)  
Tamaños=clus[-1]-clus[-(length(clus))]  
  
for(l in 1:(length(clus)-1)){  
a=(clus[l]+1):clus[l+1]  
Ind[a,l]=a  
}  
  
#Tamaños  
  
nsim2=1000  
nI=floor(NI\*0.3)  
  
for(j in 1:nsim2){  
res <- S.piPS(nI,Tamaños)  
sam <- res[,1]   
Ind.sam=Ind[,sam]  
Tamaños.sam=Tamaños[sam]  
#-------Espacio para las medias  
medias=matrix(NA)  
for(k in 1:ncol(Ind.sam)){  
medias[k]=mean(y[Ind.sam[,k]])  
}  
#-------  
Est1[j]=mean(medias)  
Est2[j]=sum(Tamaños.sam\*medias)/sum(Tamaños)  
}  
  
Esp1=mean(Est1)  
Esp2=mean(Est2)  
  
cbind(theta\_N, Esp1, Esp2)

| theta\_N | Esp1 | Esp2 |
| --- | --- | --- |
| 0.29 | 0.2907087 | 0.11143 |

Nótese que el primer estimador es insesgado (su esperanza equivale al parámetro de la población finita) porque es función del inverso de la probabilidad de inclusión de los elementos que son inducidas por la medida de probabilidad definida por el plan de muestreo. El segundo estimador es sesgado porque no tiene en cuenta el diseño de muestreo.

*Inferencia doble: los modelos y el muestreo*

En último lugar, suponga que los valores de las variables de interés sí constituyen realizaciones de variables aleatorias que siguen un modelo probabilístico. Como una población finita está constituida por la realización particular de las variables aleatorias, condicionado a la realización de una población finita, se extrae una muestra aleatoria de elementos, mediante un diseño de muestreo complejo. Nótese que, en este tercer proceso inferencial, tanto el modelo como el diseño de muestreo como la medida de probabilidad que da origen a las superpoblaciones, constituyen dos medidas de probabilidad distintas que deben regir la inferencia del parámetro de interés.

Al respecto, nótese que, dado que el diseño de muestreo es complejo, no es viable utilizar técnicas clásicas, como el método de máxima verosimilitud, puesto que los datos finales no constituyen una muestra aleatoria de variables independientes ni idénticamente distribuidas. Por lo anterior, la forma final de la función de verosimilitud, definida como la densidad conjunta de las variables en la muestra, será muy compleja, intratable e insoluble. Una solución a este problema de estimación es la técnica de máxima pseudo-verosimilitud, la cual induce estimadores que tienen en cuenta las ponderaciones del diseño de muestreo complejo. Para el ejemplo de las proporciones, el estimador cumple la siguiente relación:

Con la siguiente simulación de Monte Carlo se comprueba fácilmente que es insesgado, mientras que es sesgado:

library(TeachingSampling)  
  
N=100  
theta=0.3  
nsim1=100  
Esp1=Esp2=rep(NA,nsim1)  
  
for(i in 1:nsim1){  
y=rbinom(N, 1, theta)  
#-----Creación de los clusters---------  
clus=c(0,which((y[-N]-y[-1])!=0)+1)  
NI=(length(clus)-1)  
Ind=matrix(0, nrow=N, ncol=NI)  
Tamaños=clus[-1]-clus[-(length(clus))]  
  
for(l in 1:(length(clus)-1)){  
a=(clus[l]+1):clus[l+1]  
Ind[a,l]=a  
}  
  
Ind  
Tamaños  
  
nsim2=100  
nI=floor(NI\*0.3)  
Est1=Est2=rep(NA,nsim2)  
  
for(j in 1:nsim2){  
res <- S.piPS(nI,Tamaños)  
sam <- res[,1]   
sam  
Ind.sam=Ind[,sam]  
Tamaños.sam=Tamaños[sam]  
#-------Espacio para las medias  
medias=matrix(0)  
for(k in 1:ncol(Ind.sam)){  
medias[k]=mean(y[Ind.sam[,k]])  
}  
  
Est1[j]=mean(medias)  
Est2[j]=sum(Tamaños.sam\*medias)/sum(Tamaños)  
}  
  
Esp1[i]=mean(Est1)  
Esp2[i]=mean(Est2)  
  
}  
  
cbind(theta, mean(Esp1), mean(Esp2))

| theta |  |  |
| --- | --- | --- |
| 0.3 | 0.3098527 | 0.1155673 |

Por supuesto que, dado que el proceso de inferencia es doble, entonces este ejercicio de Monte Carlo debe ser anidado. Es decir, muchas simulaciones dentro de una simulación. Nótese que en primer lugar se debe generar todas las poblaciones finitas y para cada una de ellas se debe generar las posibles muestras.

Los métodos que se explicarán en este capítulo serán la estimación por Máxima Verosimilitud (MV) y Máxima Pseudo Verosimilitud (MPV) para modelos de regresión. El primer método se basa en estimar un parámetro desconocido suponiendo que las variables de interés constituyen una muestra aleatoria de variables independiente e idénticamente distribuidas (IID) para poder hacer inferencia sobre la población de interés. Por otra parte el método de Máxima Pseudo Verosimilitud sigue un razonamiento parecido, pero con la gran diferencia de que la variable de interés se rige por un diseño muestral específico, lo cual induce una probabilidad de inclusión del individuo que debe ser tenida en cuenta al momento de realizar cualquier tipo de inferencia.

## 8.2 Método de Máxima Verosimilitud

Uno de los métodos más utilizados en la estadística para estimar parámetros es el método de Máxima Verosimilitud, para utilizar este método debemos conocer la función de distribución de las variables de interés. Luego, si una muestra aleatoria de las variable de interés que siguen una distribución . Por lo tanto, la función de verosimilitud está dada por:

Para un mejor manejo de esta función se sugiere aplicar propiedades de los logaritmos generando la siguiente función

Calculando las derivadas con respecto a e igualando a cero tenemos el siguiente sistema de ecuaciones

Ahora, definiendo a , entonces el sistema de ecuaciones tendría la siguiente forma:

es conocido el puntaje o de la unidad -ésima. La solución de este sistema de ecuaciones, notada como , es conocida como el Estimador de Máxima Verosimilitud. Una bondad de este método es que podemos obtener una varianza asintótica del modelo , de la siguiente manera

donde,

Como el anterior término depende del parámetro, un estimador consistente estaría dado por:

donde,

*MV para una distribución Bernoulli*

En el ejemplo introductorio que sirvió como punto de partida para esta discusión, se habló de que los datos de naturaleza pueden ser modelados mediante una distribución Bernoulli, con parámetro de éxito . De esta forma, la función de verosimilitud está dada por:

Luego, aplicando logaritmo, se tiene que:

Por lo tanto, las ecuaciones de verosimilitud, definidas en función de las variables de puntaje () son:

En donde, . Por tanto, igualando a cero, se obtiene que

De lo cual se obtiene el estimador de máxima verosimilitud dado por:

Con varianza estimada dada por:

En donde,

Es decir que la estimación de la varianza para es . En donde, .

*MV para una distribución normal*

Ahora se ilustrará el método de Máxima Verosimilitud suponiendo la siguiente función de distribución de un variable aleatoria con distribución normal

Conociendo la función de distribución llegamos a la probabilidad conjunta

Con un poco de álgebra llegamos a esta expresión

Aplicando logaritmos

Maximizando la anterior expresión llegamos a obtener el score

igualando a cero despejamos y tenemos

Llegamos a que una estimación por el método de Máxima Verosimilitud, para la función que sigue una función de distribución normal, es el promedio poblacional .

*MV para una regresión lineal múltiple*

En un entorno matricial se puede tener en cuenta más de una variable predictora llevándonos a un modelo de regresión múltiple donde no solamente las variables son continuas, sino que también pueden ser categóricas. A continuación, se presenta la estimación de parámetros del modelo.

El modelo adopta la forma , disponemos de como una matriz de dimensión , donde es el tamaño de muestra e es el número de variables predictoras, también se define un vector de tamaño como la variable de interés y, por último, un vector de tamaño . Suponiendo que sigue una distribución normal tenemos la siguiente función:

Conociendo la anterior función de distribución, llegaremos a la probabilidad conjunta de

Con un poco de algebra matricial se llega a:

Aplicando propiedades de logaritmos nos queda la siguiente expresión:

Maximizando el anterior resultado podemos llegar al score :

Igualando a cero la derivada y despejando llegamos a la estimación de :

El anterior resultado es la estimación general de en una regresión múltiple, obtenida bajo el método de Máxima Verosimilitud.

## 8.3 Método de Máxima Pseudo-Verosimilitud

El anterior método tiene la particularidad de que son IID, en la vida real muchas veces no es posible poder cumplir ese supuesto. En los procedimientos actuales se recurre a obtener una muestra compleja mediante la realización de conglomerados que tengan alguna relación particular de aglomeración para luego estratificarlos y llegar al individuo de interés que nos proporcione información sobre el estudio. Con esa muestra compleja cumplimos con que todos los individuos tienen una probabilidad de inclusión desigual sin la necesidad de utilizar un marco muestral.

A partir de eso *Pfeffermann(1993)* discutió la posibilidad de hacer inferencia en la población partiendo de la información de una muestra, para esto se propuso crear un pseudo-parámetro que tenga en cuenta el diseño muestral, es decir, que el score sea ponderado por el inverso de la probabilidad de inclusión que denominaremos . Este método es conocido como Máxima Pseudo Verosimilitud(MPV).

Para un mejor manejo de esta función se sugiere aplicar propiedades de los logaritmos generando la siguiente función:

Calculando las derivadas parciales de con respecto a e igualando a cero tenemos un sistema de ecuaciones como sigue:

donde es el vector de “score” de elementos ponderado por , ahora definiremos como:

Mediante la linealización de Taylor y considerando los resultados de *Binder(1983)*, podemos obtener una varianza asintóticamente insesgada de la siguiente forma:

donde,

La estimación de la varianza anterior está definida por:

donde,

*MPV para una distribución Bernoulli*

Las ecuaciones de verosimilitud dadas anteriormente, conllevan a aplicar la técnica de pseudo-verosimilitud, para la cual, en primer lugar, se definen:

Luego, las ecuaciones de pseudo-verosimilitud son:

Por lo tanto, al igualar a cero, se tiene que:

Por lo anterior, al despejar, se tiene que el estimador de máxima pseudo-verosimilitud, está dado por:

Luego, el estimador de la varianza de es:

donde

Por ejemplo, bajo un muestreo aleatorio simple sin reemplazo, se tiene que el estimador de máxima pseudo-verosimilitud es . Además, la estimación de su varianza es:

Luego, teniendo en cuenta que bajo este diseño de muestreo, se tiene que y que , entonces el estimador de la varianza de es:

Nótese que la anterior expresión, coincide plenamente con la estimación de la varianza de la media muestral, es decir .

*MPV para una distribución normal*

Siguiendo el mismo orden de la sección de Máxima Verosimilitud, se ilustrará el método de Máxima Pseudo Verosimilitud, suponga que sigue una función de distribución normal.

Aplicaremos la productoria para llegar a la probabilidad conjunta:

Con algo de algebra llegamos a:

Utilizamos logaritmos tenemos:

Maximizamos la anterior expresión con derivadas parciales tenemos:

Despejando , se llega a un resultado interesante:

Esto nos conlleva que, para la función una estimación es el promedio muestral ponderado.

*MPV para una regresión múltiple*

Con el modelo de la forma se tiene una matriz de dimensión , donde es el tamaño de la muestra e es el número de variables predictoras, también una matriz diagonal, con los , de tamaño , y, por último, se define dos vectores, uno de tamaño como la variable de interés y otro de tamaño . Con estas condiciones se puede definir una función de verosimilitud de la siguiente manera. Conociendo la función de distribución normal de

Se halla la probabilidad conjunta matricialmente:

Simplificando la anterior expresión se llega a:

Para poder derivar mejor, se aplica propiedades de los logaritmos:

Maximizando el anterior resultado conoceremos el score :

Despejando tenemos el siguiente resultado:

Con este podemos estimar un modelo partiendo de una muestra probabilística compleja.

A continuación, se ejemplifican los códigos computacionales para los modelos lineales generalizado. Inicialmente, se carga la base de datos como sigue:

Cargue de las bases de datos,

library(survey)  
library(srvyr)  
encuesta <- readRDS("Data/encuesta.rds")  
data("BigCity", package = "TeachingSampling")

diseno <- encuesta %>%  
 as\_survey\_design(  
 strata = Stratum,  
 ids = PSU,  
 weights = wk,  
 nest = T  
 )

Luego, se define la variable pobreza con los datos de la base como sigue:

diseno <- diseno %>%   
 mutate(  
 pobreza = ifelse(Poverty != "NotPoor", 1, 0),  
 desempleo = ifelse(Employment == "Unemployed", 1, 0))

A continuación, se carga el tema de la CEPAL con el que se realizarán las gráficas:

Para la ejemplificación de la metodología de este capítulo, se definirá primero el modelo Gamma.

*Modelo gamma para variable continua*

La función de enlace para el GLM con una variable dependiente distribuida por un modelo Gamma es el recíproco, . Eso significa que el valor esperado de observado, (), está relacionado con sus variables de entrada como, por ejemplo,

o, similarmente:

Ahora bien, para el ajuste del modelo Gamma, primero se definen los pesos qweigth como sigue:

mod\_qw <- lm(wk ~ Age + Sex + Region + Zone,  
 data = encuesta)  
  
encuesta$wk2 <- encuesta$wk/predict(mod\_qw)  
  
diseno <- encuesta %>%  
 as\_survey\_design( strata = Stratum,  
 ids = PSU, weights = wk2,  
 nest = T)

El modelo ajustado es el siguiente:

modelo <- svyglm(formula = Income ~ Age + Sex +  
 Region + Zone,  
 design = diseno,   
 family = Gamma(link = "inverse"))   
broom::tidy(modelo)

| term | estimate | std.error | statistic | p.value |
| --- | --- | --- | --- | --- |
| (Intercept) | 0.0024054 | 0.0002192 | 10.9726242 | 0.0000000 |
| Age | -0.0000018 | 0.0000014 | -1.2837541 | 0.2018776 |
| SexMale | -0.0000910 | 0.0000494 | -1.8422509 | 0.0680837 |
| RegionSur | -0.0000528 | 0.0002279 | -0.2315853 | 0.8172827 |
| RegionCentro | 0.0000305 | 0.0002202 | 0.1382979 | 0.8902533 |
| RegionOccidente | 0.0002363 | 0.0002317 | 1.0196070 | 0.3101123 |
| RegionOriente | 0.0000088 | 0.0002747 | 0.0319219 | 0.9745911 |
| ZoneUrban | -0.0009295 | 0.0001906 | -4.8762260 | 0.0000036 |

De la tabla anterior, el intercepto, la variable edad y la zona rural resultaron significativo tomando una confianza del 95%.

Por otro lado, es útil la estimación de la dispersión que ofrece *svyglm* de forma predeterminada dado que no tiene en cuenta la información especial sobre la dispersión que se puede calcular utilizando la distribución Gamma. **No todos los GLM tienen una forma mejorada y específica del modelo para estimar**.

alpha = MASS::gamma.dispersion(modelo)  
alpha

## [1] 0.4830962

Luego, el parámetro de dispersión es:

mod\_s <- summary(modelo, dispersion = alpha)  
mod\_s$dispersion

## variance SE  
## [1,] 0.59097 0.0876

Los coeficientes del modelo también se pueden obtener de la siguiente manera:

mod\_s$coefficients

|  | Estimate | Std. Error | t value | Pr(>|t|) |
| --- | --- | --- | --- | --- |
| (Intercept) | 0.0024054 | 0.0002192 | 10.9726242 | 0.0000000 |
| Age | -0.0000018 | 0.0000014 | -1.2837541 | 0.2018776 |
| SexMale | -0.0000910 | 0.0000494 | -1.8422509 | 0.0680837 |
| RegionSur | -0.0000528 | 0.0002279 | -0.2315853 | 0.8172827 |
| RegionCentro | 0.0000305 | 0.0002202 | 0.1382979 | 0.8902533 |
| RegionOccidente | 0.0002363 | 0.0002317 | 1.0196070 | 0.3101123 |
| RegionOriente | 0.0000088 | 0.0002747 | 0.0319219 | 0.9745911 |
| ZoneUrban | -0.0009295 | 0.0001906 | -4.8762260 | 0.0000036 |

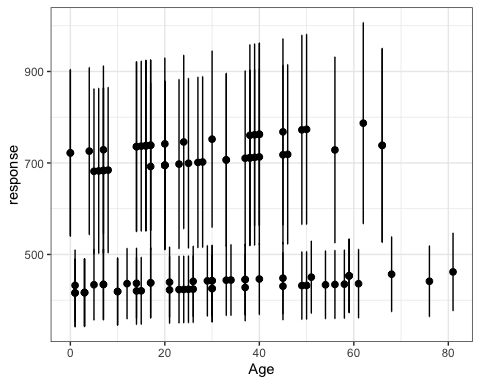
Una vez estimado los coeficientes, se estiman los intervalos de confianza para la predicción como sigue:

pred <- data.frame(predict(modelo, type = "response", se = T))  
  
pred\_IC <- data.frame(confint(predict(modelo, type = "response", se = T)))  
colnames(pred\_IC) <- c("Lim\_Inf", "Lim\_Sup")  
pred <- bind\_cols(pred, pred\_IC)  
pred$Income <- encuesta$Income  
pred$Age <- encuesta$Age  
pred %>% slice(1:6L)

| response | SE | Lim\_Inf | Lim\_Sup | Income | Age |
| --- | --- | --- | --- | --- | --- |
| 456.8271 | 41.80366 | 374.8934 | 538.7607 | 409.87 | 68 |
| 434.3813 | 38.07031 | 359.7649 | 508.9977 | 409.87 | 56 |
| 423.5225 | 37.33444 | 350.3483 | 496.6966 | 409.87 | 24 |
| 441.2123 | 39.35438 | 364.0792 | 518.3455 | 409.87 | 26 |
| 416.6866 | 37.77570 | 342.6476 | 490.7257 | 409.87 | 3 |
| 436.1285 | 38.36471 | 360.9351 | 511.3219 | 823.75 | 61 |

A continuación, como se ha mostrado anteriormente, se realiza un Scaterplot de la predicción:

pd <- position\_dodge(width = 0.2)  
ggplot(pred %>% slice(1:100L),  
 aes(x = Age , y = response)) +  
 geom\_errorbar(aes(ymin = Lim\_Inf,  
 ymax = Lim\_Sup),  
 width = .1,  
 linetype = 1) +  
 geom\_point(size = 2, position = pd) +  
 theme\_bw()



## 8.4 Modelo multinomial

En capítulos anteriores de este libro se desarrolló la manera de cómo hacer el análisis de variables aleatorias provenientes de encuestas complejas de tipo binario. Sin embargo, en encuestas complejas también existen variables de tipo multinomial, es decir, que tengan más de dos niveles de respuesta. Para el análisis de este tipo de variables se utiliza el modelo multinomial, el cual se desarrollará en esta sección.

El modelo de regresión logit multinomial es la extensión natural del modelo de regresión logística binomial simple para analizar variables con respuestas que tienen tres o más categorías distintas. Esta técnica es más apropiada para variables de encuesta con categorías de respuesta nominales.

Para el ajuste de un modelo multinomial se debe tener en cuenta que:

* Su variable dependiente debe medirse en el nivel nominal.
* Tiene una o más variables independientes que son continuas, ordinales o nominales (incluidas las variables dicotómicas).
* Tener independencia de las observaciones y la variable dependiente debe tener categorías mutuamente excluyentes y exhaustivas
* No debe haber **multicolinealidad.** La multicolinealidad ocurre cuando tiene dos o más variables independientes que están altamente correlacionadas entre sí.
* Debe haber una relación lineal entre cualquier variable independiente continua y la transformación logit de la variable dependiente.
* No debe haber valores atípicos, valores de apalancamiento elevados o puntos influyentes.

El modelo múltinomial está dado como:

donde es el vector de coeficiente de para la k-ésima categoría.

Para la estimación de los parámetros del modelo logístico se utilizará la máxima verosimilitud como técnica de estimación. Para ello, implica maximizar la siguiente versión multinomial de la función de pseudoverosimilitud *(Heeringa)*:

donde,

si para la unidad muestreada , 0 en caso contrario. es la probabilidad estimada de que . es el peso de la encuesta para la unidad muestreada .

La maximización implica la aplicación del algoritmo de Newton-Raphson para resolver el siguiente conjunto de ecuaciones de estimación , suponiendo un diseño de muestra complejo con estratos indexados por y conglomerados dentro de estratos indexados por :

Donde,

y

Por último, la matriz de varianza-covarianza de los parámetros estimados, basada en la aplicación de Binder de la linealización de la serie de Taylor a las estimaciones derivadas utilizando la estimación de pseudomáxima verosimilitud es:

Para ejemplificar el uso de los modelos multinomiales en encuestas de hogares utilizando R se utilizan los siguientes códigos computacionales. Inicialemente, para efectos de comparación, estimemos el porcentaje de personas por desempleo:

diseno %>% group\_by(Employment) %>%   
 summarise(Prop = survey\_mean(vartype = c("se", "ci")))

| Employment | Prop | Prop\_se | Prop\_low | Prop\_upp |
| --- | --- | --- | --- | --- |
| Unemployed | 0.0322524 | 0.0057006 | 0.0209647 | 0.0435401 |
| Inactive | 0.2734145 | 0.0106850 | 0.2522572 | 0.2945718 |
| Employed | 0.4141869 | 0.0138767 | 0.3867096 | 0.4416642 |
| NA | 0.2801462 | 0.0139885 | 0.2524476 | 0.3078449 |

Ahora bien, se filtran aquellas personas mayores a 15 años y se vuelve a estimar:

diseno %>% filter(Age >= 15)%>% group\_by(Employment) %>%   
 summarise(Prop = survey\_mean(vartype = c("se", "ci")))

| Employment | Prop | Prop\_se | Prop\_low | Prop\_upp |
| --- | --- | --- | --- | --- |
| Unemployed | 0.0448040 | 0.0077955 | 0.0293683 | 0.0602398 |
| Inactive | 0.3798195 | 0.0150308 | 0.3500571 | 0.4095820 |
| Employed | 0.5753764 | 0.0130688 | 0.5494988 | 0.6012540 |

Con lo anterior, se observa un aumento en el porcentaje de personas empleadas en los hogares.

Ahora bien, se realiza la estimación usando el modelo multinomial con la función svy\_vglm del paquete svyVGAM como se muestra a continuación:

library(svyVGAM)  
diseno\_15 <- diseno %>% filter(Age >= 15)  
model\_mul <- svy\_vglm(  
 formula = Employment ~ Age + Sex + Region + Zone,  
 design = diseno\_15,   
 crit = "coef",  
 family = multinomial(refLevel = "Unemployed"))  
model\_mul

## Stratified 1 - level Cluster Sampling design (with replacement)  
## With (238) clusters.  
## Called via srvyr  
## Sampling variables:  
## - ids: PSU   
## - strata: Stratum   
## - weights: wk2   
## Data variables:   
## - HHID (chr), Stratum (chr), NIh (int), nIh (dbl), dI (dbl), PersonID (chr),  
## PSU (chr), Zone (chr), Sex (chr), Age (int), MaritalST (fct), Income (dbl),  
## Expenditure (dbl), Employment (fct), Poverty (fct), dki (dbl), dk (dbl), wk  
## (dbl), Region (fct), CatAge (ord), wk2 (dbl)  
##   
## Call:  
## vglm(formula = formula, family = family, data = surveydata, weights = .survey.prob.weights,   
## crit = "coef")  
##   
##   
## Coefficients:  
## (Intercept):1 (Intercept):2 Age:1 Age:2   
## 2.29035224 2.09308478 0.02474836 0.02073894   
## SexMale:1 SexMale:2 RegionSur:1 RegionSur:2   
## -2.21949718 -0.55633376 -0.43624259 -0.27905123   
## RegionCentro:1 RegionCentro:2 RegionOccidente:1 RegionOccidente:2   
## 0.37132724 0.25576105 0.25356331 0.09279505   
## RegionOriente:1 RegionOriente:2 ZoneUrban:1 ZoneUrban:2   
## 0.61759957 0.47055482 -0.23461616 0.05674419   
##   
## Degrees of Freedom: 3758 Total; 3742 Residual  
## Residual deviance: 2778.983   
## Log-likelihood: -1389.492   
##   
## This is a multinomial logit model with 3 levels

Algo a tener en cuenta es que la función broom::tidy(), que normalmente usamos para limpiar y estandarizar la salida del modelo, no puede ser empleada en este caso, sin embargo, en el link[[5]](#footnote-341) encuentra la función que utilizamos a continuación.

tidy.svyVGAM <- function(  
 x,   
 conf.int = FALSE,   
 conf.level = 0.95,  
 exponentiate = FALSE,   
 ...  
){  
 ret <- as\_tibble(summary(x)$coeftable, rownames = "term")  
   
   
 colnames(ret) <- c("term", "estimate", "std.error", "statistic", "p.value")  
 coefs <- tibble::enframe(stats::coef(x), name = "term", value = "estimate")  
 ret <- left\_join(coefs, ret, by = c("term", "estimate"))  
 if (conf.int){  
 ci <- broom:::broom\_confint\_terms(x, level = conf.level, ...)  
 ret <- dplyr::left\_join(ret, ci, by = "term")  
 }  
 if (exponentiate){ret <- broom:::exponentiate(ret)}  
   
 ret %>%   
 tidyr::separate(term, into = c("term", "y.level"), sep = ":") %>%   
 arrange(y.level) %>%   
 relocate(y.level, .before = term)  
}

Utilizando el código anterior, obtenemos las estimaciones del modelo multinomial:

tab\_model <- tidy.svyVGAM(model\_mul,   
 exponentiate = FALSE,   
 conf.int = FALSE)   
tab\_model

A continuación, se grafica los intervalos de confianza para los coeficientes del modelo utilizando el formato definido para la CEPAL:

library(dotwhisker)  
tab\_model %>%   
 mutate(model = if\_else(  
 y.level == 1,   
 "Inactive",  
 "Employed", ),  
 sig = gtools::stars.pval(p.value)  
 ) %>%   
 dotwhisker::dwplot(  
 dodge\_size = 0.3,  
 vline = geom\_vline(xintercept = 1, colour = "grey60",  
 linetype = 2)  
 ) +   
 guides(color = guide\_legend(reverse = TRUE)) +   
 theme\_bw() + theme(legend.position = "top")

Una forma aternativa de ajustar los modelos de regresión multinomial es usando la función svy\_vglm. Sin embargo, presenta limitaciones para hacer las predicciones con el modelo, por lo tanto, se puede usar como alternativa la función svymultinom. Cabe aclarar que esta función no está en el repositorio de R y hay que instalarla desde un repositorio de github. A continuación, se muestra cómo se realiza la instalación y posteriormente el uso de la función:

#devtools::install\_github("BS1125/CMAverse")  
library(CMAverse)  
model\_mul2 <- svymultinom(  
 formula = Employment ~ Age + Sex + Region + Zone,  
 weights = diseno\_15$variables$wk2,   
 data = diseno\_15$variables  
)   
summary(model\_mul2)$summarydf

|  | Estimate | Std. Error | t value | Pr(>|t|) |
| --- | --- | --- | --- | --- |
| Inactive:(Intercept) | 2.2900932 | 0.5586694 | 4.0991924 | 0.0000432 |
| Inactive:Age | 0.0247522 | 0.0100125 | 2.4721273 | 0.0135199 |
| Inactive:SexMale | -2.2194689 | 0.3162459 | -7.0181738 | 0.0000000 |
| Inactive:RegionSur | -0.4361437 | 0.4258014 | -1.0242890 | 0.3058318 |
| Inactive:RegionCentro | 0.3715204 | 0.4910265 | 0.7566198 | 0.4493733 |
| Inactive:RegionOccidente | 0.2537092 | 0.4552794 | 0.5572605 | 0.5774164 |
| Inactive:RegionOriente | 0.6175485 | 0.5157976 | 1.1972692 | 0.2313540 |
| Inactive:ZoneUrban | -0.2346034 | 0.2906882 | -0.8070620 | 0.4197338 |
| Employed:(Intercept) | 2.0928541 | 0.5427128 | 3.8562827 | 0.0001190 |
| Employed:Age | 0.0207428 | 0.0096497 | 2.1495766 | 0.0317171 |
| Employed:SexMale | -0.5563127 | 0.3053398 | -1.8219460 | 0.0686234 |
| Employed:RegionSur | -0.2789895 | 0.4106178 | -0.6794386 | 0.4969444 |
| Employed:RegionCentro | 0.2559417 | 0.4678575 | 0.5470505 | 0.5844096 |
| Employed:RegionOccidente | 0.0929049 | 0.4438767 | 0.2093035 | 0.8342342 |
| Employed:RegionOriente | 0.4704651 | 0.5050940 | 0.9314407 | 0.3517463 |
| Employed:ZoneUrban | 0.0567466 | 0.2791795 | 0.2032619 | 0.8389525 |

El hacer uso de esta función permite obtener de manera simple la predicción de las probabilidades como se muestra a continuación:

library(tidyverse)  
tab\_pred <- predict(model\_mul2, type = "probs") %>%   
 data.frame()  
tab\_pred %>% slice(1:15)

| Unemployed | Inactive | Employed |
| --- | --- | --- |
| 0.0387205 | 0.2236639 | 0.7376156 |
| 0.0150598 | 0.5948109 | 0.3901293 |
| 0.0310297 | 0.5550689 | 0.4139014 |
| 0.0907946 | 0.1854493 | 0.7237561 |
| 0.0134342 | 0.6005134 | 0.3860523 |
| 0.0317299 | 0.5537176 | 0.4145525 |
| 0.0466669 | 0.2157326 | 0.7376005 |
| 0.0172653 | 0.5878100 | 0.3949247 |
| 0.0790766 | 0.1920709 | 0.7288525 |
| 0.0294856 | 0.2349694 | 0.7355450 |
| 0.0095218 | 0.6169869 | 0.3734913 |
| 0.0621464 | 0.2031545 | 0.7346991 |
| 0.0687482 | 0.1985744 | 0.7326773 |
| 0.0839204 | 0.1892482 | 0.7268314 |
| 0.0730133 | 0.1958009 | 0.7311858 |

Las predicciones del modelo se realizan de la siguiente manera:

diseno\_15$variables %<>%   
 mutate(predicion = predict(model\_mul2))  
  
diseno\_15 %>% group\_by(Employment) %>%   
 summarise(Prop = survey\_mean(vartype = c("se", "ci")))

| Employment | Prop | Prop\_se | Prop\_low | Prop\_upp |
| --- | --- | --- | --- | --- |
| Unemployed | 0.0448040 | 0.0077955 | 0.0293683 | 0.0602398 |
| Inactive | 0.3798195 | 0.0150308 | 0.3500571 | 0.4095820 |
| Employed | 0.5753764 | 0.0130688 | 0.5494988 | 0.6012540 |

De la anterior tabla se puede observar que, el porcentaje de personas en los hogares que están desempleados es del 4.48%, inactivos del 37.9% y empleados del 57%.

# 9 Modelos multinivel en el contexto de encuestas de hogares

Los modelos multinivel, también conocidos como modelos de efectos mixtos o modelos jerárquicos, son una técnica estadística utilizada en el análisis de datos de encuestas de hogares que tienen una estructura jerárquica o multinivel. En estas encuestas, los datos se recopilan a nivel individual (por ejemplo, sobre la edad, el género y la educación de cada miembro del hogar) y a nivel del hogar (por ejemplo, sobre el ingreso del hogar, la propiedad de la vivienda y la ubicación geográfica).

Los modelos multinivel permiten analizar cómo los factores a nivel del hogar y a nivel individual influyen en las respuestas a las preguntas de la encuesta. Por ejemplo, un modelo multinivel podría utilizarse para investigar cómo el ingreso del hogar y la edad de los miembros del hogar influyen en el consumo de alimentos saludables.

En los modelos multinivel, se modelan los efectos aleatorios y fijos. Los efectos fijos representan las relaciones promedio entre las variables, mientras que los efectos aleatorios modelan la variación en estas relaciones entre los hogares. De esta manera, los modelos multinivel permiten tener en cuenta la heterogeneidad en la población y obtener estimaciones más precisas de los efectos de interés.

En resumen, los modelos multinivel son una herramienta valiosa en el análisis de datos de encuestas de hogares al permitir analizar cómo los factores a nivel del hogar y a nivel individual influyen en las respuestas a las preguntas de la encuesta y al tener en cuenta la estructura jerárquica de los datos.

algunas referencias bibliográficas relevantes sobre el uso de modelos multinivel en encuestas de hogares son:

* “Multilevel statistical models” de Harvey Goldstein (2011): Este libro es una referencia clásica para el análisis de datos multinivel, y aborda el uso de modelos jerárquicos en una variedad de contextos, incluyendo encuestas de hogares. El libro cubre tanto modelos de regresión como de varianza-covarianza, y discute temas como la selección de variables, la validación de modelos y la interpretación de resultados.
* “Data analysis using regression and multilevel/hierarchical models” de Andrew Gelman y Jennifer Hill (2006): Este libro también es una referencia popular para el análisis de datos multinivel, y ofrece una introducción accesible a la teoría y la práctica de los modelos jerárquicos. El libro cubre tanto modelos de regresión como de varianza-covarianza, y utiliza ejemplos de encuestas de hogares para ilustrar los conceptos.
* “Multilevel and longitudinal modeling using Stata” de Sophia Rabe-Hesketh y Anders Skrondal (2012): Este libro es una guía práctica para el análisis de datos multinivel y longitudinales utilizando el software estadístico Stata. El libro cubre tanto modelos de regresión como de varianza-covarianza, y utiliza ejemplos de encuestas de hogares para ilustrar los conceptos.
* “A comparison of Bayesian and likelihood-based methods for fitting multilevel models” de William J. Browne y David Draper (2006): Este artículo compara el rendimiento de los enfoques Bayesianos y basados en verosimilitud para ajustar modelos jerárquicos en el contexto de encuestas de hogares. Los autores concluyen que ambos enfoques pueden ser efectivos, pero que el enfoque Bayesiano puede ofrecer mayores ventajas en términos de flexibilidad y precisión.
* “A brief conceptual tutorial of multilevel analysis in social epidemiology: using measures of clustering in multilevel logistic regression to investigate contextual phenomena” de Juan Merlo y otros (2006): Este artículo presenta una introducción a los modelos jerárquicos en el contexto de la epidemiología social, y utiliza ejemplos de encuestas de hogares para ilustrar los conceptos. Los autores discuten cómo los modelos jerárquicos pueden ser utilizados para investigar fenómenos contextuales, como la segregación residencial y las desigualdades de salud.

Para iniciar este capítulo se cargan las librerías necesarias, la base de datos y el tema de la Cepal para realizar los gráficos:

Cargue de librerías:

knitr::opts\_chunk$set(cache = TRUE, warning = FALSE, message = FALSE, error = FALSE)  
options(digits = 4)  
options(tinytex.verbose = TRUE)  
library (survey)  
library(srvyr)  
library(convey)  
library(TeachingSampling)  
library(printr)  
library(stargazer)  
library(broom)  
library(jtools)  
library(modelsummary)  
library(patchwork)  
library(ggplot2)

Cargue de la base de datos:

encuesta <- readRDS("Data/encuesta.rds")

Creando el tema de la CEPAL para generar los gráficos en este capítulo:

theme\_cepal <- function(...) theme\_light(10) +   
 theme(axis.text.x = element\_blank(),  
 axis.ticks.x = element\_blank(),  
 axis.text.y = element\_blank(),  
 axis.ticks.y = element\_blank(),  
 legend.position="bottom",   
 legend.justification = "left",   
 legend.direction="horizontal",  
 plot.title = element\_text(size = 20, hjust = 0.5),  
 ...)

Para efectos de ejemplificar los conceptos que se presentarán en este capítulo, definamos una muestra con 6 estratos como se muestra a continuación:

encuesta\_plot <- encuesta %>%  
 dplyr::select(HHID, Stratum) %>% unique() %>%  
 group\_by(Stratum) %>% tally() %>%   
 arrange(desc(n)) %>% dplyr::select(-n) %>%   
 slice(1:6L) %>%  
 inner\_join(encuesta) %>% filter(Expenditure <700) %>%   
 dplyr::select(Income, Expenditure, Stratum,   
 Sex, Region, Zone)   
encuesta\_plot %>% slice(1:10L)

| Income | Expenditure | Stratum | Sex | Region | Zone |
| --- | --- | --- | --- | --- | --- |
| 697.3 | 296.1 | idStrt017 | Male | Norte | Rural |
| 697.3 | 296.1 | idStrt017 | Female | Norte | Rural |
| 697.3 | 296.1 | idStrt017 | Male | Norte | Rural |
| 697.3 | 296.1 | idStrt017 | Female | Norte | Rural |
| 526.8 | 294.8 | idStrt017 | Male | Norte | Rural |
| 526.8 | 294.8 | idStrt017 | Female | Norte | Rural |
| 526.8 | 294.8 | idStrt017 | Female | Norte | Rural |
| 526.8 | 294.8 | idStrt017 | Male | Norte | Rural |
| 526.8 | 294.8 | idStrt017 | Male | Norte | Rural |
| 526.8 | 294.8 | idStrt017 | Female | Norte | Rural |

A modo introductorio en este capítulo, se comenzará ajustando un modelo lineal cuya variable a modelar son los ingresos de los hogares y cuya variable explicativa son los gastos de los hogares sin considerar el efecto de los estratos del diseño muestral. A continuación, se muestra la gráfica:

library(latex2exp)  
ggplot(data = encuesta\_plot,  
 aes(y = Income, x = Expenditure)) +   
 geom\_jitter() +  
 theme( legend.position="none",  
 plot.title = element\_text(hjust = 0.5)) +  
 geom\_smooth( formula = y ~ x,  
 method = "lm", se = F) +   
 ggtitle(  
latex2exp::TeX("$Ingreso\_{i}\\sim\\hat{\\beta}\_{0}+\\hat{\\beta}\_{1}Gasto\_{i}+\\epsilon\_{i}$")) +  
 theme\_cepal()



Como se pudo observar en la gráfica anterior, el modelo lineal ajustado es común y se ajusta con los métodos estadísticos antes explicados. Como se ha explicado en capítulos anteriores, este modelo se basa en varios supuestos principales con respecto a la naturaleza de los datos en la población; más específicamente asume independencia de las observaciones a algunas variables de interés, por ejemplo, los estratos socioeconómicos.

Naturalmente este supuesto no es válido más aún, cuando la selección de la muestra en muestreo probabilístico se hace por estrato muestral y además, el comportamiento de los estratos muestrales es diferente entre ellos.

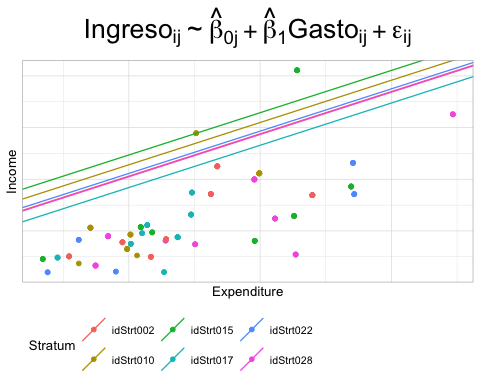
Teniendo en cuenta lo anterior, a continuación, se ajusta un modelo de regresión en donde el intercepto cambia de acuerdo con cada estrato. Es decir, se fija una pendiente común y se varía el intercepto, como se muestra a continuación:

B1 <- coef(lm(Income ~ Expenditure, data = encuesta\_plot))[2]  
(coef\_Mod <- encuesta\_plot %>% group\_by(Stratum) %>%   
 summarise(B0 = coef(lm(Income ~ 1))[1]) %>%   
 mutate(B1 = B1))

| Stratum | B0 | B1 |
| --- | --- | --- |
| idStrt002 | 496.9 | 1.637 |
| idStrt010 | 584.7 | 1.637 |
| idStrt015 | 660.6 | 1.637 |
| idStrt017 | 408.3 | 1.637 |
| idStrt022 | 517.9 | 1.637 |
| idStrt028 | 492.1 | 1.637 |

Ahora, se grafican cada uno de los modelos ajustados previamente:

ggplot(data = encuesta\_plot,  
 aes(y = Income, x = Expenditure,  
 colour = Stratum)) +   
 geom\_jitter() + theme(legend.position="none",  
 plot.title = element\_text(hjust = 0.5)) +  
 geom\_abline(data = coef\_Mod,  
 mapping=aes(slope=B1,   
 intercept=B0,   
 colour = Stratum)) +  
 ggtitle(  
 latex2exp::TeX("$Ingreso\_{ij}\\sim\\hat{\\beta}\_{0j}+\\hat{\\beta}\_{1}Gasto\_{ij}+\\epsilon\_{ij}$"))+  
 theme\_cepal()



En la gráfica anterior Se puede observar que el ajuste no fue el mejor: sin embargo, dicho ajuste se puede ir mejorando a medida que el modelo se vaya afinando e incluyendo efectos adicionales.

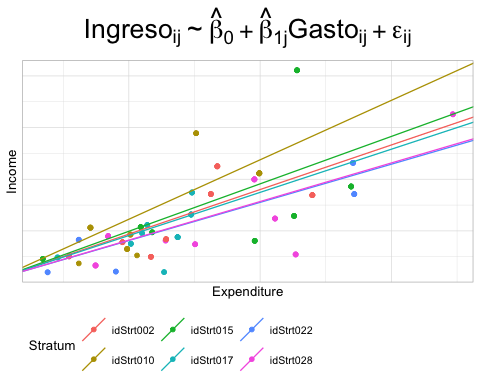
Por otro lado, se ajustará un modelo con pendiente aleatoria. Dicha pendiente se estimará para cada uno de los estratos definidos en el diseño muestral como se presenta a continuación:

B0 <- coef(lm(Income ~ Expenditure,  
 data = encuesta\_plot))[1]  
(coef\_Mod <- encuesta\_plot %>% group\_by(Stratum) %>%   
 summarise(  
 B1 = coef(lm(Income ~ -1 + Expenditure))[1]) %>%   
 mutate(B0 = B0))

| Stratum | B1 | B0 |
| --- | --- | --- |
| idStrt002 | 1.727 | 29.56 |
| idStrt010 | 2.303 | 29.56 |
| idStrt015 | 1.837 | 29.56 |
| idStrt017 | 1.672 | 29.56 |
| idStrt022 | 1.478 | 29.56 |
| idStrt028 | 1.495 | 29.56 |

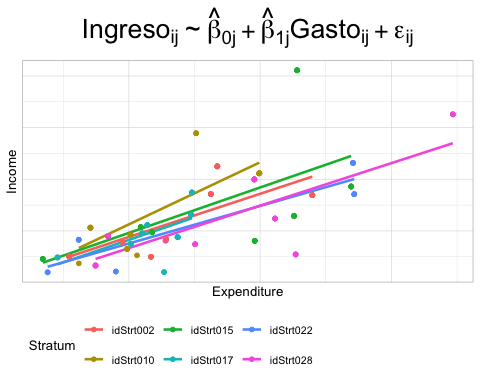
Graficando el modelo con pendiente aleatoria definido anteriormente tenemos:

ggplot(data = encuesta\_plot,  
 aes(y = Income, x = Expenditure,  
 colour = Stratum)) +   
 geom\_jitter() + theme(legend.position="none",  
 plot.title = element\_text(hjust = 0.5)) +  
 geom\_abline(data = coef\_Mod,  
 mapping=aes(slope=B1,   
 intercept=B0, colour = Stratum)) +  
 ggtitle(  
 latex2exp::TeX("$Ingreso\_{ij}\\sim\\hat{\\beta}\_{0}+\\hat{\\beta}\_{1j}Gasto\_{ij}+\\epsilon\_{ij}$"))+  
 theme\_cepal()



A continuación, se genera un gráfico para un modelo con intercepto y pendientes aleatorias,

ggplot(data = encuesta\_plot,  
 aes(y = Income, x = Expenditure,  
 colour = Stratum)) +   
 geom\_smooth( formula = y ~ x, method = "lm", se = F) +   
 geom\_jitter() + theme(legend.position="none",  
 plot.title = element\_text(hjust = 0.5)) +  
 ggtitle(  
 latex2exp::TeX("$Ingreso\_{ij}\\sim\\hat{\\beta}\_{0j}+\\hat{\\beta}\_{1j}Gasto\_{ij}+\\epsilon\_{ij}$"))+  
 theme\_cepal()



Se puede observar que este modelo se ajusta mejor a los datos que el modelo anterior y que el modelo lineal clásico.

Con los anteriores modelos se quería mostrar la importancia de realizar modelos multinivel cuando la naturaleza de los datos es anidadas. Para ejemplificar (Finch, W. H., et al, 2019), suponga que un investigador está interesado en medir el impacto que tendrá un nuevo método de enseñanza en el aprendizaje de los estudiantes. Para esto, planea un diseño muestral en donde selecciona escuelas de manera aleatoria y las ubica en un grupo de tratamiento o de control. En este sentido, si la escuela A se asigna en el grupo tratamiento y dado que el diseño muestral utilizado fue de conglomerados entonces, todos los estudiantes de la escuela también estarán en la condición de tratamiento. Es de tener en cuenta que, las escuelas (conglomerados) son las que se asignan a un grupo en particular y no directamente los estudiantes. Además, se sabe que, la escuela misma, más allá de la condición de tratamiento, tiene un impacto directo en el desempeño de los estudiantes. Este impacto se manifiesta como correlaciones en los puntajes de las pruebas de rendimiento entre las personas que asisten a esa escuela. Por lo tanto, si utilizáramos un ANOVA simple de una vía para comparar las medias de la prueba de rendimiento para los grupos de tratamiento y control con dichos datos muestreados por conglomerados, probablemente estaríamos violando la suposición de errores independientes porque un factor más allá de la condición de tratamiento (en este caso de la escuela) tendría un impacto adicional en la variable de resultado.

Por lo general, nos referimos a la estructura de datos descrita anteriormente como anidada, lo que significa que los puntos de datos individuales en un nivel (por ejemplo, estudiante) aparecen solo en un nivel de una variable de nivel superior, como la escuela. Por lo tanto, los estudiantes están anidados dentro de la escuela. Dichos diseños pueden contrastarse con una estructura de datos cruzados en la que los individuos del primer nivel aparecen en múltiples niveles de la segunda variable.

## 9.1 Modelos multinivel en muestras complejas

En el análisis de los modelos multinivel hay dos tipos de índices que son relevantes:

* Los coeficientes de regresión, generalmente denominados como los parámetros fijos del modelo.
* Las estimaciones de la varianza, generalmente denominadas parámetros aleatorios del modelo.

Cualquier análisis de regresión multinivel siempre debe comenzar con la estimación de la varianza de los Niveles 1 y 2 para la variable dependiente. El primer paso recomendado en el análisis de regresión multinivel consiste en una descomposición de la varianza de la variable dependiente en los diferentes niveles. Por ejemplo, la varianza del ingreso se descompondrá en dos componentes:

* La varianza dentro del estrato
* la varianza entre los estratos.

Estos dos componentes de varianza se pueden obtener en una regresión multinivel. Ahora bien, un modelo básico está dado por:

Con,

Donde,

* Los ingresos de la persona en el estrato .
* El intercepto en el estrato .
* El residual de la persona en el estrato .
* El intercepto general.
* Efecto aleatorio para el intercepto.

Para este modelo se asume que,

y,

.

Para poder continuar con las características de este modelo, se introduce un concepto que se denomina *correlación intra clásica* la cual se calcula como:

La correlación intraclase (ICC, por sus siglas en inglés) hace referencia a la proporción de la varianza total de una variable que se explica por las diferencias entre los grupos (o niveles) en el modelo. En otras palabras, la ICC mide la similitud o correlación entre las observaciones dentro del mismo grupo o nivel en comparación con las observaciones de diferentes grupos.

En un modelo multinivel, los datos están organizados en diferentes niveles, como, por ejemplo, estudiantes dentro de escuelas o pacientes dentro de hospitales. La ICC se calcula para cada nivel de agrupación en el modelo y ayuda a determinar qué tan importante es la variación entre los grupos en comparación con la variación dentro de los grupos.

Una ICC alta indica que una gran proporción de la variación total de la variable se debe a las diferencias entre los grupos, lo que sugiere que los grupos son distintos entre sí y que los efectos de los grupos deben ser considerados en el modelo. Por otro lado, una ICC baja indica que la mayoría de la variación en la variable está dentro de los grupos y que los efectos de los grupos no son tan importantes para explicar la variabilidad en la variable.

Aunque existe evidencia suficiente de que las ponderaciones de muestreo deben usarse en el modelado multinivel (MLM, por sus siglas en inglés) para obtener estimaciones insesgadas[[6]](#footnote-357), y también sobre cómo deben usarse estas ponderaciones en los análisis de un solo nivel, hay poca discusión en la literatura sobre qué y cómo usar pesos de muestreo en MLM.

Actualmente, diferentes autores recomiendan diferentes enfoques sobre cómo usar los pesos de muestreo en modelos jerárquicos.

* Pfefermann et al. (1998) y Asparouhov (2006) aconsejan utilizar un enfoque de pseudomáxima verosimilitud para calcular estimaciones dentro y entre los diferentes niveles utilizando la técnica de maximización de mínimos cuadrados generalizados ponderados por probabilidad (PWGLS) para obtener estimaciones insesgadas.
* Rabe-Hesketh y Skrondal (2006) proporcionan técnicas de maximización de expectativas para maximizar la pseudoverosimilitud.

## 9.2 Estimación de pseudo máxima verosimilitud

EL método de máxima verosimilitud tiene como objetivo estimar los parámetros del modelo que maximizan la probabilidad de que se obtenga la muestra que de hecho se obtuvo. Es decir, los valores de los parámetros estimados deberían maximizar la probabilidad de elegir la muestra que se eligió para realizar el modelo . Lo anterior se realiza con la identificación de dichos valores muestrales mediante la comparación de los datos observados con los predichos por el modelo asociado a los valores de los parámetros. Cuanto más cerca estén entre sí los valores observados y predichos, mayor será la probabilidad de que los datos observados provengan de una población con parámetros cercanos a los utilizados para generar los valores predichos.

Existe una variante del método de máxima verosimilitud (MLE, por sus siglas en inglés) y es la máxima verosimilitud restringida (RMLE, por sus siglas en inglés), que ha demostrado ser más precisa con respecto a la estimación de parámetros de varianza que MLE (Kreft y De Leeuw, 1998). En particular, los dos métodos difieren con respecto a cómo se calculan los grados de libertad en la estimación de las varianzas.

En este sentido y como se ha definido en capítulos anteriores, la función de log-verosimilitud para la población está dada por:

El estimador de máxima verosimilitud esta dada por:

La dificultad que se encuentra aquí es transferir los pesos muéstrales a los niveles inferiores, por ejemplo de UPMs a estratos. Pfeffermann et al. (1998) argumentaron que, debido a la estructura de datos agrupados, ya no se asume que las observaciones sean independientes y que la probabilidad logarítmica se convierta en una suma entre los elementos de nivel uno y dos en lugar de una simple suma de las contribuciones de los elementos.

Pfeffermann et al. (1998) argumentaron que, debido a la estructura de datos agrupados, ya no se asume que las observaciones sean independientes y que la probabilidad logarítmica se convierta en una suma entre los elementos de nivel uno y dos en lugar de una simple suma de las contribuciones de los elementos.

Para ajustar los modelos multinivel en R se usará la función lmer() de la librería lme4.

A continuación se empezará a ejemplificar el ajuste de los modelos multinivel con encuestas complejas, iniciando con el ajuste de un modelo nulo.

*Modelo Nulo*

asuma que la información dentro del estrato está definida por el intercepto,

Como se mostró en capítulos anteriores, para tener estimaciones consistentes se calculan los pesos Qweighted siguiendo los pasos mostrados, tomando en este caso como covariables la edad del entrevistado, el sexo, la región y la zona donde reside. Adicionalmente, se calculan también los *senate-weight* para el ajuste de los modelos (Wk3, en el ajuste del modelo en R).

Al usar pesos para estimar los parámetros de los modelos, se debe tener en cuenta la naturaleza del diseño de muestreo. Particularmente, cuando se trata de estimaciones de parámetros multinivel, debe tener en cuenta no solo los pesos de la unidad de muestreo final sino también los primeros pesos de la unidad de muestreo. Por ejemplo, supongamos que tiene una muestra de estudiantes seleccionados de un marco nacional de escuelas. Luego, tenemos dos conjuntos de pesos, el primero con respecto a las escuelas (observe que una escuela seleccionada se representa así misma y a otros que no están en la muestra) y la segunda con respecto a los estudiantes. Para continuar con el ejemplo, supongamos también que en la población finita se tienen 10.000 estudiantes y 40 escuelas. Consideremos que se han seleccionado a 500 estudiantes asignados en 8 escuelas. En aras de la facilidad, pensemos que se usa una muestra aleatoria simple para seleccionar estudiantes. Si se tiene en cuenta solo los pesos del estudiante para que se ajusten al modelo multinivel, se encontrarán que está estimando parámetros con una muestra ampliada que representa a 10.000 estudiantes que se asignan en una muestra de solo ocho escuelas. Entonces, cualquier conclusión establecida será incorrecta. Por ejemplo, al realizar un análisis de varianza simple, el porcentaje de varianza explicado por las escuelas será extremadamente bajo. Ahora, si se tiene en cuenta ambos conjuntos de pesos (estudiantes y escuelas), se encontrará ajustando un modelo con muestras expandidas que representan 10.000 estudiantes y 40 escuelas, lo cual es lo correcto.

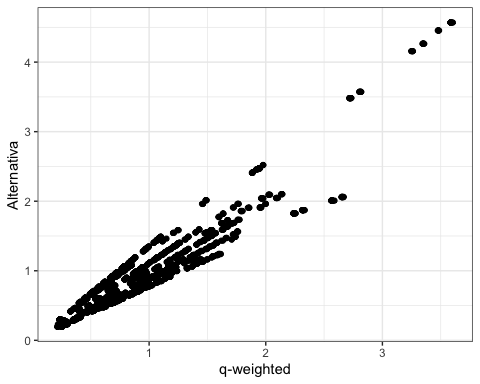
A continuación, se presentan los pesos de muestreos usados para el ajuste del modelo. Primero, se ajustará con los pesos que se tienen directamente del diseño muestral (Wk), los segundos son los pesos q-weigth (wk2) y los senate weigth (wk3).

mod\_qw <- lm(wk ~ Age + Sex + Region + Zone,  
 data = encuesta)  
encuesta$wk2 <- encuesta$wk/predict(mod\_qw)  
  
n = nrow(encuesta)  
encuesta <- encuesta %>% mutate(wk3 = n\*wk/sum(wk))  
encuesta %>% summarise(fep = sum(wk),  
 q\_wei = sum(wk2),  
 fep2 = sum(wk3) )

| fep | q\_wei | fep2 |
| --- | --- | --- |
| 150266 | 2602 | 2605 |

A continuación, se presenta un gráfico comparando los pesos q-weigth y los senate weigth que se usarán para el ajuste del modelo multinivel:

ggplot(encuesta, aes(x = wk2, y = wk3)) +   
 geom\_point() + theme\_bw() +   
 labs(x = "q-weighted", y = "Alternativa")



En este sentido, se realizarán los ajustes de los modelos utilizando los dos pesos mostrados anteriormente:

library(lme4)   
  
mod\_null <- lmer( Income ~ ( 1 | Stratum ),  
 data = encuesta,   
 weights = wk2 )   
  
mod\_null2 <- lmer( Income ~ ( 1 | Stratum ),   
 data = encuesta,   
 weights = wk3 )

Comparando los modelos obtenidos:

coef\_mod\_null <- bind\_cols(coef( mod\_null )$Stratum,   
 coef(mod\_null2 )$Stratum)  
colnames(coef\_mod\_null) <- c("Intercept Mod 1",  
 "Intercept Mod 2")  
coef\_mod\_null %>% slice(1:12)

|  | Intercept Mod 1 | Intercept Mod 2 |
| --- | --- | --- |
| idStrt001 | 630.7 | 630.1 |
| idStrt002 | 505.4 | 506.2 |
| idStrt003 | 481.3 | 484.7 |
| idStrt004 | 959.6 | 954.5 |
| idStrt005 | 514.6 | 515.9 |
| idStrt006 | 433.8 | 438.2 |
| idStrt007 | 467.5 | 470.5 |
| idStrt008 | 371.6 | 376.4 |
| idStrt009 | 207.6 | 218.1 |
| idStrt010 | 591.6 | 592.1 |
| idStrt011 | 588.8 | 588.3 |
| idStrt012 | 352.0 | 361.2 |

Se puede observar que las estimaciones de los interceptos son similares utilizando los dos factores de expansión. Sin embargo, se debe tener en cuenta las características de cada metolodología.

Para efectos de ejemplificar el cálculo de la correlación intraclases, se utiliza la función icc de la librería performance. El cálculo es el siguiente:

performance::icc(mod\_null)

| ICC\_adjusted | ICC\_unadjusted | optional |
| --- | --- | --- |
| 0.3218 | 0.3218 | FALSE |

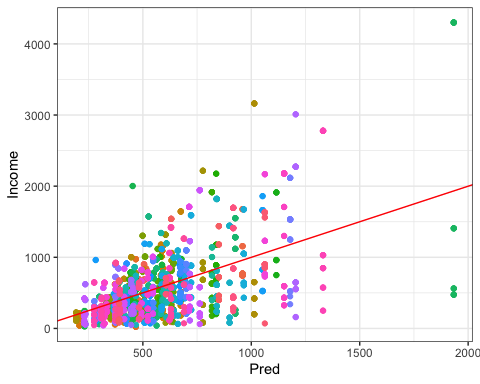
se puede observar que la correlación intraclase, utilizando el modelo nulo es del 32%. Por otro lado, como el modelo que se está ajustando es el “modelo nulo”, la predicción del ingreso dentro de los estrato es constante, como se muestra a continuación:

(tab\_pred <- data.frame(Pred = predict(mod\_null),   
 Income = encuesta$Income,   
 Stratum = encuesta$Stratum)) %>% distinct() %>%   
 slice(1:6L) # Son las pendientes aleatorias

|  | Pred | Income | Stratum |
| --- | --- | --- | --- |
| 1 | 630.7 | 409.87 | idStrt001 |
| 6 | 630.7 | 823.75 | idStrt001 |
| 10 | 630.7 | 90.92 | idStrt001 |
| 13 | 630.7 | 135.33 | idStrt001 |
| 18 | 630.7 | 336.19 | idStrt001 |
| 22 | 630.7 | 1539.75 | idStrt001 |

Como es bien sabido, si la predicción es correcta se espera estar sobre la línea de 45°. Naturalmente, en este caso eso no se obtiene y se puede verificar en la siguiente gráfica:

ggplot(data = tab\_pred, aes(x = Pred, y = Income, colour = Stratum)) +   
 geom\_point() + geom\_abline(intercept = 0, slope = 1, colour = "red") +  
 theme\_bw() + theme(legend.position = "none")



*Modelo con intercepto aleatoria*

Un modelo con pendiente aleatoria es un tipo de modelo de regresión que permite que la relación entre la variable independiente y la variable dependiente cambie según alguna otra variable explicativa. En otras palabras, permite que la pendiente de la relación entre las variables sea diferente en diferentes grupos o subconjuntos de datos.

Por ejemplo, en un modelo de regresión lineal simple, la relación entre la variable independiente (X) y la variable dependiente (Y) se modela como una línea recta con una pendiente fija. Sin embargo, en un modelo con pendiente aleatoria, se permite que la pendiente varíe según otra variable explicativa, como el tiempo, la edad, el género, la ubicación geográfica, etc.

En este tipo de modelos, la relación entre X e Y no se asume como lineal, sino que se puede ajustar a una curva con diferentes pendientes para diferentes subgrupos. Los modelos con pendiente aleatoria son útiles en situaciones donde se espera que la relación entre las variables cambie de manera no lineal o cuando se desea modelar diferencias en la pendiente entre subgrupos. Consideremos el siguiente modelo:

donde esta dado como

Para este caso en particular, se va a variar la pendiente de acuerdo con los estratos socioeconómicos. Para ajustar el modelo se utiliza la función lmer como se muestra a continuación:

mod\_Int\_Aleatorio <- lmer( Income ~ Expenditure + (1 | Stratum),  
 data = encuesta, weights = wk2)  
performance::icc(mod\_Int\_Aleatorio)

| ICC\_adjusted | ICC\_unadjusted | optional |
| --- | --- | --- |
| 0.1958 | 0.1022 | FALSE |

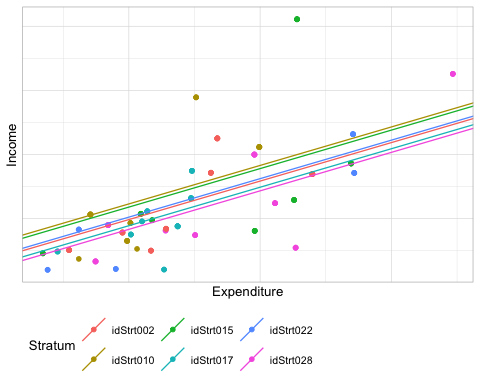
Para cada estrato se tienen las siguientes estimaciones de

coef(mod\_Int\_Aleatorio)$Stratum %>% slice(1:8L)

|  | (Intercept) | Expenditure |
| --- | --- | --- |
| idStrt001 | 248.257 | 1.202 |
| idStrt002 | 152.988 | 1.202 |
| idStrt003 | 139.765 | 1.202 |
| idStrt004 | 292.650 | 1.202 |
| idStrt005 | -42.165 | 1.202 |
| idStrt006 | 46.766 | 1.202 |
| idStrt007 | 2.841 | 1.202 |
| idStrt008 | 103.346 | 1.202 |

Organizando los coeficientes en un gráfico se tiene:

Coef\_Estimado <- inner\_join(  
 coef(mod\_Int\_Aleatorio)$Stratum %>%   
 add\_rownames(var = "Stratum"),  
encuesta\_plot %>% dplyr::select(Stratum) %>% distinct())  
  
ggplot(data = encuesta\_plot,  
 aes(y = Income, x = Expenditure,  
 colour = Stratum)) +   
 geom\_jitter() + theme(legend.position="none",  
 plot.title = element\_text(hjust = 0.5)) +  
 geom\_abline(data = Coef\_Estimado,  
 mapping=aes(slope=Expenditure,   
 intercept=`(Intercept)`,   
 colour = Stratum))+  
 theme\_cepal()

 Se puede observar que, la estimación de la pendiente varía de manera importante para cada uno de los estratos, por tanto, el ajuste del modelo con pendiente aleatoria es adecuado.

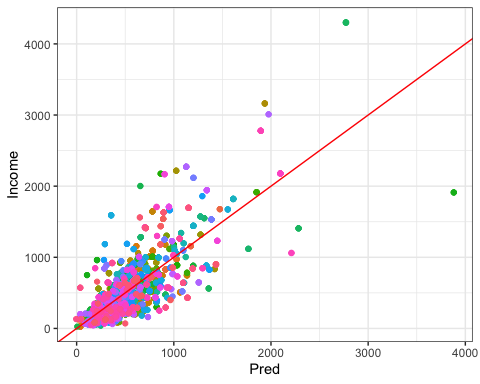
Por otro lado, la predicción de los ingresos usando este modelo se muestra a continuación:

(tab\_pred <- data.frame(  
 Pred = predict(mod\_Int\_Aleatorio),   
 Income = encuesta$Income,   
 Stratum = encuesta$Stratum)) %>% distinct() %>%   
 slice(1:6L)

|  | Pred | Income | Stratum |
| --- | --- | --- | --- |
| 1 | 664.4 | 409.87 | idStrt001 |
| 6 | 719.6 | 823.75 | idStrt001 |
| 10 | 337.3 | 90.92 | idStrt001 |
| 13 | 348.9 | 135.33 | idStrt001 |
| 18 | 560.9 | 336.19 | idStrt001 |
| 22 | 890.5 | 1539.75 | idStrt001 |

Gráficamente se muestran las estimaciones versus los valores estimados de los ingresos y se logra observar que la predicción está más cerca a la línea de 45 grados que el modelo anterior.

ggplot(data = tab\_pred, aes(x = Pred, y = Income, colour = Stratum)) +   
 geom\_point() + geom\_abline(intercept = 0, slope = 1, colour = "red") +  
 theme\_bw() + theme(legend.position = "none")



*Modelo con intercepto y pendiente aleatoria*

Los modelos con intercepto y pendiente aleatoria, también conocidos como modelos mixtos de efectos mixtos o modelos de regresión lineal mixta, son un tipo de modelo estadístico que permite modelar la relación entre una variable de respuesta y una o más variables predictoras, teniendo en cuenta tanto efectos fijos como efectos aleatorios.

En los modelos con intercepto y pendiente aleatoria, los coeficientes de la regresión (es decir, la pendiente y el intercepto) se consideran aleatorios en lugar de fijos. Esto significa que se asume que estos coeficientes pueden variar entre las unidades de análisis, que pueden ser individuos, grupos, regiones geográficas, etc. Estas variaciones se modelan como efectos aleatorios que se incorporan en la ecuación de regresión.

Los modelos con intercepto y pendiente aleatoria son útiles cuando se trabaja con datos que tienen una estructura jerárquica o de agrupamiento, donde las unidades de análisis están agrupadas en diferentes niveles. Por ejemplo, en un estudio sobre el rendimiento académico de los estudiantes, las unidades de análisis pueden ser los estudiantes y las escuelas a las que asisten. En este caso, se puede modelar tanto la variación entre los estudiantes como la variación entre las escuelas en la relación entre el rendimiento académico y los predictores.

Estos modelos son ampliamente utilizados en diversas áreas, como la psicología, la medicina, la sociología, la economía, la biología y la ecología, entre otras.

Para la ejemplificación, consideremos el siguiente modelo:

donde,

y,

El ajuste del modelo se realiza utilizando la función lmer como se presenta a continuación:

mod\_Pen\_Aleatorio <- lmer(Income ~ Expenditure + (1 + Expenditure| Stratum),  
 data = encuesta, weights = wk2)  
  
performance::icc(mod\_Pen\_Aleatorio)

| ICC\_adjusted | ICC\_unadjusted | optional |
| --- | --- | --- |
| 0.6975 | 0.4603 | FALSE |

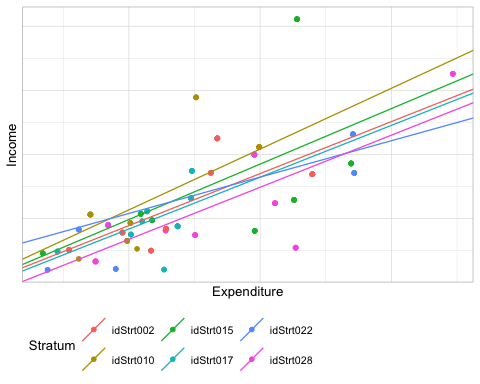
Los coeficientes del modelo son:

coef(mod\_Pen\_Aleatorio)$Stratum %>% slice(1:10L)

|  | (Intercept) | Expenditure |
| --- | --- | --- |
| idStrt001 | -231.46 | 2.7781 |
| idStrt002 | 30.81 | 1.6243 |
| idStrt003 | 152.49 | 1.1621 |
| idStrt004 | 229.89 | 1.3464 |
| idStrt005 | -96.42 | 1.2955 |
| idStrt006 | 30.79 | 1.2036 |
| idStrt007 | 38.00 | 1.0769 |
| idStrt008 | 167.88 | 0.9016 |
| idStrt009 | 30.73 | 0.7496 |
| idStrt010 | 74.05 | 1.8989 |

Gráficamente,

Coef\_Estimado <- inner\_join(  
 coef(mod\_Pen\_Aleatorio)$Stratum %>%   
 add\_rownames(var = "Stratum"),  
encuesta\_plot %>% dplyr::select(Stratum) %>% distinct())  
  
ggplot(data = encuesta\_plot,  
 aes(y = Income, x = Expenditure,  
 colour = Stratum)) +   
 geom\_jitter() + theme(legend.position="none",  
 plot.title = element\_text(hjust = 0.5)) +  
 geom\_abline(data = Coef\_Estimado,  
 mapping=aes(slope=Expenditure,   
 intercept=`(Intercept)`,   
 colour = Stratum))+  
 theme\_cepal()



Como se pudo observar en la gráfica anterior y en el coeficiente de correlación intraclase, el ajuste del modelo con intercepto y pendiente aleatoria se ajusta mejor a los datos que los otros dos modelos mostrados anteriormente.

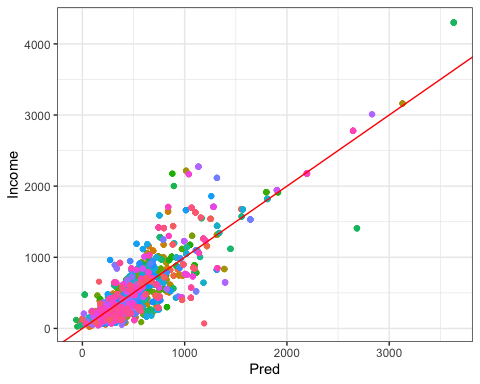
A continuación, se realizan las predicciones de los ingresos con el modelo:

(tab\_pred <- data.frame(Pred = predict(mod\_Pen\_Aleatorio),   
 Income = encuesta$Income,   
 Stratum = encuesta$Stratum)) %>% distinct() %>%   
 slice(1:6L) # Son las pendientes aleatorias

|  | Pred | Income | Stratum |
| --- | --- | --- | --- |
| 1 | 730.727 | 409.87 | idStrt001 |
| 6 | 858.245 | 823.75 | idStrt001 |
| 10 | -25.678 | 90.92 | idStrt001 |
| 13 | 1.325 | 135.33 | idStrt001 |
| 18 | 491.362 | 336.19 | idStrt001 |
| 22 | 1253.269 | 1539.75 | idStrt001 |

Para poder ver qué tan buena son las predicciones, se realiza el siguiente gráfico:

ggplot(data = tab\_pred, aes(x = Pred, y = Income, colour = Stratum)) +   
 geom\_point() + geom\_abline(intercept = 0, slope = 1, colour = "red") +  
 theme\_bw() + theme(legend.position = "none")



Del anterior gráfico se logra observar que las predicciones del ingreso son mejores que las realizadas con los dos modelos anteriores. Lo anterior se debe a la naturaleza misma de los datos en una encuesta de hogares.

Ahora bien, para robustecer el modelo, se ajusta nuevamente, pero agregando la variable zona como se muestra a continuación:

Donde,

y,

donde es el gasto medio de los hogares en el estrato . En R el ajuste se hace de la siguiente manera:

media\_estrato <- encuesta %>% group\_by(Stratum) %>%   
 summarise(mu = mean(Expenditure))  
  
encuesta <- inner\_join(encuesta, media\_estrato, by = "Stratum")   
  
mod\_Pen\_Aleatorio2 <- lmer(Income ~ 1 + Expenditure + Zone + mu +   
 (1 + Expenditure + Zone + mu | Stratum ),  
 data = encuesta, weights = wk2)

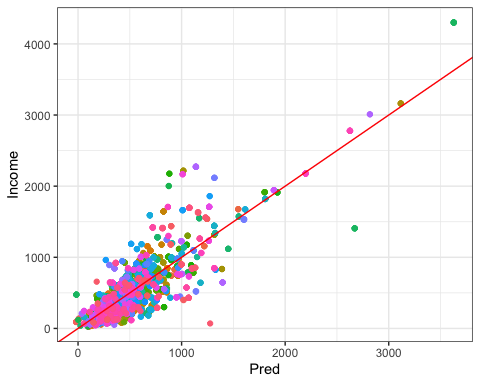
calculando las predicciones de los ingresos de los hogares por estrato:

(tab\_pred <- data.frame(Pred = predict(mod\_Pen\_Aleatorio2),   
 Income = encuesta$Income,   
 Stratum = encuesta$Stratum)) %>% distinct() %>%   
 slice(1:6L)

|  | Pred | Income | Stratum |
| --- | --- | --- | --- |
| 1 | 728.718 | 409.87 | idStrt001 |
| 6 | 854.725 | 823.75 | idStrt001 |
| 10 | -18.730 | 90.92 | idStrt001 |
| 13 | 7.954 | 135.33 | idStrt001 |
| 18 | 492.188 | 336.19 | idStrt001 |
| 22 | 1245.072 | 1539.75 | idStrt001 |

Por último, haciendo el gráfico de las predicciones con los datos observados observándose que el ajuste de este modelo es levemente mejor que el que no incluye la variable zona.

ggplot(data = tab\_pred, aes(x = Pred, y = Income, colour = Stratum)) +   
 geom\_point() + geom\_abline(intercept = 0, slope = 1, colour = "red") +  
 theme\_bw() + theme(legend.position = "none")



A continuación, se presentan todos los componentes que se requieren para escribir el modelo ajustado. Inicialmente, se calcula la matriz diseño:

as.data.frame( model.matrix(mod\_Pen\_Aleatorio2)) %>%   
 distinct()

|  | (Intercept) | Expenditure | ZoneUrban | mu |
| --- | --- | --- | --- | --- |
| 1 | 1 | 346.34 | 0 | 303.3 |
| 6 | 1 | 392.24 | 0 | 303.3 |
| 10 | 1 | 74.07 | 0 | 303.3 |
| 13 | 1 | 83.79 | 0 | 303.3 |
| 18 | 1 | 260.18 | 0 | 303.3 |
| 22 | 1 | 534.43 | 0 | 303.3 |
| 28 | 1 | 256.74 | 1 | 286.2 |
| 32 | 1 | 233.69 | 1 | 286.2 |
| 36 | 1 | 324.97 | 1 | 286.2 |
| 41 | 1 | 334.75 | 1 | 286.2 |
| 46 | 1 | 109.00 | 1 | 286.2 |
| 51 | 1 | 190.27 | 1 | 286.2 |
| 55 | 1 | 479.55 | 1 | 286.2 |
| 61 | 1 | 108.34 | 0 | 270.3 |
| 64 | 1 | 238.34 | 0 | 270.3 |
| 68 | 1 | 451.59 | 0 | 270.3 |
| 73 | 1 | 207.94 | 0 | 270.3 |
| 76 | 1 | 236.24 | 0 | 270.3 |
| 80 | 1 | 278.17 | 0 | 270.3 |
| 84 | 1 | 383.93 | 1 | 569.0 |
| 88 | 1 | 733.67 | 1 | 569.0 |
| 92 | 1 | 982.88 | 1 | 569.0 |
| 95 | 1 | 492.00 | 1 | 569.0 |
| 98 | 1 | 259.14 | 1 | 569.0 |
| 101 | 1 | 555.88 | 0 | 486.1 |
| 105 | 1 | 420.02 | 0 | 486.1 |
| 109 | 1 | 796.05 | 0 | 486.1 |
| 116 | 1 | 463.88 | 0 | 486.1 |
| 126 | 1 | 240.70 | 0 | 486.1 |
| 134 | 1 | 313.90 | 1 | 321.0 |
| 138 | 1 | 120.23 | 1 | 321.0 |
| 139 | 1 | 223.03 | 1 | 321.0 |
| 143 | 1 | 474.20 | 1 | 321.0 |
| 151 | 1 | 119.81 | 1 | 321.0 |
| 154 | 1 | 28.51 | 0 | 376.7 |
| 157 | 1 | 853.57 | 0 | 376.7 |
| 161 | 1 | 109.46 | 0 | 376.7 |
| 163 | 1 | 520.14 | 0 | 376.7 |
| 165 | 1 | 68.98 | 0 | 376.7 |
| 167 | 1 | 144.99 | 1 | 206.4 |
| 171 | 1 | 93.71 | 1 | 206.4 |
| 177 | 1 | 64.84 | 1 | 206.4 |
| 179 | 1 | 376.18 | 1 | 206.4 |
| 183 | 1 | 132.91 | 1 | 206.4 |
| 184 | 1 | 326.18 | 1 | 206.4 |
| 189 | 1 | 59.68 | 0 | 215.9 |
| 192 | 1 | 376.75 | 0 | 215.9 |
| 197 | 1 | 167.66 | 0 | 215.9 |
| 202 | 1 | 329.07 | 0 | 215.9 |
| 208 | 1 | 140.99 | 0 | 215.9 |
| 213 | 1 | 82.71 | 0 | 215.9 |
| 216 | 1 | 197.36 | 1 | 255.8 |
| 220 | 1 | 302.51 | 1 | 255.8 |
| 223 | 1 | 202.45 | 1 | 255.8 |
| 226 | 1 | 398.71 | 1 | 255.8 |
| 233 | 1 | 141.48 | 1 | 255.8 |
| 238 | 1 | 212.50 | 1 | 255.8 |
| 239 | 1 | 123.80 | 1 | 255.8 |
| 240 | 1 | 862.81 | 0 | 483.0 |
| 244 | 1 | 229.26 | 0 | 483.0 |
| 250 | 1 | 628.40 | 0 | 483.0 |
| 257 | 1 | 178.52 | 0 | 483.0 |
| 259 | 1 | 402.06 | 0 | 483.0 |
| 264 | 1 | 325.49 | 1 | 245.0 |
| 266 | 1 | 95.74 | 1 | 245.0 |
| 267 | 1 | 115.42 | 1 | 245.0 |
| 269 | 1 | 277.43 | 1 | 245.0 |
| 271 | 1 | 320.50 | 1 | 245.0 |
| 275 | 1 | 125.28 | 1 | 245.0 |
| 276 | 1 | 497.55 | 0 | 361.9 |
| 286 | 1 | 271.28 | 0 | 361.9 |
| 289 | 1 | 227.36 | 0 | 361.9 |
| 293 | 1 | 132.85 | 0 | 361.9 |
| 296 | 1 | 492.37 | 0 | 361.9 |
| 302 | 1 | 233.50 | 0 | 361.9 |
| 307 | 1 | 719.16 | 1 | 482.1 |
| 310 | 1 | 376.79 | 1 | 482.1 |
| 312 | 1 | 243.75 | 1 | 482.1 |
| 314 | 1 | 904.78 | 1 | 482.1 |
| 319 | 1 | 189.24 | 1 | 482.1 |
| 322 | 1 | 230.27 | 1 | 482.1 |
| 327 | 1 | 538.66 | 0 | 357.0 |
| 332 | 1 | 218.20 | 0 | 357.0 |
| 337 | 1 | 456.25 | 0 | 357.0 |
| 342 | 1 | 451.70 | 0 | 357.0 |
| 346 | 1 | 68.97 | 0 | 357.0 |
| 349 | 1 | 235.59 | 0 | 357.0 |
| 352 | 1 | 391.82 | 0 | 357.0 |
| 356 | 1 | 117.33 | 1 | 334.0 |
| 357 | 1 | 436.66 | 1 | 334.0 |
| 359 | 1 | 467.95 | 1 | 334.0 |
| 363 | 1 | 119.16 | 1 | 334.0 |
| 364 | 1 | 317.60 | 1 | 334.0 |
| 367 | 1 | 203.95 | 1 | 334.0 |
| 369 | 1 | 296.15 | 0 | 244.8 |
| 373 | 1 | 294.81 | 0 | 244.8 |
| 379 | 1 | 253.68 | 0 | 244.8 |
| 381 | 1 | 220.17 | 0 | 244.8 |
| 383 | 1 | 228.06 | 0 | 244.8 |
| 387 | 1 | 203.12 | 0 | 244.8 |
| 390 | 1 | 91.42 | 0 | 244.8 |
| 393 | 1 | 274.38 | 0 | 244.8 |
| 399 | 1 | 421.27 | 1 | 376.3 |
| 405 | 1 | 330.54 | 1 | 376.3 |
| 407 | 1 | 398.10 | 1 | 376.3 |
| 411 | 1 | 313.67 | 1 | 376.3 |
| 414 | 1 | 379.96 | 1 | 376.3 |
| 416 | 1 | 333.83 | 1 | 376.3 |
| 418 | 1 | 156.71 | 0 | 305.8 |
| 424 | 1 | 540.66 | 0 | 305.8 |
| 435 | 1 | 144.88 | 0 | 305.8 |
| 438 | 1 | 583.90 | 0 | 305.8 |
| 442 | 1 | 301.06 | 0 | 305.8 |
| 445 | 1 | 238.03 | 1 | 553.7 |
| 448 | 1 | 402.60 | 1 | 553.7 |
| 456 | 1 | 447.85 | 1 | 553.7 |
| 461 | 1 | 1348.56 | 1 | 553.7 |
| 465 | 1 | 141.40 | 0 | 127.2 |
| 467 | 1 | 119.50 | 0 | 127.2 |
| 471 | 1 | 74.83 | 0 | 127.2 |
| 474 | 1 | 96.00 | 0 | 127.2 |
| 475 | 1 | 192.33 | 0 | 127.2 |
| 478 | 1 | 125.17 | 0 | 127.2 |
| 480 | 1 | 543.36 | 1 | 524.0 |
| 484 | 1 | 123.76 | 1 | 524.0 |
| 487 | 1 | 76.31 | 1 | 524.0 |
| 489 | 1 | 541.50 | 1 | 524.0 |
| 493 | 1 | 180.18 | 1 | 524.0 |
| 495 | 1 | 712.67 | 1 | 524.0 |
| 498 | 1 | 1041.83 | 1 | 524.0 |
| 502 | 1 | 124.25 | 0 | 188.0 |
| 505 | 1 | 380.66 | 0 | 188.0 |
| 511 | 1 | 130.84 | 0 | 188.0 |
| 512 | 1 | 172.93 | 0 | 188.0 |
| 516 | 1 | 91.83 | 0 | 188.0 |
| 522 | 1 | 154.64 | 0 | 188.0 |
| 530 | 1 | 185.75 | 0 | 188.0 |
| 531 | 1 | 246.33 | 1 | 389.3 |
| 535 | 1 | 296.07 | 1 | 389.3 |
| 538 | 1 | 222.35 | 1 | 389.3 |
| 542 | 1 | 87.29 | 1 | 389.3 |
| 543 | 1 | 938.25 | 1 | 389.3 |
| 548 | 1 | 204.71 | 1 | 389.3 |
| 553 | 1 | 293.31 | 0 | 259.0 |
| 557 | 1 | 89.85 | 0 | 259.0 |
| 558 | 1 | 213.39 | 0 | 259.0 |
| 561 | 1 | 295.43 | 0 | 259.0 |
| 567 | 1 | 414.46 | 0 | 259.0 |
| 570 | 1 | 129.85 | 0 | 259.0 |
| 574 | 1 | 160.82 | 1 | 334.1 |
| 576 | 1 | 219.45 | 1 | 334.1 |
| 578 | 1 | 177.59 | 1 | 334.1 |
| 582 | 1 | 179.17 | 1 | 334.1 |
| 584 | 1 | 636.34 | 1 | 334.1 |
| 589 | 1 | 98.65 | 0 | 249.8 |
| 592 | 1 | 301.72 | 0 | 249.8 |
| 599 | 1 | 303.73 | 0 | 249.8 |
| 603 | 1 | 329.32 | 0 | 249.8 |
| 604 | 1 | 163.09 | 0 | 249.8 |
| 607 | 1 | 277.55 | 0 | 249.8 |
| 609 | 1 | 149.28 | 1 | 337.1 |
| 613 | 1 | 422.71 | 1 | 337.1 |
| 617 | 1 | 693.61 | 1 | 337.1 |
| 620 | 1 | 300.97 | 1 | 337.1 |
| 623 | 1 | 256.02 | 1 | 337.1 |
| 625 | 1 | 454.13 | 1 | 337.1 |
| 628 | 1 | 391.25 | 1 | 337.1 |
| 633 | 1 | 168.52 | 1 | 337.1 |
| 639 | 1 | 331.67 | 0 | 238.7 |
| 643 | 1 | 169.60 | 0 | 238.7 |
| 645 | 1 | 262.26 | 0 | 238.7 |
| 649 | 1 | 116.56 | 0 | 238.7 |
| 653 | 1 | 90.94 | 0 | 238.7 |
| 655 | 1 | 314.83 | 0 | 238.7 |
| 661 | 1 | 115.19 | 1 | 282.4 |
| 663 | 1 | 242.92 | 1 | 282.4 |
| 667 | 1 | 309.83 | 1 | 282.4 |
| 671 | 1 | 261.93 | 1 | 282.4 |
| 675 | 1 | 202.87 | 1 | 282.4 |
| 678 | 1 | 399.54 | 1 | 282.4 |
| 684 | 1 | 123.62 | 0 | 263.8 |
| 685 | 1 | 385.49 | 0 | 263.8 |
| 688 | 1 | 314.09 | 0 | 263.8 |
| 691 | 1 | 342.38 | 0 | 263.8 |
| 701 | 1 | 174.96 | 0 | 263.8 |
| 705 | 1 | 129.47 | 0 | 263.8 |
| 711 | 1 | 123.17 | 1 | 455.2 |
| 712 | 1 | 595.53 | 1 | 455.2 |
| 715 | 1 | 315.00 | 1 | 455.2 |
| 716 | 1 | 138.07 | 1 | 455.2 |
| 719 | 1 | 576.54 | 1 | 455.2 |
| 722 | 1 | 614.68 | 1 | 455.2 |
| 726 | 1 | 304.99 | 0 | 218.1 |
| 731 | 1 | 255.87 | 0 | 218.1 |
| 735 | 1 | 239.59 | 0 | 218.1 |
| 739 | 1 | 96.40 | 0 | 218.1 |
| 744 | 1 | 124.90 | 0 | 218.1 |
| 746 | 1 | 258.01 | 0 | 218.1 |
| 753 | 1 | 165.96 | 0 | 218.1 |
| 756 | 1 | 132.19 | 1 | 335.5 |
| 760 | 1 | 221.19 | 1 | 335.5 |
| 764 | 1 | 323.83 | 1 | 335.5 |
| 768 | 1 | 160.73 | 1 | 335.5 |
| 771 | 1 | 663.81 | 1 | 335.5 |
| 775 | 1 | 416.28 | 1 | 335.5 |
| 778 | 1 | 407.22 | 1 | 335.5 |
| 782 | 1 | 209.71 | 0 | 321.9 |
| 792 | 1 | 405.61 | 0 | 321.9 |
| 802 | 1 | 326.32 | 0 | 321.9 |
| 805 | 1 | 342.52 | 0 | 321.9 |
| 810 | 1 | 377.90 | 0 | 321.9 |
| 813 | 1 | 1530.50 | 1 | 686.4 |
| 818 | 1 | 142.61 | 1 | 686.4 |
| 819 | 1 | 935.17 | 1 | 686.4 |
| 822 | 1 | 273.74 | 1 | 686.4 |
| 824 | 1 | 77.81 | 1 | 686.4 |
| 825 | 1 | 245.40 | 1 | 686.4 |
| 830 | 1 | 555.77 | 1 | 686.4 |
| 836 | 1 | 220.67 | 0 | 402.2 |
| 839 | 1 | 525.84 | 0 | 402.2 |
| 845 | 1 | 254.35 | 0 | 402.2 |
| 849 | 1 | 215.09 | 0 | 402.2 |
| 853 | 1 | 172.29 | 0 | 402.2 |
| 856 | 1 | 568.83 | 0 | 402.2 |
| 867 | 1 | 492.63 | 1 | 505.9 |
| 871 | 1 | 287.21 | 1 | 505.9 |
| 872 | 1 | 526.77 | 1 | 505.9 |
| 874 | 1 | 639.26 | 1 | 505.9 |
| 879 | 1 | 667.66 | 1 | 505.9 |
| 884 | 1 | 272.24 | 1 | 505.9 |
| 887 | 1 | 233.79 | 1 | 505.9 |
| 889 | 1 | 3367.53 | 1 | 1311.6 |
| 892 | 1 | 312.13 | 1 | 1311.6 |
| 896 | 1 | 226.50 | 1 | 1311.6 |
| 898 | 1 | 71.05 | 0 | 342.6 |
| 900 | 1 | 440.57 | 0 | 342.6 |
| 904 | 1 | 516.79 | 0 | 342.6 |
| 906 | 1 | 166.56 | 0 | 342.6 |
| 908 | 1 | 402.71 | 0 | 342.6 |
| 912 | 1 | 321.24 | 0 | 342.6 |
| 916 | 1 | 418.53 | 1 | 370.9 |
| 918 | 1 | 560.13 | 1 | 370.9 |
| 923 | 1 | 282.76 | 1 | 370.9 |
| 927 | 1 | 242.68 | 1 | 370.9 |
| 931 | 1 | 91.33 | 1 | 370.9 |
| 933 | 1 | 516.67 | 1 | 370.9 |
| 937 | 1 | 171.67 | 1 | 370.9 |
| 938 | 1 | 567.89 | 0 | 517.1 |
| 943 | 1 | 380.40 | 0 | 517.1 |
| 948 | 1 | 437.62 | 0 | 517.1 |
| 952 | 1 | 308.07 | 0 | 517.1 |
| 956 | 1 | 912.84 | 0 | 517.1 |
| 960 | 1 | 322.27 | 1 | 359.9 |
| 965 | 1 | 380.67 | 1 | 359.9 |
| 967 | 1 | 349.11 | 1 | 359.9 |
| 971 | 1 | 371.38 | 1 | 359.9 |
| 976 | 1 | 454.20 | 1 | 359.9 |
| 983 | 1 | 172.72 | 1 | 359.9 |
| 985 | 1 | 282.56 | 1 | 359.9 |
| 987 | 1 | 223.55 | 0 | 224.4 |
| 995 | 1 | 147.35 | 0 | 224.4 |
| 999 | 1 | 288.59 | 0 | 224.4 |
| 1007 | 1 | 220.72 | 0 | 224.4 |
| 1011 | 1 | 41.26 | 0 | 224.4 |
| 1012 | 1 | 559.33 | 1 | 311.0 |
| 1016 | 1 | 217.71 | 1 | 311.0 |
| 1018 | 1 | 129.08 | 1 | 311.0 |
| 1019 | 1 | 484.33 | 1 | 311.0 |
| 1021 | 1 | 241.66 | 1 | 311.0 |
| 1035 | 1 | 152.06 | 0 | 364.2 |
| 1038 | 1 | 255.48 | 0 | 364.2 |
| 1043 | 1 | 604.25 | 0 | 364.2 |
| 1049 | 1 | 277.41 | 0 | 364.2 |
| 1052 | 1 | 616.32 | 1 | 1124.8 |
| 1055 | 1 | 1385.65 | 1 | 1124.8 |
| 1059 | 1 | 245.28 | 1 | 1124.8 |
| 1063 | 1 | 1791.50 | 1 | 1124.8 |
| 1069 | 1 | 137.27 | 0 | 199.8 |
| 1072 | 1 | 94.66 | 0 | 199.8 |
| 1074 | 1 | 181.66 | 0 | 199.8 |
| 1075 | 1 | 158.54 | 0 | 199.8 |
| 1077 | 1 | 449.20 | 0 | 199.8 |
| 1079 | 1 | 398.27 | 1 | 706.2 |
| 1084 | 1 | 364.26 | 1 | 706.2 |
| 1088 | 1 | 509.08 | 1 | 706.2 |
| 1095 | 1 | 1053.00 | 1 | 706.2 |
| 1099 | 1 | 1431.41 | 1 | 706.2 |
| 1103 | 1 | 305.82 | 0 | 301.7 |
| 1108 | 1 | 58.25 | 0 | 301.7 |
| 1110 | 1 | 183.36 | 0 | 301.7 |
| 1112 | 1 | 407.40 | 0 | 301.7 |
| 1120 | 1 | 336.90 | 0 | 301.7 |
| 1125 | 1 | 221.86 | 0 | 301.7 |
| 1129 | 1 | 112.09 | 1 | 213.0 |
| 1131 | 1 | 307.20 | 1 | 213.0 |
| 1135 | 1 | 231.25 | 1 | 213.0 |
| 1138 | 1 | 191.77 | 1 | 213.0 |
| 1141 | 1 | 162.08 | 1 | 213.0 |
| 1143 | 1 | 148.33 | 1 | 213.0 |
| 1144 | 1 | 264.78 | 0 | 367.2 |
| 1152 | 1 | 210.43 | 0 | 367.2 |
| 1156 | 1 | 984.08 | 0 | 367.2 |
| 1160 | 1 | 237.22 | 0 | 367.2 |
| 1165 | 1 | 274.55 | 0 | 367.2 |
| 1169 | 1 | 403.28 | 1 | 340.3 |
| 1172 | 1 | 290.73 | 1 | 340.3 |
| 1175 | 1 | 439.26 | 1 | 340.3 |
| 1180 | 1 | 112.06 | 1 | 340.3 |
| 1181 | 1 | 242.05 | 1 | 340.3 |
| 1182 | 1 | 236.34 | 1 | 340.3 |
| 1184 | 1 | 416.35 | 1 | 419.8 |
| 1188 | 1 | 452.00 | 1 | 419.8 |
| 1191 | 1 | 396.57 | 1 | 419.8 |
| 1192 | 1 | 377.49 | 1 | 419.8 |
| 1193 | 1 | 415.43 | 1 | 419.8 |
| 1197 | 1 | 776.23 | 1 | 527.1 |
| 1199 | 1 | 383.07 | 1 | 527.1 |
| 1201 | 1 | 259.07 | 1 | 527.1 |
| 1204 | 1 | 675.53 | 1 | 527.1 |
| 1208 | 1 | 262.04 | 1 | 386.3 |
| 1211 | 1 | 200.33 | 1 | 386.3 |
| 1212 | 1 | 371.04 | 1 | 386.3 |
| 1215 | 1 | 570.77 | 1 | 386.3 |
| 1217 | 1 | 433.46 | 1 | 386.3 |
| 1222 | 1 | 830.90 | 1 | 445.3 |
| 1225 | 1 | 257.00 | 1 | 445.3 |
| 1227 | 1 | 250.20 | 1 | 445.3 |
| 1231 | 1 | 268.25 | 1 | 346.4 |
| 1232 | 1 | 407.97 | 1 | 346.4 |
| 1236 | 1 | 381.56 | 1 | 346.4 |
| 1239 | 1 | 343.00 | 1 | 346.4 |
| 1244 | 1 | 343.99 | 1 | 346.4 |
| 1249 | 1 | 297.27 | 1 | 346.4 |
| 1254 | 1 | 281.26 | 0 | 233.9 |
| 1258 | 1 | 256.92 | 0 | 233.9 |
| 1260 | 1 | 380.08 | 0 | 233.9 |
| 1263 | 1 | 185.74 | 0 | 233.9 |
| 1266 | 1 | 105.79 | 0 | 233.9 |
| 1272 | 1 | 85.18 | 1 | 442.3 |
| 1274 | 1 | 285.13 | 1 | 442.3 |
| 1278 | 1 | 282.25 | 1 | 442.3 |
| 1283 | 1 | 622.50 | 1 | 442.3 |
| 1288 | 1 | 856.20 | 1 | 442.3 |
| 1291 | 1 | 344.24 | 0 | 290.0 |
| 1294 | 1 | 402.75 | 0 | 290.0 |
| 1299 | 1 | 233.12 | 0 | 290.0 |
| 1304 | 1 | 79.20 | 0 | 290.0 |
| 1310 | 1 | 303.00 | 0 | 290.0 |
| 1314 | 1 | 529.55 | 1 | 637.7 |
| 1318 | 1 | 315.60 | 1 | 637.7 |
| 1322 | 1 | 334.90 | 1 | 637.7 |
| 1328 | 1 | 209.62 | 1 | 637.7 |
| 1331 | 1 | 532.26 | 1 | 637.7 |
| 1339 | 1 | 1267.11 | 1 | 637.7 |
| 1348 | 1 | 891.51 | 1 | 648.7 |
| 1353 | 1 | 184.79 | 1 | 648.7 |
| 1361 | 1 | 812.17 | 1 | 648.7 |
| 1364 | 1 | 139.67 | 1 | 648.7 |
| 1367 | 1 | 295.20 | 0 | 393.9 |
| 1372 | 1 | 569.67 | 0 | 393.9 |
| 1383 | 1 | 256.05 | 0 | 393.9 |
| 1386 | 1 | 175.96 | 0 | 393.9 |
| 1389 | 1 | 446.70 | 0 | 393.9 |
| 1393 | 1 | 102.69 | 0 | 393.9 |
| 1395 | 1 | 726.16 | 1 | 429.5 |
| 1399 | 1 | 443.82 | 1 | 429.5 |
| 1401 | 1 | 475.36 | 1 | 429.5 |
| 1405 | 1 | 254.42 | 1 | 429.5 |
| 1409 | 1 | 80.52 | 1 | 429.5 |
| 1411 | 1 | 894.26 | 1 | 944.4 |
| 1413 | 1 | 514.84 | 1 | 944.4 |
| 1416 | 1 | 1412.17 | 1 | 944.4 |
| 1420 | 1 | 974.53 | 1 | 944.4 |
| 1428 | 1 | 799.96 | 1 | 944.4 |
| 1433 | 1 | 244.15 | 0 | 448.7 |
| 1437 | 1 | 132.73 | 0 | 448.7 |
| 1442 | 1 | 188.33 | 0 | 448.7 |
| 1444 | 1 | 442.43 | 0 | 448.7 |
| 1449 | 1 | 947.68 | 0 | 448.7 |
| 1456 | 1 | 312.87 | 0 | 448.7 |
| 1460 | 1 | 332.09 | 1 | 462.0 |
| 1463 | 1 | 471.45 | 1 | 462.0 |
| 1467 | 1 | 221.38 | 1 | 462.0 |
| 1472 | 1 | 480.97 | 1 | 462.0 |
| 1478 | 1 | 606.00 | 1 | 462.0 |
| 1488 | 1 | 330.08 | 1 | 558.1 |
| 1491 | 1 | 680.34 | 1 | 558.1 |
| 1496 | 1 | 636.47 | 1 | 558.1 |
| 1505 | 1 | 346.98 | 1 | 558.1 |
| 1508 | 1 | 377.67 | 0 | 378.3 |
| 1511 | 1 | 298.31 | 0 | 378.3 |
| 1515 | 1 | 468.38 | 0 | 378.3 |
| 1519 | 1 | 393.58 | 0 | 378.3 |
| 1522 | 1 | 338.80 | 0 | 378.3 |
| 1526 | 1 | 402.56 | 0 | 378.3 |
| 1529 | 1 | 292.17 | 1 | 278.0 |
| 1532 | 1 | 245.64 | 1 | 278.0 |
| 1536 | 1 | 274.80 | 1 | 278.0 |
| 1538 | 1 | 324.62 | 1 | 278.0 |
| 1540 | 1 | 300.50 | 0 | 244.4 |
| 1544 | 1 | 335.06 | 0 | 244.4 |
| 1547 | 1 | 86.95 | 0 | 244.4 |
| 1549 | 1 | 143.80 | 0 | 244.4 |
| 1553 | 1 | 295.93 | 0 | 244.4 |
| 1556 | 1 | 241.27 | 0 | 244.4 |
| 1558 | 1 | 262.50 | 0 | 244.4 |
| 1562 | 1 | 429.52 | 1 | 482.0 |
| 1567 | 1 | 709.45 | 1 | 482.0 |
| 1569 | 1 | 769.80 | 1 | 482.0 |
| 1573 | 1 | 258.00 | 1 | 482.0 |
| 1579 | 1 | 160.42 | 0 | 181.3 |
| 1583 | 1 | 140.52 | 0 | 181.3 |
| 1586 | 1 | 173.08 | 0 | 181.3 |
| 1588 | 1 | 122.07 | 0 | 181.3 |
| 1590 | 1 | 130.44 | 0 | 181.3 |
| 1592 | 1 | 196.96 | 0 | 181.3 |
| 1594 | 1 | 283.86 | 0 | 181.3 |
| 1598 | 1 | 425.13 | 1 | 312.4 |
| 1602 | 1 | 222.33 | 1 | 312.4 |
| 1610 | 1 | 138.00 | 1 | 312.4 |
| 1611 | 1 | 258.23 | 1 | 312.4 |
| 1614 | 1 | 244.34 | 1 | 312.4 |
| 1617 | 1 | 537.50 | 0 | 381.3 |
| 1623 | 1 | 365.90 | 0 | 381.3 |
| 1624 | 1 | 237.00 | 0 | 381.3 |
| 1629 | 1 | 123.50 | 0 | 381.3 |
| 1630 | 1 | 395.68 | 0 | 381.3 |
| 1634 | 1 | 289.63 | 1 | 368.1 |
| 1638 | 1 | 537.50 | 1 | 368.1 |
| 1643 | 1 | 374.15 | 1 | 368.1 |
| 1645 | 1 | 299.92 | 1 | 368.1 |
| 1653 | 1 | 447.18 | 0 | 312.9 |
| 1657 | 1 | 116.44 | 0 | 312.9 |
| 1660 | 1 | 249.21 | 0 | 312.9 |
| 1666 | 1 | 234.33 | 0 | 312.9 |
| 1670 | 1 | 645.88 | 0 | 312.9 |
| 1672 | 1 | 489.68 | 0 | 312.9 |
| 1675 | 1 | 88.65 | 0 | 312.9 |
| 1677 | 1 | 230.93 | 1 | 325.8 |
| 1681 | 1 | 461.33 | 1 | 325.8 |
| 1685 | 1 | 216.83 | 1 | 325.8 |
| 1687 | 1 | 576.55 | 1 | 325.8 |
| 1691 | 1 | 197.07 | 1 | 325.8 |
| 1699 | 1 | 337.60 | 1 | 325.8 |
| 1706 | 1 | 117.21 | 0 | 254.7 |
| 1710 | 1 | 210.67 | 0 | 254.7 |
| 1713 | 1 | 243.97 | 0 | 254.7 |
| 1715 | 1 | 594.27 | 0 | 254.7 |
| 1719 | 1 | 164.00 | 0 | 254.7 |
| 1723 | 1 | 157.14 | 0 | 254.7 |
| 1726 | 1 | 435.10 | 1 | 408.7 |
| 1731 | 1 | 262.00 | 1 | 408.7 |
| 1736 | 1 | 269.64 | 1 | 408.7 |
| 1740 | 1 | 265.89 | 1 | 408.7 |
| 1745 | 1 | 748.02 | 1 | 408.7 |
| 1752 | 1 | 382.56 | 1 | 408.7 |
| 1758 | 1 | 234.66 | 1 | 408.7 |
| 1760 | 1 | 216.20 | 0 | 202.1 |
| 1762 | 1 | 87.01 | 0 | 202.1 |
| 1764 | 1 | 177.01 | 0 | 202.1 |
| 1767 | 1 | 150.99 | 0 | 202.1 |
| 1771 | 1 | 349.33 | 0 | 202.1 |
| 1772 | 1 | 257.83 | 0 | 202.1 |
| 1778 | 1 | 365.42 | 1 | 586.4 |
| 1782 | 1 | 179.25 | 1 | 586.4 |
| 1783 | 1 | 766.09 | 1 | 586.4 |
| 1792 | 1 | 612.54 | 1 | 586.4 |
| 1796 | 1 | 516.71 | 1 | 586.4 |
| 1800 | 1 | 434.67 | 1 | 586.4 |
| 1801 | 1 | 278.90 | 0 | 205.6 |
| 1805 | 1 | 174.58 | 0 | 205.6 |
| 1808 | 1 | 87.54 | 0 | 205.6 |
| 1809 | 1 | 252.57 | 0 | 205.6 |
| 1813 | 1 | 112.95 | 0 | 205.6 |
| 1815 | 1 | 177.51 | 0 | 205.6 |
| 1818 | 1 | 233.89 | 1 | 343.1 |
| 1819 | 1 | 421.09 | 1 | 343.1 |
| 1823 | 1 | 388.14 | 1 | 343.1 |
| 1829 | 1 | 297.87 | 1 | 343.1 |
| 1834 | 1 | 123.09 | 1 | 343.1 |
| 1837 | 1 | 425.79 | 1 | 343.1 |
| 1842 | 1 | 123.97 | 0 | 316.0 |
| 1844 | 1 | 528.80 | 0 | 316.0 |
| 1848 | 1 | 322.14 | 0 | 316.0 |
| 1852 | 1 | 228.28 | 0 | 316.0 |
| 1856 | 1 | 306.89 | 0 | 316.0 |
| 1862 | 1 | 298.64 | 0 | 316.0 |
| 1867 | 1 | 295.33 | 1 | 323.7 |
| 1871 | 1 | 346.25 | 1 | 323.7 |
| 1875 | 1 | 91.89 | 1 | 323.7 |
| 1876 | 1 | 54.68 | 1 | 323.7 |
| 1877 | 1 | 286.72 | 1 | 323.7 |
| 1882 | 1 | 337.09 | 1 | 323.7 |
| 1887 | 1 | 430.50 | 1 | 323.7 |
| 1893 | 1 | 174.64 | 0 | 179.7 |
| 1897 | 1 | 64.58 | 0 | 179.7 |
| 1899 | 1 | 361.17 | 0 | 179.7 |
| 1902 | 1 | 113.19 | 0 | 179.7 |
| 1914 | 1 | 305.55 | 0 | 179.7 |
| 1918 | 1 | 287.53 | 1 | 457.6 |
| 1922 | 1 | 1122.09 | 1 | 457.6 |
| 1924 | 1 | 416.34 | 1 | 457.6 |
| 1930 | 1 | 89.09 | 1 | 457.6 |
| 1932 | 1 | 524.63 | 1 | 457.6 |
| 1937 | 1 | 270.68 | 0 | 237.1 |
| 1943 | 1 | 258.34 | 0 | 237.1 |
| 1954 | 1 | 249.76 | 0 | 237.1 |
| 1958 | 1 | 127.70 | 0 | 237.1 |
| 1961 | 1 | 157.92 | 0 | 237.1 |
| 1963 | 1 | 706.28 | 1 | 491.5 |
| 1969 | 1 | 119.92 | 1 | 491.5 |
| 1971 | 1 | 317.23 | 1 | 491.5 |
| 1974 | 1 | 701.09 | 1 | 491.5 |
| 1980 | 1 | 198.17 | 1 | 491.5 |
| 1982 | 1 | 145.12 | 1 | 491.5 |
| 1984 | 1 | 68.24 | 0 | 228.0 |
| 1988 | 1 | 108.86 | 0 | 228.0 |
| 1991 | 1 | 147.13 | 0 | 228.0 |
| 1993 | 1 | 526.50 | 0 | 228.0 |
| 1996 | 1 | 323.93 | 0 | 228.0 |
| 2000 | 1 | 160.00 | 0 | 228.0 |
| 2004 | 1 | 303.32 | 0 | 228.0 |
| 2006 | 1 | 94.48 | 1 | 418.9 |
| 2007 | 1 | 192.51 | 1 | 418.9 |
| 2011 | 1 | 902.79 | 1 | 418.9 |
| 2017 | 1 | 126.78 | 1 | 418.9 |
| 2019 | 1 | 333.77 | 1 | 418.9 |
| 2024 | 1 | 198.09 | 1 | 418.9 |
| 2027 | 1 | 306.30 | 0 | 187.1 |
| 2030 | 1 | 172.70 | 0 | 187.1 |
| 2035 | 1 | 189.83 | 0 | 187.1 |
| 2039 | 1 | 105.99 | 0 | 187.1 |
| 2042 | 1 | 191.91 | 0 | 187.1 |
| 2045 | 1 | 153.37 | 0 | 187.1 |
| 2047 | 1 | 157.02 | 1 | 222.7 |
| 2051 | 1 | 94.54 | 1 | 222.7 |
| 2054 | 1 | 158.63 | 1 | 222.7 |
| 2057 | 1 | 367.00 | 1 | 222.7 |
| 2060 | 1 | 209.60 | 1 | 222.7 |
| 2063 | 1 | 371.21 | 1 | 222.7 |
| 2066 | 1 | 355.19 | 0 | 195.9 |
| 2069 | 1 | 90.92 | 0 | 195.9 |
| 2071 | 1 | 112.20 | 0 | 195.9 |
| 2073 | 1 | 95.12 | 0 | 195.9 |
| 2074 | 1 | 328.00 | 0 | 195.9 |
| 2076 | 1 | 107.76 | 0 | 195.9 |
| 2079 | 1 | 317.76 | 1 | 379.2 |
| 2081 | 1 | 641.38 | 1 | 379.2 |
| 2084 | 1 | 459.73 | 1 | 379.2 |
| 2087 | 1 | 237.09 | 1 | 379.2 |
| 2091 | 1 | 294.93 | 1 | 379.2 |
| 2095 | 1 | 404.02 | 0 | 300.2 |
| 2098 | 1 | 76.07 | 0 | 300.2 |
| 2103 | 1 | 149.65 | 0 | 300.2 |
| 2108 | 1 | 520.16 | 0 | 300.2 |
| 2115 | 1 | 107.78 | 0 | 300.2 |
| 2120 | 1 | 326.53 | 1 | 289.5 |
| 2124 | 1 | 227.56 | 1 | 289.5 |
| 2127 | 1 | 376.33 | 1 | 289.5 |
| 2129 | 1 | 304.31 | 1 | 289.5 |
| 2133 | 1 | 94.33 | 1 | 289.5 |
| 2134 | 1 | 132.19 | 0 | 226.2 |
| 2139 | 1 | 313.16 | 0 | 226.2 |
| 2140 | 1 | 117.66 | 0 | 226.2 |
| 2142 | 1 | 380.93 | 0 | 226.2 |
| 2145 | 1 | 277.31 | 0 | 226.2 |
| 2148 | 1 | 220.55 | 0 | 226.2 |
| 2151 | 1 | 447.99 | 1 | 446.1 |
| 2154 | 1 | 249.19 | 1 | 446.1 |
| 2157 | 1 | 338.90 | 1 | 446.1 |
| 2160 | 1 | 790.23 | 1 | 446.1 |
| 2164 | 1 | 289.25 | 1 | 446.1 |
| 2167 | 1 | 317.20 | 0 | 239.6 |
| 2173 | 1 | 169.03 | 0 | 239.6 |
| 2176 | 1 | 334.50 | 0 | 239.6 |
| 2185 | 1 | 241.51 | 0 | 239.6 |
| 2190 | 1 | 105.40 | 0 | 239.6 |
| 2196 | 1 | 135.06 | 0 | 239.6 |
| 2199 | 1 | 424.40 | 1 | 338.3 |
| 2203 | 1 | 526.03 | 1 | 338.3 |
| 2208 | 1 | 304.45 | 1 | 338.3 |
| 2211 | 1 | 234.08 | 1 | 338.3 |
| 2215 | 1 | 196.95 | 1 | 338.3 |
| 2216 | 1 | 257.20 | 1 | 338.3 |
| 2220 | 1 | 238.98 | 1 | 338.3 |
| 2223 | 1 | 90.53 | 0 | 120.4 |
| 2226 | 1 | 83.24 | 0 | 120.4 |
| 2233 | 1 | 184.50 | 0 | 120.4 |
| 2236 | 1 | 146.50 | 0 | 120.4 |
| 2244 | 1 | 112.26 | 0 | 120.4 |
| 2248 | 1 | 115.67 | 0 | 120.4 |
| 2252 | 1 | 113.16 | 1 | 642.6 |
| 2255 | 1 | 377.33 | 1 | 642.6 |
| 2258 | 1 | 169.75 | 1 | 642.6 |
| 2265 | 1 | 381.46 | 1 | 642.6 |
| 2267 | 1 | 1437.12 | 1 | 642.6 |
| 2272 | 1 | 296.91 | 1 | 642.6 |
| 2282 | 1 | 202.09 | 0 | 533.3 |
| 2285 | 1 | 568.06 | 0 | 533.3 |
| 2292 | 1 | 190.82 | 0 | 533.3 |
| 2296 | 1 | 365.37 | 0 | 533.3 |
| 2301 | 1 | 116.77 | 0 | 533.3 |
| 2303 | 1 | 1797.58 | 0 | 533.3 |
| 2306 | 1 | 1166.35 | 1 | 707.5 |
| 2313 | 1 | 362.25 | 1 | 707.5 |
| 2316 | 1 | 417.98 | 1 | 707.5 |
| 2320 | 1 | 667.99 | 1 | 707.5 |
| 2325 | 1 | 297.33 | 1 | 707.5 |
| 2327 | 1 | 151.10 | 0 | 216.5 |
| 2330 | 1 | 111.08 | 0 | 216.5 |
| 2331 | 1 | 256.29 | 0 | 216.5 |
| 2335 | 1 | 159.94 | 0 | 216.5 |
| 2343 | 1 | 326.43 | 0 | 216.5 |
| 2349 | 1 | 134.71 | 0 | 216.5 |
| 2355 | 1 | 301.62 | 0 | 216.5 |
| 2360 | 1 | 264.17 | 1 | 253.6 |
| 2361 | 1 | 229.07 | 1 | 253.6 |
| 2365 | 1 | 174.67 | 1 | 253.6 |
| 2366 | 1 | 340.25 | 1 | 253.6 |
| 2369 | 1 | 196.44 | 1 | 253.6 |
| 2373 | 1 | 276.20 | 1 | 253.6 |
| 2379 | 1 | 294.95 | 0 | 332.9 |
| 2382 | 1 | 193.67 | 0 | 332.9 |
| 2385 | 1 | 166.44 | 0 | 332.9 |
| 2389 | 1 | 418.02 | 0 | 332.9 |
| 2396 | 1 | 167.79 | 0 | 332.9 |
| 2398 | 1 | 619.35 | 0 | 332.9 |
| 2402 | 1 | 226.12 | 0 | 332.9 |
| 2404 | 1 | 807.67 | 1 | 523.6 |
| 2409 | 1 | 408.00 | 1 | 523.6 |
| 2411 | 1 | 525.15 | 1 | 523.6 |
| 2415 | 1 | 101.90 | 1 | 523.6 |
| 2416 | 1 | 136.73 | 1 | 523.6 |
| 2418 | 1 | 176.89 | 0 | 180.5 |
| 2422 | 1 | 244.16 | 0 | 180.5 |
| 2431 | 1 | 128.21 | 0 | 180.5 |
| 2432 | 1 | 163.48 | 0 | 180.5 |
| 2446 | 1 | 445.75 | 1 | 330.0 |
| 2449 | 1 | 173.43 | 1 | 330.0 |
| 2451 | 1 | 273.75 | 1 | 330.0 |
| 2454 | 1 | 431.55 | 1 | 330.0 |
| 2458 | 1 | 262.24 | 1 | 330.0 |
| 2462 | 1 | 856.47 | 0 | 542.9 |
| 2466 | 1 | 128.53 | 0 | 542.9 |
| 2470 | 1 | 459.77 | 0 | 542.9 |
| 2478 | 1 | 89.47 | 0 | 542.9 |
| 2479 | 1 | 124.84 | 0 | 542.9 |
| 2482 | 1 | 1005.74 | 0 | 542.9 |
| 2488 | 1 | 800.53 | 1 | 625.2 |
| 2496 | 1 | 434.22 | 1 | 625.2 |
| 2504 | 1 | 791.03 | 1 | 625.2 |
| 2511 | 1 | 133.67 | 1 | 625.2 |
| 2512 | 1 | 380.70 | 1 | 625.2 |
| 2514 | 1 | 836.26 | 1 | 625.2 |
| 2520 | 1 | 360.85 | 1 | 625.2 |
| 2525 | 1 | 313.97 | 0 | 276.7 |
| 2529 | 1 | 245.33 | 0 | 276.7 |
| 2535 | 1 | 321.27 | 0 | 276.7 |
| 2540 | 1 | 215.69 | 0 | 276.7 |
| 2543 | 1 | 293.89 | 1 | 647.7 |
| 2545 | 1 | 168.45 | 1 | 647.7 |
| 2546 | 1 | 497.00 | 1 | 647.7 |
| 2552 | 1 | 575.38 | 1 | 647.7 |
| 2556 | 1 | 945.34 | 1 | 647.7 |
| 2564 | 1 | 151.50 | 0 | 293.3 |
| 2566 | 1 | 462.25 | 0 | 293.3 |
| 2572 | 1 | 485.27 | 0 | 293.3 |
| 2575 | 1 | 143.98 | 0 | 293.3 |
| 2576 | 1 | 214.24 | 0 | 293.3 |
| 2582 | 1 | 122.45 | 0 | 293.3 |
| 2586 | 1 | 391.67 | 1 | 354.7 |
| 2590 | 1 | 388.67 | 1 | 354.7 |
| 2594 | 1 | 474.78 | 1 | 354.7 |
| 2597 | 1 | 309.80 | 1 | 354.7 |
| 2603 | 1 | 229.54 | 1 | 354.7 |

Agregando la variable estrato y así poder incorporar las zonas a la matriz de coeficientes,

(Coef\_Estimado <- inner\_join( coef(mod\_Pen\_Aleatorio2)$Stratum %>%  
 add\_rownames(var = "Stratum"),  
 encuesta\_plot %>% dplyr::select(Stratum, Zone) %>% distinct() ))

| Stratum | (Intercept) | Expenditure | ZoneUrban | mu | Zone |
| --- | --- | --- | --- | --- | --- |
| idStrt002 | 46.15 | 1.594 | 30.46 | -0.1138 | Urban |
| idStrt010 | 84.82 | 1.980 | 152.75 | -0.6501 | Urban |
| idStrt015 | 24.73 | 1.746 | -159.02 | 0.0095 | Rural |
| idStrt017 | 50.91 | 1.579 | 35.45 | -0.1201 | Rural |
| idStrt022 | 40.66 | 1.133 | 25.13 | 0.2744 | Urban |
| idStrt028 | 45.78 | 1.572 | -82.04 | 0.0126 | Urban |

Ahora bien, se agrega la media calculada para los estratos a la matriz de coeficientes:

(Coef\_Estimado<- Coef\_Estimado %>% inner\_join(media\_estrato, by = "Stratum"))

| Stratum | (Intercept) | Expenditure | ZoneUrban | mu.x | Zone | mu.y |
| --- | --- | --- | --- | --- | --- | --- |
| idStrt002 | 46.15 | 1.594 | 30.46 | -0.1138 | Urban | 286.2 |
| idStrt010 | 84.82 | 1.980 | 152.75 | -0.6501 | Urban | 255.8 |
| idStrt015 | 24.73 | 1.746 | -159.02 | 0.0095 | Rural | 357.0 |
| idStrt017 | 50.91 | 1.579 | 35.45 | -0.1201 | Rural | 244.8 |
| idStrt022 | 40.66 | 1.133 | 25.13 | 0.2744 | Urban | 524.0 |
| idStrt028 | 45.78 | 1.572 | -82.04 | 0.0126 | Urban | 337.1 |

Por último, el modelo para el estrato *idStrt002* viene dado por:

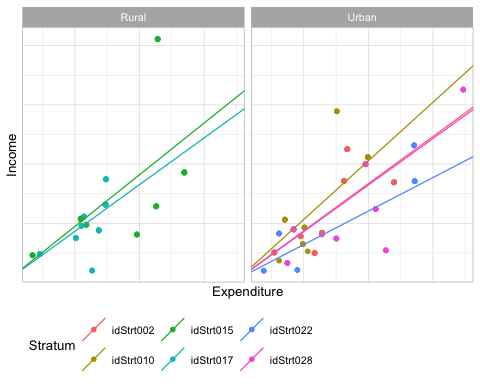
Escribiendo el anterior modelo en R se tiene:

(Coef\_Estimado %>% dplyr::mutate(B0 = ifelse( Zone == "Urban", `(Intercept)` + mu.y \* mu.x + ZoneUrban,`(Intercept)` + mu.y \* mu.x)) %>%  
 dplyr::select(Stratum, Zone, B0, Expenditure))

| Stratum | Zone | B0 | Expenditure |
| --- | --- | --- | --- |
| idStrt002 | Urban | 44.06 | 1.594 |
| idStrt010 | Urban | 71.28 | 1.980 |
| idStrt015 | Rural | 28.13 | 1.746 |
| idStrt017 | Rural | 21.51 | 1.579 |
| idStrt022 | Urban | 209.57 | 1.133 |
| idStrt028 | Urban | -32.02 | 1.572 |

Gráficamente,

ggplot(data = encuesta\_plot,  
 aes(y = Income, x = Expenditure,  
 colour = Stratum)) +  
 geom\_jitter() +  
 theme(legend.position = "none",  
 plot.title = element\_text(hjust = 0.5)) +  
 facet\_grid( ~ Zone) +  
 geom\_abline(  
 data = Coef\_Estimado,  
 mapping = aes(  
 slope = Expenditure,  
 intercept = B0,  
 colour = Stratum  
 )  
 ) +  
 theme\_cepal()



## 9.3 Introducción a los modelos logístico multinivel.

Los modelos logísticos multinivel son una extensión de los modelos logísticos simples, que se utilizan para predecir la probabilidad de un resultado binario en función de una o varias variables explicativas. Sin embargo, en muchas situaciones, los datos se recogen de individuos que están agrupados en diferentes niveles o unidades de análisis, como escuelas, ciudades o países. En estos casos, los modelos logísticos simples pueden no ser suficientes para capturar la estructura jerárquica de los datos y la variación en las respuestas entre los diferentes grupos.

Los modelos logísticos multinivel resuelven este problema al permitir que los coeficientes del modelo varíen a través de los diferentes niveles de análisis. En otras palabras, se permite que la relación entre las variables predictoras y la respuesta varíe en función del grupo al que pertenece cada individuo.

Además, los modelos logísticos multinivel permiten incluir tanto variables a nivel individual como variables a nivel de grupo, lo que aumenta la precisión de las estimaciones y la capacidad de explicar la variabilidad en las respuestas. También permiten estimar la varianza en las respuestas entre los diferentes grupos, lo que es útil para identificar las fuentes de variabilidad y para comparar la variabilidad entre grupos.

En general, los modelos logísticos multinivel son una herramienta poderosa para analizar datos de respuestas binarias en contextos jerárquicos, y son ampliamente utilizados en muchas áreas de investigación, como la educación, la salud, las ciencias sociales y la psicología.

Sea la variable si el individuo en el estrato está por encima de la línea de pobreza y en caso contrario, la variable se puede modelar mediante el modelo logístico:

ó

donde,

.

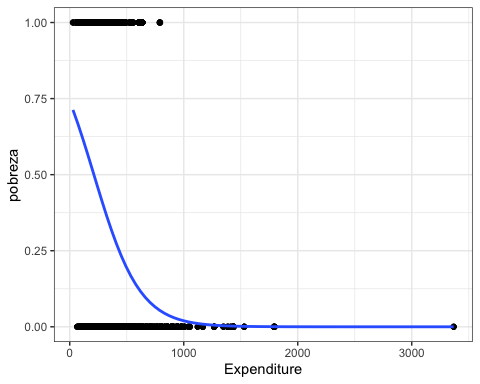
A modo de ejemplo, se ajustará un modelo logístico para la variable pobreza. Inicialmente, se crea la variable dicotómica en la base como se muestra a continuación:

encuesta\_plot <- encuesta %>%  
 dplyr::select(Stratum,Expenditure) %>% unique() %>%  
 group\_by(Stratum) %>%   
 summarise(sd = sd(Expenditure)) %>%   
 arrange(desc(sd)) %>% dplyr::select(-sd) %>%   
 slice(1:20L) %>%  
 inner\_join(encuesta) %>%   
 dplyr::select(Poverty, Expenditure, Stratum,   
 Sex, Region, Zone)   
encuesta\_plot %>% slice(1:15L)

| Poverty | Expenditure | Stratum | Sex | Region | Zone |
| --- | --- | --- | --- | --- | --- |
| NotPoor | 3367.5 | idStrt039 | Male | Sur | Urban |
| NotPoor | 3367.5 | idStrt039 | Female | Sur | Urban |
| NotPoor | 3367.5 | idStrt039 | Male | Sur | Urban |
| NotPoor | 312.1 | idStrt039 | Female | Sur | Urban |
| NotPoor | 312.1 | idStrt039 | Female | Sur | Urban |
| NotPoor | 312.1 | idStrt039 | Female | Sur | Urban |
| NotPoor | 312.1 | idStrt039 | Male | Sur | Urban |
| NotPoor | 226.5 | idStrt039 | Male | Sur | Urban |
| NotPoor | 226.5 | idStrt039 | Female | Sur | Urban |
| NotPoor | 616.3 | idStrt047 | Female | Sur | Urban |
| NotPoor | 616.3 | idStrt047 | Female | Sur | Urban |
| NotPoor | 616.3 | idStrt047 | Female | Sur | Urban |
| NotPoor | 1385.7 | idStrt047 | Male | Sur | Urban |
| NotPoor | 1385.7 | idStrt047 | Female | Sur | Urban |
| NotPoor | 1385.7 | idStrt047 | Female | Sur | Urban |

Para poder observar la distribución la distribución de la variable pobreza, se presenta el siguiente gráfico:

encuesta <- encuesta %>% mutate( pobreza = ifelse(Poverty != "NotPoor", 1, 0))  
encuesta\_plot %<>% mutate(pobreza = ifelse(Poverty != "NotPoor", 1, 0))  
  
ggplot(data = encuesta, aes(y = pobreza, x = Expenditure)) +   
 geom\_point() + geom\_smooth(formula = y~x, method = "glm",se=FALSE,   
 method.args = list(family=binomial(link = "logit"))) + theme\_bw()

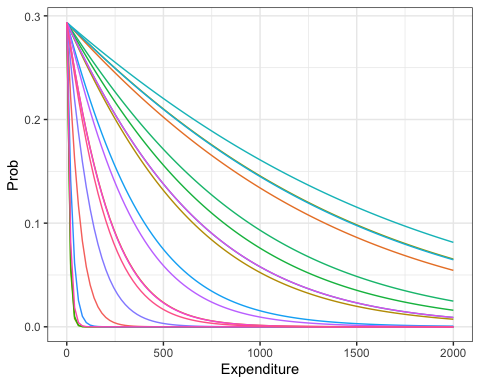
 El ajuste del modelo logístico se realiza con la función glm y la función link “logit”. Una vez se ajusta el modelo, se extraen los coeficientes del modelo y así poder calcular las probabilidades, como sigue a continuación:

auxLogit <- function(x,b0,b1){  
 1/(1+exp(-(b0+b1\*x)))  
}  
  
B0 = coef(glm(pobreza~1,data = encuesta\_plot,  
 family=binomial(link = "logit")))  
  
(coef\_Mod <- encuesta\_plot %>% group\_by(Stratum) %>%   
 summarise(B1 = coef(glm(pobreza ~ -1 + Expenditure,  
 family=binomial(link = "logit")))) %>%   
mutate(B0 = B0)) %>% slice(1:6L)

| Stratum | B1 | B0 |
| --- | --- | --- |
| idStrt007 | -0.0189 | -0.8782 |
| idStrt020 | -0.0010 | -0.8782 |
| idStrt022 | -0.0057 | -0.8782 |
| idStrt024 | -0.0020 | -0.8782 |
| idStrt036 | -0.0009 | -0.8782 |
| idStrt039 | -0.0976 | -0.8782 |

A continuación, se grafican los diferentes modelos logísticos ajustados para cada uno de los estratos observándose que, hay una variación importante entre los estratos:

pred\_logit <- coef\_Mod %>% mutate(Expenditure = list(seq(0,2000, length =100))) %>%   
 tidyr::unnest\_legacy()  
  
pred\_logit %<>% mutate(Prob = auxLogit(Expenditure,B0,B1))   
   
ggplot(data = pred\_logit, aes(y = Prob, x = Expenditure, colour = Stratum)) +   
 geom\_line() + theme\_bw() + theme(legend.position = "none")



Un modelo logístico básico o nulo se escribe de la siguiente manera:

Donde los componentes son los siguientes:

* .
* El intercepto en el estrato .
* El residual de la persona en el estrato .
* El intercepto en general.
* Efecto aleatorio para el intercepto.

con, y .

En este caso, la correlación intra clásica está dada por:

En R el modelo nulo se ajusta de la siguiente manera:

library(lme4)   
mod\_logist\_null <- glmer( pobreza ~ ( 1 | Stratum ),  
 data = encuesta,   
 weights = wk2,  
 family = binomial(link = "logit") )  
  
  
coef( mod\_logist\_null )$Stratum %>% slice(1:12)

|  | (Intercept) |
| --- | --- |
| idStrt001 | -0.8334 |
| idStrt002 | -0.0133 |
| idStrt003 | -2.6023 |
| idStrt004 | -2.7770 |
| idStrt005 | -1.0268 |
| idStrt006 | 1.0100 |
| idStrt007 | -1.0134 |
| idStrt008 | 0.2035 |
| idStrt009 | 2.1966 |
| idStrt010 | -0.5948 |
| idStrt011 | -1.2986 |
| idStrt012 | 0.2825 |

Las estadísticas resumen del modelo se presentan a continuación:

library(sjstats)  
mod\_logist\_null

## Generalized linear mixed model fit by maximum likelihood (Laplace  
## Approximation) [glmerMod]  
## Family: binomial ( logit )  
## Formula: pobreza ~ (1 | Stratum)  
## Data: encuesta  
## Weights: wk2  
## AIC BIC logLik -2\*log(L) df.resid   
## 2966 2978 -1481 2962 2603   
## Random effects:  
## Groups Name Std.Dev.  
## Stratum (Intercept) 1.29   
## Number of obs: 2605, groups: Stratum, 119  
## Fixed Effects:  
## (Intercept)   
## -0.802

A continuación, se presenta la correlación intraclase, las predicciones del modelo y las variables observadas:

performance::icc(mod\_logist\_null)

| ICC\_adjusted | ICC\_unadjusted | optional |
| --- | --- | --- |
| 0.3342 | 0.3342 | FALSE |

(tab\_pred <- data.frame(Pred = predict(mod\_logist\_null, type = "response"),   
 pobreza = encuesta$pobreza,   
 Stratum = encuesta$Stratum)) %>% distinct() %>%   
 slice(1:6L) # Son las pendientes aleatorias

|  | Pred | pobreza | Stratum |
| --- | --- | --- | --- |
| 1 | 0.3029 | 0 | idStrt001 |
| 10 | 0.3029 | 1 | idStrt001 |
| 28 | 0.4967 | 1 | idStrt002 |
| 36 | 0.4967 | 0 | idStrt002 |
| 61 | 0.0690 | 0 | idStrt003 |
| 84 | 0.0586 | 0 | idStrt004 |

Para efecto de verificar qué tan buena fueron las predicciones del modelo, se estiman el porcentaje de pobreza de la variable observada y de las predicciones de modelo utilizando la función weighted.mean. Se logra observar que son muy similares las estimaciones:

weighted.mean(encuesta$pobreza, encuesta$wk2)

## [1] 0.3859

weighted.mean(tab\_pred$Pred, encuesta$wk2)

## [1] 0.385

*Modelo con intercepto aleatoria*

EL modelo se define de la siguiente manera:

Donde,

Siguiendo las ideas de la sección anterior, el ajuste del modelo en R se realiza de la siguiente manera:

mod\_logit\_Int\_Aleatorio <- glmer(pobreza ~ Expenditure + (1 | Stratum),  
 data = encuesta, family = binomial(link = "logit"),weights = wk2)  
  
performance::icc(mod\_logit\_Int\_Aleatorio)

| ICC\_adjusted | ICC\_unadjusted | optional |
| --- | --- | --- |
| 0.3151 | 0.1867 | FALSE |

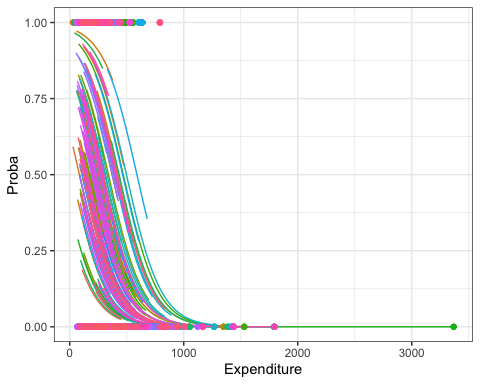
Los coeficientes estimados son:

coef(mod\_logit\_Int\_Aleatorio)$Stratum %>% slice(1:10L)

|  | (Intercept) | Expenditure |
| --- | --- | --- |
| idStrt001 | 0.9889 | -0.0066 |
| idStrt002 | 1.8837 | -0.0066 |
| idStrt003 | -0.7463 | -0.0066 |
| idStrt004 | -0.1484 | -0.0066 |
| idStrt005 | 1.7155 | -0.0066 |
| idStrt006 | 3.2456 | -0.0066 |
| idStrt007 | 0.5601 | -0.0066 |
| idStrt008 | 1.6848 | -0.0066 |
| idStrt009 | 3.9332 | -0.0066 |
| idStrt010 | 1.1207 | -0.0066 |

Gráficamente, los modelos ajustados se muestran a continuación:

dat\_pred <- encuesta %>% group\_by(Stratum) %>%   
 summarise(Expenditure = list(seq(min(Expenditure),   
 max(Expenditure), len = 100))) %>% tidyr::unnest\_legacy()  
  
dat\_pred <- mutate(dat\_pred,Proba = predict(mod\_logit\_Int\_Aleatorio,   
 newdata = dat\_pred , type = "response"))  
  
ggplot(data = dat\_pred,  
 aes(y = Proba, x = Expenditure,  
 colour = Stratum)) +  
 geom\_line()+ theme\_bw() +  
 geom\_point(data = encuesta, aes(y = pobreza, x = Expenditure))+  
 theme(legend.position = "none",  
 plot.title = element\_text(hjust = 0.5))



Las predicciones del modelo se presentan a continuación:

(tab\_pred <- data.frame(Pred = predict(mod\_logit\_Int\_Aleatorio,  
 type = "response"),   
 pobreza = encuesta$pobreza,   
 Stratum = encuesta$Stratum,   
 wk2 = encuesta$wk2)) %>% distinct() %>%   
 slice(1:6L)

| Pred | pobreza | Stratum | wk2 |
| --- | --- | --- | --- |
| 0.2149 | 0 | idStrt001 | 0.7770 |
| 0.2149 | 0 | idStrt001 | 0.7501 |
| 0.2149 | 0 | idStrt001 | 0.7463 |
| 0.2149 | 0 | idStrt001 | 0.7717 |
| 0.2149 | 0 | idStrt001 | 0.7438 |
| 0.1682 | 0 | idStrt001 | 0.7507 |

Como se indicó anteriormente, para verificar la calidad del modelo se realizan las estimaciones de las predicciones y de las variables observadas, teniendo estimaciones similares:

tab\_pred %>%   
 summarise(Pred = weighted.mean(Pred, wk2),   
 pobreza = weighted.mean(pobreza,wk2))

| Pred | pobreza |
| --- | --- |
| 0.3855 | 0.3859 |

*Modelo con intercepto y pendiente aleatoria*

Los modelos logísticos con intercepto y pendiente aleatoria son un tipo de modelo logístico multinivel que permiten que tanto el intercepto como la pendiente varíen aleatoriamente entre los diferentes grupos de observación.

En los modelos logísticos básicos, la relación entre las variables predictoras y la variable de respuesta se modela mediante una función logística, donde la respuesta es la probabilidad de que el resultado binario ocurra. En los modelos con intercepto y pendiente aleatoria, la función logística se ajusta para cada grupo de observación, y tanto el intercepto como la pendiente son variables aleatorias que varían de un grupo a otro. Esto permite que los coeficientes del modelo, que representan la relación entre las variables predictoras y la respuesta, varíen según el grupo de observación.

La incorporación de coeficientes aleatorios en los modelos logísticos multinivel permite capturar la heterogeneidad en la relación entre las variables predictoras y la respuesta en diferentes grupos de observación, y mejora la precisión de las estimaciones. Además, estos modelos también permiten la inclusión de variables a nivel individual y a nivel de grupo, lo que permite una mejor comprensión de la estructura jerárquica de los datos. El modelo se define de la siguiente manera:

con,

donde,

En R, el ajuste se hace de la siguiente manera:

mod\_logit\_Pen\_Aleatorio <- glmer(pobreza ~ Expenditure + (1 + Expenditure| Stratum),  
 data = encuesta, weights = wk2, binomial(link = "logit"))  
  
performance::icc(mod\_logit\_Pen\_Aleatorio)

| ICC\_adjusted | ICC\_unadjusted | optional |
| --- | --- | --- |
| 0.8859 | 0.6534 | FALSE |

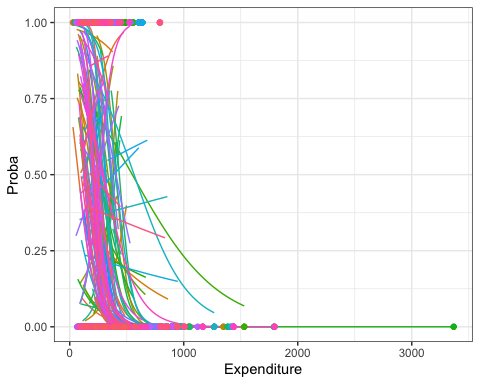
Los coeficientes del modelo son:

coef(mod\_logit\_Pen\_Aleatorio)$Stratum %>% slice(1:10L)

|  | (Intercept) | Expenditure |
| --- | --- | --- |
| idStrt001 | 5.245 | -0.0272 |
| idStrt002 | 11.061 | -0.0394 |
| idStrt003 | -1.613 | -0.0060 |
| idStrt004 | 1.657 | -0.0153 |
| idStrt005 | 9.056 | -0.0289 |
| idStrt006 | -1.355 | 0.0100 |
| idStrt007 | 1.035 | -0.0136 |
| idStrt008 | 1.473 | -0.0056 |
| idStrt009 | 4.050 | -0.0048 |
| idStrt010 | 4.311 | -0.0214 |

Gráficamente el ajuste de los modelo se muestra a continuación:

dat\_pred <- encuesta %>% group\_by(Stratum) %>%   
 summarise(Expenditure = list(seq(min(Expenditure), max(Expenditure), len = 100))) %>%   
 tidyr::unnest\_legacy()  
  
dat\_pred <- mutate(dat\_pred,Proba = predict(mod\_logit\_Pen\_Aleatorio, newdata = dat\_pred , type = "response"))  
  
ggplot(data = dat\_pred,  
 aes(y = Proba, x = Expenditure,  
 colour = Stratum)) +  
 geom\_line()+ theme\_bw() +  
 geom\_point(data = encuesta, aes(y = pobreza, x = Expenditure))+  
 theme(legend.position = "none",  
 plot.title = element\_text(hjust = 0.5))



Las predicciones se muestran a continuación:

(tab\_pred <- data.frame(  
 Pred = predict(mod\_logit\_Pen\_Aleatorio,  
 type = "response"),   
 pobreza = encuesta$pobreza,   
 Stratum = encuesta$Stratum,   
 wk2 = encuesta$wk2)) %>% distinct() %>%   
 slice(1:6L)

| Pred | pobreza | Stratum | wk2 |
| --- | --- | --- | --- |
| 0.0154 | 0 | idStrt001 | 0.7770 |
| 0.0154 | 0 | idStrt001 | 0.7501 |
| 0.0154 | 0 | idStrt001 | 0.7463 |
| 0.0154 | 0 | idStrt001 | 0.7717 |
| 0.0154 | 0 | idStrt001 | 0.7438 |
| 0.0045 | 0 | idStrt001 | 0.7507 |

La calidad de la predicción del modelo es muy buena como se muestra a continuación:

tab\_pred %>%   
 summarise(Pred = weighted.mean(Pred, wk2),   
 pobreza = weighted.mean(pobreza,wk2))

| Pred | pobreza |
| --- | --- |
| 0.3845 | 0.3859 |

Por otro lado, se ajusta un modelo agregando ahora la variable zona. La idea es entonces medir el porcentaje de pobreza discriminando por zona. El modelo es el siguiente:

Donde,

con,

y,

donde es el gasto medio en el estrato .

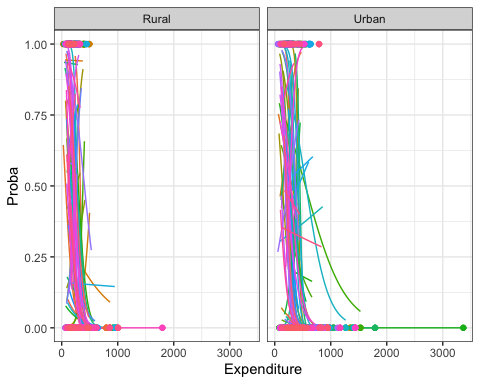
El ajuste del modelo es el siguiente:

mod\_logit\_Pen\_Aleatorio2 <- glmer(  
 pobreza ~ 1 + Expenditure + Zone + mu +  
 (1 + Expenditure + Zone + mu | Stratum ),  
 data = encuesta, weights = wk2,   
 binomial(link = "logit"))  
performance::icc(mod\_logit\_Pen\_Aleatorio2)

## [1] NA

Se grafican los modelos ajustados anteriormente:

dat\_pred <- encuesta %>% group\_by(Stratum, Zone, mu) %>%   
 summarise(  
 Expenditure = list(seq(min(Expenditure),   
 max(Expenditure), len = 100))) %>%   
 tidyr::unnest\_legacy()  
  
dat\_pred$Proba = predict(mod\_logit\_Pen\_Aleatorio2,   
 newdata = dat\_pred , type = "response")  
  
ggplot(data = dat\_pred,  
 aes(y = Proba, x = Expenditure,  
 colour = Stratum)) +  
 geom\_line()+ theme\_bw() +facet\_grid(.~Zone)+  
 geom\_point(data = encuesta, aes(y = pobreza, x = Expenditure))+  
 theme(legend.position = "none",  
 plot.title = element\_text(hjust = 0.5))

 Se logra observar que, hay una variación importante en el ajuste de los modelos para cada zona. Ahora bien, las predicciones del porcentaje de pobreza por zona se calculan a continuación:

(tab\_pred <- data.frame(  
 Pred = predict(mod\_logit\_Pen\_Aleatorio2,   
 type = "response"),   
 pobreza = encuesta$pobreza,   
 Stratum = encuesta$Stratum,  
 Zone = encuesta$Zone,  
 wk2 = encuesta$wk2)) %>% distinct() %>%   
 slice(1:6L)

| Pred | pobreza | Stratum | Zone | wk2 |
| --- | --- | --- | --- | --- |
| 0.0145 | 0 | idStrt001 | Rural | 0.7770 |
| 0.0145 | 0 | idStrt001 | Rural | 0.7501 |
| 0.0145 | 0 | idStrt001 | Rural | 0.7463 |
| 0.0145 | 0 | idStrt001 | Rural | 0.7717 |
| 0.0145 | 0 | idStrt001 | Rural | 0.7438 |
| 0.0041 | 0 | idStrt001 | Rural | 0.7507 |

Por último, se verifica la calidad de las predicciones, obteniendo, como en los modelos anteriores, unas predicciones de buena calidad haciendo las comparaciones con las estimaciones de la variable observada para cada una de las zonas.

tab\_pred %>% group\_by(Zone) %>%   
 summarise(Pred = weighted.mean(Pred, wk2),   
 pobreza = weighted.mean(pobreza,wk2))

| Zone | Pred | pobreza |
| --- | --- | --- |
| Rural | 0.4366 | 0.4298 |
| Urban | 0.3366 | 0.3437 |

# 10 Imputación múltiple en encuestas de hogares

La no respuesta en encuestas de hogares es un fenómenos que desde siempre ha sucedido, más ahora después de la pandemia del Covid -19. Esto sucede por muchas razones, por ejemplo, la longitud de los cuestionarios en las encuestas, el no conocer algunas características particulares del hogar, la renuencia por entregar información sensible, entre otras. Si la no respuesta en algún indicador del estudio es muy alto, puede poner en riesgo la calidad de las estimaciones que se obtienen utilizando los estimadores clásicos.

Debido a este problema, se ha avanzado en la literatura especializada metodologías que permiten atacar este problema, el cual es una realidad en todas las encuestas de hogares en los distintos países. 427

En este sentido, alguno de los avances en la teoría de los métodos en el contexto de muestras comlejas son: imputación múltiple (MI; Van Buuren, 2012; Carpenter y Kenward, 2013; Berglund y Heeringa, 2014; Stata Corp, 2015; Raghunathan, 2016). También se encuentra en la literatura métodos como el de imputación fraccional (FI; Kim y Fuller, 2004; Kim y Shao, 2014), el Bootstrap Bayesiano de Población Finita (Zhou et al., 2016a,b), y métodos de máxima verosimilitud como los de Chambers et al., (2012).

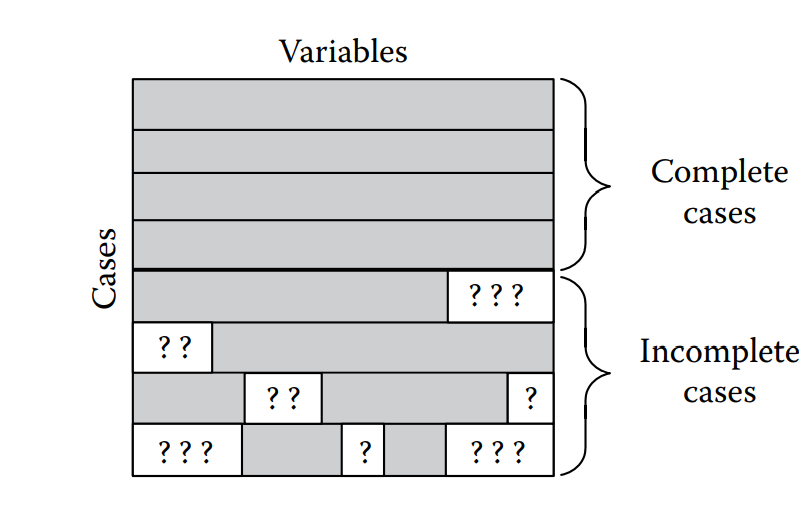
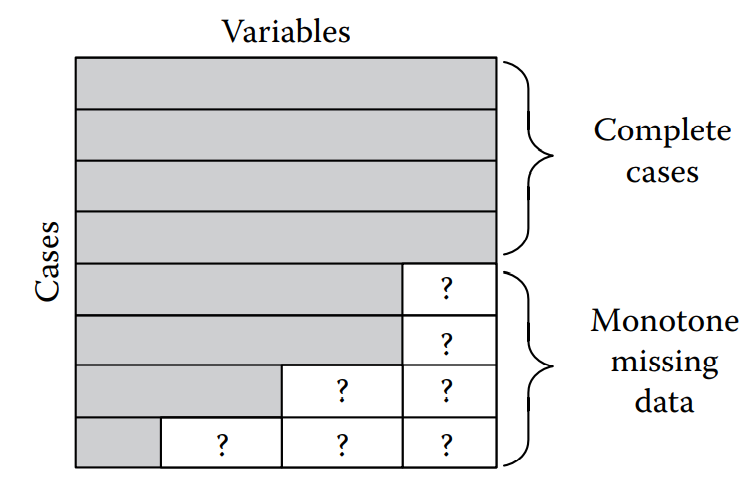
El objetivo principal de este capítulo es abordar el problema de la falta de datos en encuestas de hogares, revisar las posibles causas, el impacto que tiene en la estimación de los indicadores, y mostrar algunas soluciones a la falta de datos en las encuesta.

Siguiendo las ideas anteriores, sea una matriz completa (sin valores perdidos) de tal forma que es el valor de la variable con e con . Adicionalmente, se define una matriz indicadoradonde si el valor de es un dato perdido y si está presente.

Ahora bie, note que la matriz describe el patrón de missing (datos faltantes), y su media marginal de columna puede ser interpretada como la probabilidad de que sea missing. A continuación, se describen alguna de las particularidades de la matriz :

* La matriz presenta un comportamiento completamente al azar (MCAR): si la probabilidad de respuesta es independiente de las variables observadas y de las no observadas completamente. El mecanismo de pérdida es ignorable tanto para inferencias basadas en muestreo como en máxima verosimilitud.
* Los valores de la matriz son al azar (MAR): si la probabilidad de respuesta es independiente de las variables no observadas completamente y no de las observadas. El mecanismo de pérdida es ignorable para inferencias basadas en máxima verosimilitud.
* Los datos no están perdidos al azar (MNAR): si la probabilidad de respuesta no es independiente de las variables no observadas completamente y posiblemente, también, de las observadas. El mecanismo de pérdida es no ignorable.

En las dos figuras siguientes, se ilustran los casos de observaciones perdidas de manera aleatoria y con un patrón identificado:

Como se ha venido trabajando en los capítulos anteriores, primero carguemos la base de datos con la muestra seleccionada y con el fin de poder ejemplificar el tratamiento de datos faltantes, se incluiran manualmente “valores perdidos”. En este sentido, la lectura de la base se hará a continuación:

encuesta <- readRDS("Data/encuesta.rds")

Se filtran encuestados mayores a 15 años y se calcula la proporción de la población desempleada, inactiva y empleada antes de generar los valores faltantes. Pero antes de eso, se cargan todas las librerías que se utilizarán en este capítulo:

knitr::opts\_chunk$set(cache = FALSE, warning = FALSE, message = FALSE, error = FALSE)  
#options(digits = 4)  
options(tinytex.verbose = TRUE)  
library (survey)  
library(srvyr)  
library(convey)  
library(TeachingSampling)  
library(printr)  
library(stargazer)  
library(broom)  
library(jtools)  
library(modelsummary)  
library(patchwork)  
library(ggplot2)

Cálculo de las proprociones de personas desempleadas, inactivas y empleados:

encuesta<- encuesta |> filter(Age >= 15)  
(tab\_antes <- prop.table(table(encuesta$Employment)))

| Unemployed | Inactive | Employed |
| --- | --- | --- |
| 0.0409792 | 0.373603 | 0.5854178 |

También se calcula el promedio de ingresos en al muestra:

(med\_antes <- mean(encuesta$Income, na.rm = TRUE))

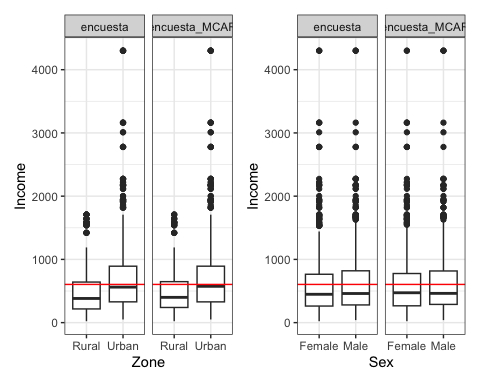
## [1] 604.2494

Luego de los conteos anteriores, se genera un 20% de valores faltantes siguiendo un esquema MCAR como sigue:

set.seed(1234)  
encuesta\_MCAR <- sample\_frac(encuesta, 0.8 )  
dat\_plot <- bind\_rows(  
 list(encuesta\_MCAR = encuesta\_MCAR,   
 encuesta = encuesta), .id = "Caso" )

Ahora bien, para poder ver el efecto de la inclusión de datos faltantes de manera gráfica por zona y sexo para la variable ingreso, se realizan las siguientes gráficas:

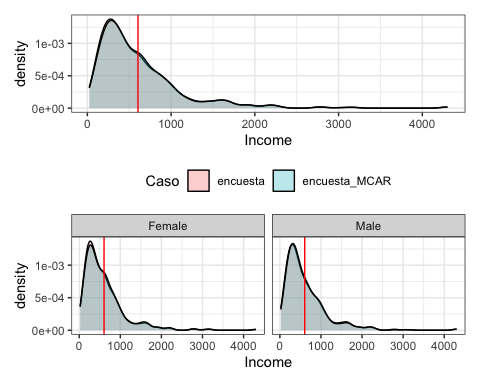
p1 <- ggplot(dat\_plot, aes(x=Zone, y = Income)) +   
 geom\_boxplot() + facet\_grid(.~Caso) + theme\_bw()+  
 geom\_hline(yintercept = mean(encuesta$Income),   
 col = "red")  
  
p2 <- ggplot(dat\_plot, aes(x=Sex, y = Income)) +   
 geom\_boxplot() + facet\_grid(.~Caso) +theme\_bw()+  
 geom\_hline(yintercept = mean(encuesta$Income),   
 col = "red")  
library(patchwork)  
p1|p2



Como se puede observar en las gráficas anteriores, la distribución de los ingresos por Zona y Sexo se mantiene similar con o sin presencia de la no respuesta. Esto se debe a que la no respuesta que se incluyó no depende de la variable de estudio.

Ahora bien, analizando la variable de interés se tiene que tampoco hay cambios distribucionales notables entre las distribuciones con y sin datos faltantes por sexo, como se puede observar a continuación:

p1 <- ggplot(dat\_plot, aes(x = Income, fill = Caso)) +   
 geom\_density(alpha = 0.3) + theme\_bw() +  
 theme(legend.position = "bottom") +  
 geom\_vline(xintercept = mean(encuesta$Income),   
 col = "red")  
  
p2 <- ggplot(dat\_plot, aes(x = Income, fill = Caso)) +   
 geom\_density(alpha = 0.3) + facet\_grid(.~Sex) +   
 theme\_bw()+   
 geom\_vline(xintercept = mean(encuesta$Income),   
 col = "red") +  
 theme(legend.position = "none")  
(p1/p2)



Si graficamos ahora la varibale gastos, se tienen los mismos resultados que para ingresos.

p1 <- ggplot(dat\_plot, aes(x = Expenditure, fill = Caso)) +   
 geom\_density(alpha = 0.3) + theme\_bw() +  
 theme(legend.position = "bottom") +  
 geom\_vline(xintercept = mean(encuesta$Expenditure),   
 col = "red")  
  
p2 <- ggplot(dat\_plot, aes(x = Expenditure, fill = Caso)) +   
 geom\_density(alpha = 0.3) + facet\_grid(.~Sex) +   
 theme\_bw()+   
 geom\_vline(xintercept = mean(encuesta$Expenditure),   
 col = "red") +  
 theme(legend.position = "none")  
(p1/p2)



Por otro lado, simulemos ahora una pérdida de información MAR como sigue:

library(TeachingSampling)  
set.seed(1234)  
temp\_estrato <- paste0(encuesta$Zone, encuesta$Sex)   
table(temp\_estrato)

| RuralFemale | RuralMale | UrbanFemale | UrbanMale |
| --- | --- | --- | --- |
| 481 | 428 | 531 | 439 |

sel <- S.STSI(S = temp\_estrato,   
 Nh = c(481,428,531,439),  
 nh = c(20, 380, 20,280))  
encuesta\_MAR <- encuesta[-sel,]  
dat\_plot2 <- bind\_rows(  
 list(encuesta\_MAR = encuesta\_MAR,  
 encuesta = encuesta), .id = "Caso" )

El código anterior utiliza la librería *TeachingSampling* para realizar un muestreo aleatorio estratificado. Primero, se establece la semilla aleatoria en 1234 para asegurarse de que los resultados sean reproducibles. A continuación, se crea una variable llamada *temp\_estrato* que combina dos variables de la encuesta “Zone” y “Sex” utilizando la función “paste0” para crear grupos de estratos. La función *table* se usa para mostrar la frecuencia de cada estrato.

Luego, se realiza el muestreo estratificado utilizando la función *S.STSI* que toma los siguientes argumentos:

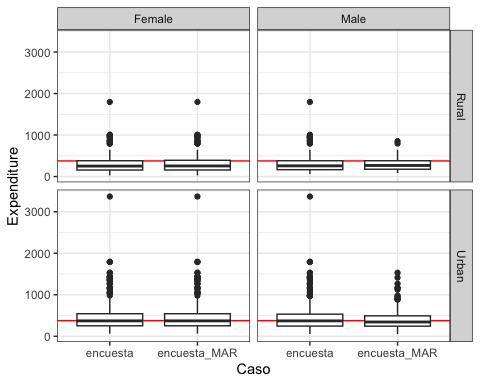
* “S”: el vector de estratos creado anteriormente
* “Nh”: el número de unidades en cada estrato (en este caso, 469, 411, 510 y 390)
* “nh”: el tamaño de muestra deseado para cada estrato (en este caso, 20, 380, 20 y 280)

El resultado del muestreo estratificado es un vector de índices de fila que corresponden a las observaciones seleccionadas para la muestra. Luego, se crea un nuevo conjunto de datos llamado “encuesta\_MAR” que excluye las observaciones seleccionadas en la muestra.

Finalmente, se usa la función *bind\_rows* del paquete *dplyr* para unir los dos conjuntos de datos, “encuesta” y “encuesta\_MAR”, en un solo conjunto de datos llamado “dat\_plot2”, con una nueva variable “Caso” que indica el caso de cada observación en el conjunto de datos.

Observemos gráficamente el efecto de la perdida de dinformación en una encuesta en un esquema MAR:

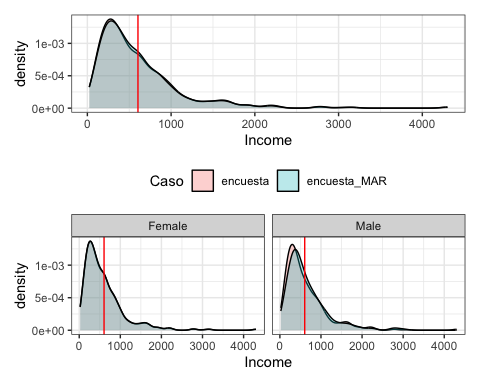
p1 <- ggplot(dat\_plot2, aes(x= Caso, y = Expenditure)) +   
 geom\_hline(yintercept = mean(encuesta$Expenditure),   
 col = "red") +   
 geom\_boxplot() +  
 facet\_grid(Zone~Sex) + theme\_bw()  
  
p1



En el gráfico anterior se logra observar un cambio en la distribución de los datos en las distintas desagregaciones cuando en la encuesta no se tiene pérdida de información y cuando sí se tiene con un esquema MAR. Naturalmente, esto afectaría en las estimaciones finales que se hagan de los parámetros estudiados.

Con mayor claridad, se puede ver el cambio distribucional en la siguiente gráfica:

p1 <- ggplot(dat\_plot2, aes(x = Income, fill = Caso)) +   
 geom\_density(alpha = 0.3) + theme\_bw() +  
 theme(legend.position = "bottom") +  
 geom\_vline(xintercept = mean(encuesta$Income),   
 col = "red")  
  
p2 <- ggplot(dat\_plot2, aes(x = Income, fill = Caso)) +   
 facet\_grid(.~Sex) +  
 geom\_density(alpha = 0.3) + theme\_bw() +  
 theme(legend.position = "none") +  
 geom\_vline(xintercept = mean(encuesta$Income),   
 col = "red")  
p1/p2

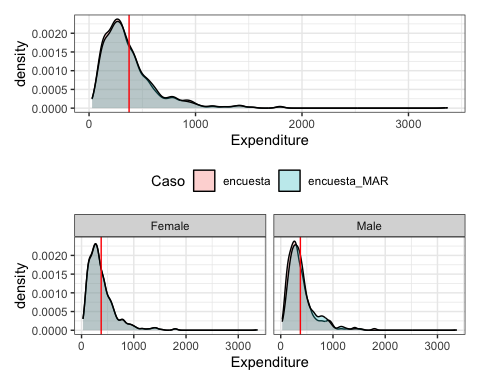


Este comportamiento es natural que suceda en un esquema MAR de datos faltantes puesto que, como se mencionó anteriormente, cuando se dice que los datos faltantes están “missing at random” (MAR), significa que la probabilidad de que los datos estén ausentes está relacionada con los valores observados en otras variables del conjunto de datos. En otras palabras, la probabilidad de que un valor esté ausente no está relacionada con el valor real del dato en sí mismo, sino que depende de la distribución de los datos en otras variables.

La ventaja que tienen los mecanismo de missing MAR es que se puede estimar el valor real de los datos faltantes utilizando la información de otras variables disponibles en el conjunto de datos. Esto puede mejorar la calidad de los resultados de los análisis y evitar la necesidad de descartar observaciones con datos faltantes.

Otra gráfica en donde se evidencia el cambio de distribución de los gastos entre hombres y mujeres.

p1 <- ggplot(dat\_plot2,   
 aes(x = Expenditure, fill = Caso)) +   
 geom\_density(alpha = 0.3) + theme\_bw() +  
 theme(legend.position = "bottom") +  
 geom\_vline(  
 xintercept = mean(encuesta$Expenditure),   
 col = "red")  
  
p2 <- ggplot(dat\_plot2,   
 aes(x = Expenditure, fill = Caso)) +   
 facet\_grid(.~Sex) +  
 geom\_density(alpha = 0.3) + theme\_bw() +  
 theme(legend.position = "none") +  
 geom\_vline(  
 xintercept = mean(encuesta$Expenditure),   
 col = "red")  
p1/p2



Ahora bien, para seguir con la ejemplificación de los esquemas de datos faltantes, generemos ahora un esquema de pérdida de información en una encuesta NMAR (siglas en inglés de “Not Missing at Random”). Como se mencionó al inicio de este capítulo, en este tipo de esquema, la probabilidad de que un dato falte está relacionada con el propio valor de ese dato, es decir, la probabilidad de que falte un dato no es aleatoria y depende de alguna característica o variable del propio dato.

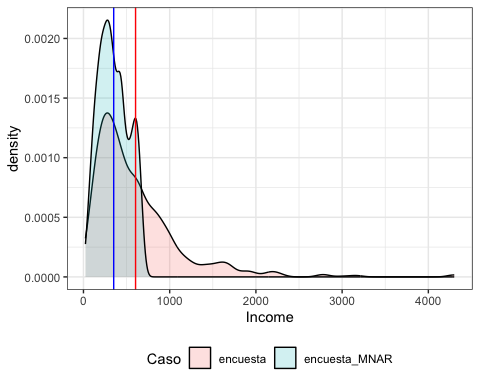
En otras palabras, en un esquema aleatorio NMAR, la probabilidad de que falte un dato no es independiente del valor de ese dato, sino que está influenciada por algún factor que puede estar relacionado con el fenómeno que se está estudiando. Esto puede llevar a que los datos faltantes introduzcan un sesgo en los resultados del análisis estadístico, lo que hace que el manejo adecuado de los datos faltantes en este tipo de esquemas sea particularmente importante en la investigación.

encuesta\_MNAR <- encuesta %>%   
 arrange((Income)) %>%   
 slice(1:1300L)  
  
dat\_plot3 <- bind\_rows(  
 list(encuesta\_MNAR = encuesta\_MNAR,  
 encuesta = encuesta), .id = "Caso" )

El código anterior tiene como objetivo crear un nuevo conjunto de datos llamado *encuesta\_MNAR* que contiene las primeras 1300 observaciones del conjunto de datos original *encuesta*, ordenadas por la variable *Income*. Luego, el código une el conjunto de datos original *encuesta* con el conjunto de datos *encuesta\_MNAR* usando la función *bind\_rows()*, y crea una nueva variable llamada “Caso” que indica la fuente de los datos.

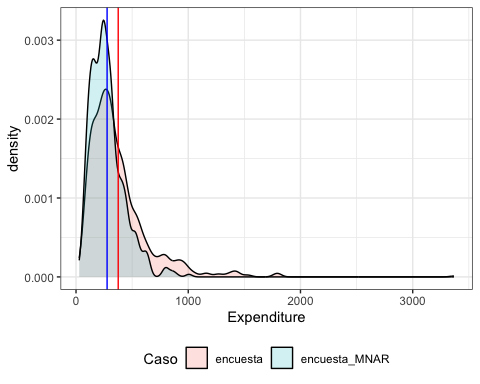
Ahora bien, para ver el efecto que tiene en una encuesta el tener datos faltante con esquema NMAR, se ilustran los siguientes gráficos:

p1 <- ggplot(dat\_plot3, aes(x = Income, fill = Caso)) +   
 geom\_density(alpha = 0.2) + theme\_bw() +  
 theme(legend.position = "bottom") +  
 geom\_vline(  
 xintercept = mean(encuesta$Income),   
 col = "red") +  
 geom\_vline(  
 xintercept = mean(encuesta\_MNAR$Income),   
 col = "blue")  
p1



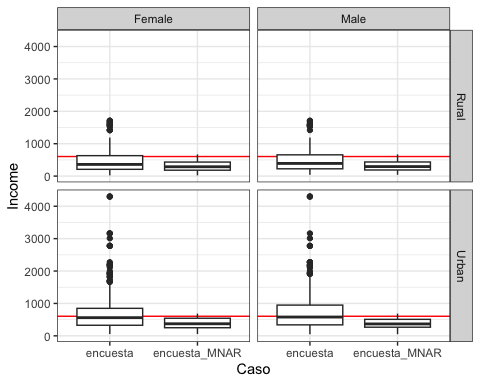
Como se puede observar en la gráfica anterior, la distribución de los ingresos cambia notablemente cuando se tienen datos faltantes con esquema NMAR, lo mismo sucede con la variable gastos, como se puede observar en la siguiente gráfica:

p1 <- ggplot(dat\_plot3,   
 aes(x = Expenditure, fill = Caso)) +   
 geom\_density(alpha = 0.2) + theme\_bw() +  
 theme(legend.position = "bottom") +  
 geom\_vline(  
 xintercept = mean(encuesta$Expenditure),   
 col = "red") +  
 geom\_vline(  
 xintercept = mean(encuesta\_MNAR$Expenditure),   
 col = "blue")  
p1



Para ver más al detalle el impacto que tiene la no respuesta con un esquema NMAR, a continuación se muestra una gráfica del ingreso discriminada por sexo y por zona. También se nota un cambio en la distribución de los ingresos significativos.

p1 <- ggplot(dat\_plot3, aes(x= Caso, y = Income)) +   
 geom\_hline(yintercept = mean(encuesta$Income),  
 col = "red") + geom\_boxplot() +  
 facet\_grid(Zone~Sex) + theme\_bw()  
  
p1



Para efectos de ejemplificar la solución del problema a los datos faltantes en una encuesta de hogares, generemos la siguiente base de datos:

encuesta <- full\_join(  
 encuesta,  
 encuesta\_MCAR %>%   
 dplyr::select(HHID, PersonID, Income, Employment) %>%  
 mutate(  
 Income\_missin = Income,  
 Employment\_missin = Employment,  
 Employment = NULL,  
 Income = NULL  
 )  
)

El código anterior utiliza la función *full\_join()* de la librería *dplyr* de ´R´ para combinar dos conjuntos de datos: encuesta y encuesta\_MCAR. La combinación se realiza mediante la unión completa (o full join), que devuelve todas las filas de ambas tablas, uniendo las filas con valores coincidentes y rellenando con valores faltantes para las columnas que no tienen una coincidencia en ambas tablas.

La segunda Base de datos, encuesta\_MCAR, se transforma previamente con las siguientes operaciones:

* Se seleccionan las columnas HHID, PersonID, Income y Employment mediante la función *dplyr::select()*.
* Se agregan dos nuevas columnas, Income\_missin y Employment\_missin, utilizando la función mutate().

Los valores de estas columnas son idénticos a los de las columnas Income y Employment, respectivamente. Las columnas Income y Employment se eliminan del conjunto de datos utilizando la función *mutate()* y asignando el valor NULL a ambas columnas. Finalmente, el resultado de la unión completa se asigna a la variable encuesta.

Ahora bien, para tener como referencia el porcentaje de datos faltantes, se ejecuta el siguiente comando:

encuesta %>% group\_by(Zone) %>%   
 summarise(Income = sum(is.na(Income\_missin) / n()))

| Zone | Income |
| --- | --- |
| Rural | 0.2079208 |
| Urban | 0.1927835 |

encuesta %>% group\_by(Sex) %>%   
 summarise(Income = sum(is.na(Income\_missin) / n()))

| Sex | Income |
| --- | --- |
| Female | 0.1837945 |
| Male | 0.2191465 |

## 10.1 Imputación por la media no condicional.

La imputación por la media no condicional consiste en reemplazar los valores faltantes con la media aritmética de la variable completa, sin tener en cuenta ninguna otra variable. Es decir, la media se calcula a partir de todos los valores disponibles en la variable en cuestión, independientemente de las características de los demás datos.

Este método es bastante simple y rápido, y puede ser útil en ciertas situaciones, especialmente cuando la variable en cuestión no tiene una distribución muy sesgada o cuando los valores faltantes son relativamente pocos en comparación con el tamaño de la muestra. Sin embargo, el método de imputación por la media no condicional también tiene limitaciones y puede no ser adecuado en todas las situaciones, especialmente cuando hay sesgos o patrones en los datos faltantes o cuando los datos están altamente correlacionados. Adicionalmente, este método no afecta el promedio, pero si afecta la variabilidad, el sesgo y los percentiles. A continuación, se ejemplifica con los datos de ejemplo este método:

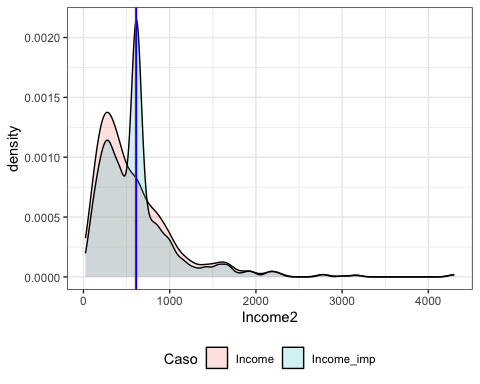
promedio <- mean(encuesta$Income\_missin, na.rm = TRUE)  
encuesta %<>% dplyr::mutate(  
 Income\_imp = ifelse(is.na(Income\_missin),   
 promedio, Income\_missin))  
sum(is.na(encuesta$Income\_imp))

## [1] 0

En el código anterior la imputación se realiza utilizando la media aritmética de los valores no faltantes en *Income\_missin* y se almacena en una nueva variable llamada *Income\_imp*. A continuación, se describen cada una de las líneas del código:

* promedio <- mean(encuesta$Income\_missin, na.rm = TRUE): esta línea calcula la media aritmética de los valores no faltantes en la columna *Income\_missin* de la base de datos *encuesta* y la almacena en una variable llamada *promedio*. El argumento *na.rm = TRUE* se utiliza para excluir los valores faltantes en el cálculo de la media.
* encuesta %<>%: este operador de *magrittr (%<>%)* se utiliza para asignar el resultado de la siguiente operación al objeto *encuesta*. Es equivalente a utilizar *encuesta <- encuesta %>%*.
* mutate: esta función de *dplyr* se utiliza para crear una nueva columna en la base de datos *encuesta* con la imputación de los valores faltantes.
* Income\_imp = ifelse(is.na(Income\_missin), promedio, Income\_missin): esta línea utiliza la función *ifelse* para asignar el valor imputado a la columna *Income\_imp* en la base de datos *encuesta*. Si un valor en la columna *Income\_missin* es NA (es decir, faltante), se reemplaza con el valor de promedio. Si no es NA, se mantiene el valor original.
* sum(is.na(encuesta$Income\_imp)): esta línea cuenta el número de valores faltantes en la nueva *columna Income\_imp* de la base de datos encuesta utilizando la función *sum* y *is.na*.

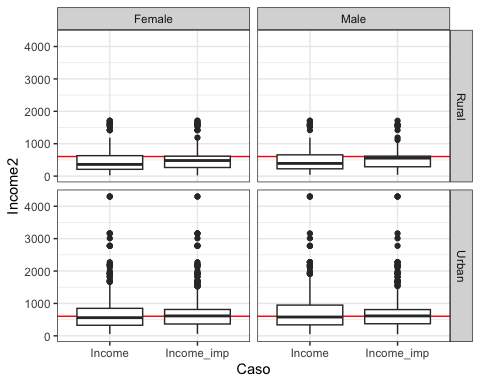
## Ordenando la base para gráfica   
dat\_plot4 <- tidyr::gather(  
 encuesta %>% dplyr::select(Zone,Sex,Income, Income\_imp),  
 key = "Caso", value = "Income2", -Zone,-Sex)  
  
p1 <- ggplot(dat\_plot4, aes(x = Income2, fill = Caso)) +   
 geom\_density(alpha = 0.2) + theme\_bw() +  
 theme(legend.position = "bottom") +  
 geom\_vline(  
 xintercept = mean(encuesta$Income),   
 col = "red") +  
 geom\_vline(  
 xintercept = mean(encuesta$Income\_imp),   
 col = "blue")  
p1



En la gráfica anterior se puede observar que, a pesar de ser imputada la variable ingresos utilizando la media incondicional, la distribución real y la imputada cambia de manera significativa lo que demuestra que este método, para este conjunto de datos no es el más apropiado dado lo sesgado de la distribución de la variable ingresos.

Un caso similar al anterior ocurre si graficamos la variable ingreso por zona y sexo. A continuación, se muestra de manera gráfica los boxplot para revisar la distribución de los datos, arrojando nuevamente las conclusiones obtenidas con el gráfico anterior:

p1 <- ggplot(dat\_plot4, aes(x= Caso, y = Income2)) +   
 geom\_hline(yintercept = mean(encuesta$Income),  
 col = "red") + geom\_boxplot() +  
 facet\_grid(Zone~Sex) + theme\_bw()  
  
p1



## 10.2 Imputación por la media condicional

El método de imputación por la media condicional es una técnica utilizada en el análisis de datos para tratar valores faltantes o perdidos en una variable numérica. A diferencia del método de imputación por la media no condicional, el método de imputación por la media condicional tiene en cuenta otras variables en el conjunto de datos al calcular la media.

La imputación por la media condicional se basa en la idea de que la media de una variable puede variar en función de los valores de otras variables. Por lo tanto, en lugar de simplemente reemplazar los valores faltantes con la media aritmética de la variable completa, se utiliza la media de la variable para grupos de observaciones que tienen valores similares en otras variables.

Por ejemplo, si se tiene una variable numérica llamada “Ingreso” y una variable categórica llamada “Educación”, se podría utilizar la media de ingresos para cada nivel de educación para imputar los valores faltantes en la variable “Ingreso”. De esta manera, se tiene en cuenta la relación entre la educación y el ingreso al realizar la imputación.

El método de imputación por la media condicional puede ser más preciso que el método de imputación por la media no condicional en situaciones en las que las variables están correlacionadas o cuando hay patrones de valores faltantes en los datos. Sin embargo, también puede ser más complicado y requiere más tiempo y recursos computacionales para implementar. A continuación, se ejemplifica la técnica de imputación utilizando la variable estrato (Stratum en la base de datos) para hacer el cálculo de los promedios por cada uno de los estratos y así poder imputar los datos faltantes. Asumiendo que hay una relación directa entre los estratos y los ingresos de los hogares:

encuesta %<>% group\_by(Stratum) %>%  
 mutate( Income\_imp = ifelse(is.na(Income\_missin),  
 mean(Income\_missin, na.rm = TRUE),  
 Income\_missin)) %>% data.frame()  
sum(is.na(encuesta$Income\_imp))

## [1] 0

encuesta %<>%  
 mutate( Income\_imp = ifelse(is.na(Income\_imp),   
 promedio, Income\_imp))  
sum(is.na(encuesta$Income\_imp))

## [1] 0

A continuación, se decribe el código computacional utilizado:

* encuesta %<>% group\_by(Stratum) %>%: este código utiliza la función *group\_by* de *dplyr* para agrupar las observaciones de la base de datos encuesta por los niveles de la variable *Stratum*. El operador %>% se utiliza para concatenar este código con el siguiente código en una sola cadena de operaciones.
* mutate: esta función de *dplyr* se utiliza para crear una nueva columna en la base de datos encuesta con la imputación de los valores faltantes.
* Income\_imp = ifelse(is.na(Income\_missin), mean(Income\_missin, na.rm = TRUE), Income\_missin)): esta línea utiliza la función *ifelse* para asignar el valor imputado a la columna *Income\_imp* en la base de datos encuesta. Si un valor en la columna *Income\_missin* es NA (es decir, faltante), se reemplaza con la media aritmética de los valores no faltantes en *Income\_missin* dentro del grupo correspondiente. Si no es NA, se mantiene el valor original.
* sum(is.na(encuesta$Income\_imp)): esta línea cuenta el número de valores faltantes en la nueva columna *Income\_imp* de la base de datos encuesta utilizando la función *sum* y *is.na*.
* encuesta %<>% mutate( Income\_imp = ifelse(is.na(Income\_imp), promedio, Income\_imp)): este código utiliza la función *mutate* para crear una nueva columna en la base de datos encuesta y reemplazar cualquier valor faltante en la columna *Income\_imp* con la media aritmética de los valores no faltantes en *Income\_missin*. El valor de la media aritmética se almacena en una variable llamada promedio. Si un valor en la columna *Income\_imp* no es NA, se mantiene el valor original.
* sum(is.na(encuesta$Income\_imp)): esta línea cuenta el número de valores faltantes en la nueva columna *Income\_imp* de la base de datos encuesta utilizando la función *sum* y *is.na*.

A continuación, se calculan las medias y desviaciones estándar tanto para los datos imputados como los originales y así poder comparar le efecto de la imputación realizada:

encuesta %>% summarise(  
 Income\_ = mean(Income),  
 Income\_sd = sd(Income),  
 Income\_imp\_ = mean(Income\_imp),  
 Income\_imp\_sd = sd(Income\_imp))

| Income\_ | Income\_sd | Income\_imp\_ | Income\_imp\_sd |
| --- | --- | --- | --- |
| 604.2494 | 513.1078 | 611.545 | 488.7209 |

Para poder comparar los resultados, calculemos el sesgo relativo de la imputación el cual se calcula como sigue:

100\*(604.2494- 611.545)/604.2494

## [1] -1.207382

Como se puede observar, el sesgo relativo para el primedio de los ingresos es menor al 1.5%. Ahora bien, siguiendo la misma idea, el sesgo relativo para la desviación es:

100\*(513.1078- 488.7209)/513.1078

## [1] 4.752783

Lo que generó un sesgo relativo para la desviación estándar inferior al 5%. Ahora bien, si se realiza la imputación utilizando la media condicional agrupando por la variable zona, se tienen los siguientes resultados:

encuesta %>%group\_by(Zone) %>% summarise(  
 Income\_ = mean(Income),  
 Income\_sd = sd(Income),  
 Income\_imp\_ = mean(Income\_imp),  
 Income\_imp\_sd = sd(Income\_imp))

| Zone | Income\_ | Income\_sd | Income\_imp\_ | Income\_imp\_sd |
| --- | --- | --- | --- | --- |
| Rural | 469.1217 | 336.5861 | 477.9042 | 305.5101 |
| Urban | 730.8793 | 609.0304 | 736.7815 | 585.6550 |

Realizando el mismo ejercicio anterior, se obtienen sesgos relativos para la media de los ingresos para la zona rural de 1.87%:

100\*(469.1217- 477.9042)/469.1217

## [1] -1.872115

y para la zona urbana de 0.8%.:

100\*(730.8793- 736.7815)/730.8793

## [1] -0.8075478

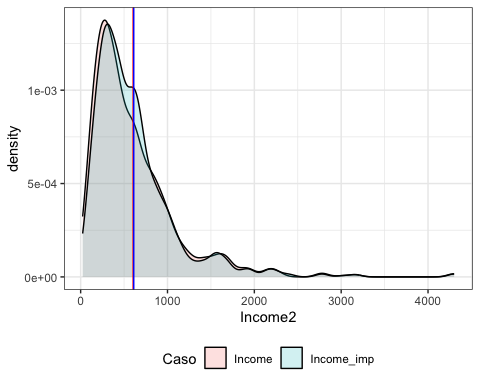
En ambos casos se observa una buena imputación de los ingresos. Ejercicio similar se puede realizar para sexo (Se le deja al lector realizar el cálculo del sesgo relativo).

encuesta %>%group\_by(Sex) %>% summarise(  
 Income\_ = mean(Income),  
 Income\_sd = sd(Income),  
 Income\_imp\_ = mean(Income\_imp),  
 Income\_imp\_sd = sd(Income\_imp))

| Sex | Income\_ | Income\_sd | Income\_imp\_ | Income\_imp\_sd |
| --- | --- | --- | --- | --- |
| Female | 589.2330 | 504.3041 | 600.3235 | 486.5818 |
| Male | 621.7771 | 522.9428 | 624.6431 | 491.1625 |

Ahora bien, para observar la distribución de los datos imputados en comparación con los no imputado se realizan las siguientes gráficas:

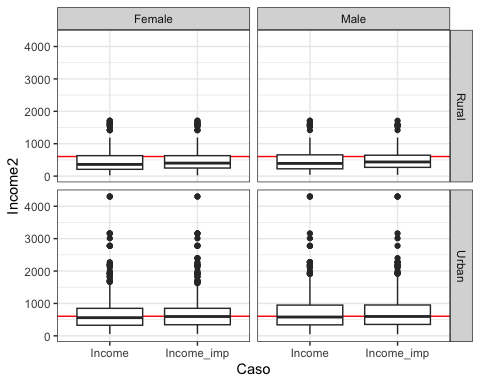
dat\_plot5 <- tidyr::gather(  
 encuesta %>% dplyr::select(Zone,Sex,Income, Income\_imp),  
 key = "Caso", value = "Income2", -Zone,-Sex)  
  
p1 <- ggplot(dat\_plot5, aes(x = Income2, fill = Caso)) +   
 geom\_density(alpha = 0.2) + theme\_bw() +  
 theme(legend.position = "bottom") +  
 geom\_vline(  
 xintercept = mean(encuesta$Income),   
 col = "red") +  
 geom\_vline(  
 xintercept = mean(encuesta$Income\_imp),   
 col = "blue")  
p1



Se puede observar que de manera general, la distribución de las observaciones imputadas y originales tienen un comportamiento mejor que con la media no condicional.

Si se observa ahora la distribunción de los datos por zona y sexo, se puede observar también una buena imputación de las observaciones.

p1 <- ggplot(dat\_plot5, aes(x= Caso, y = Income2)) +   
 geom\_hline(yintercept = mean(encuesta$Income),  
 col = "red") + geom\_boxplot() +  
 facet\_grid(Zone~Sex) + theme\_bw()  
p1



## 10.3 Imputación por Hot-deck y Cold-deck

La imputación *Hot deck* consiste en reemplazar los valores faltantes de una o más variables para un no encuestado (llamado receptor) con valores observados de un encuestado (el donante) que es similar al no encuestado con respecto a las características observadas en ambos casos.

La técnica se basa en la idea de que las observaciones similares pueden tener valores similares para las variables de interés. En el enfoque por hot-deck, se selecciona una observación donante que sea similar a la observación receptora en términos de características relevantes (por ejemplo, edad, género, ubicación geográfica, etc.), y se utiliza su valor observado para imputar el valor faltante en la observación receptora.

El término “hot-deck” se refiere a una tarjeta perforada que se utilizaba en los primeros sistemas informáticos para almacenar y recuperar datos. En el enfoque por hot-deck, las observaciones se organizan en una “pila” o “mazo” de tarjetas, y las observaciones similares se seleccionan de esta pila para imputar los valores faltantes.

La imputación por hot-deck es una técnica relativamente simple y eficaz para imputar valores faltantes en conjuntos de datos pequeños o medianos, y se utiliza comúnmente en encuestas y estudios de investigación social. Sin embargo, puede ser menos efectiva en conjuntos de datos grandes o complejos, donde puede ser difícil encontrar observaciones similares o donde las características relevantes son difíciles de definir o medir de manera confiable.

Por otro lado, el método llamado *Cold-deck* por analogía con *Hot-deck* consiste en reemplazar el valor faltante por valores de una fuente no relacionada con el conjunto de datos en consideración. El método de imputación Cold-deck es una técnica de imputación de datos faltantes que se basa en la sustitución de los valores faltantes por valores observados de una fuente externa, tal como un conjunto de datos históricos, registros administrativos u otras fuentes de datos secundarios.

A diferencia del método de imputación por hot-deck, que utiliza información de la propia muestra para imputar los valores faltantes, el método de imputación cold-deck se basa en la utilización de información externa para sustituir los valores faltantes. La fuente de datos externa se utiliza para imputar los valores faltantes en la muestra de estudio.

El término “cold-deck” hace referencia a la tarjeta perforada que se utilizaba en los primeros sistemas informáticos para almacenar y recuperar datos. En el método de imputación cold-deck, los valores faltantes se imputan a partir de los datos de la “pila fría” o “mazo frío” de tarjetas perforadas, que corresponden a los datos históricos o a la fuente externa de datos.

El método de imputación cold-deck se utiliza a menudo cuando la muestra de estudio es pequeña o no hay suficientes observaciones similares para aplicar el método de imputación por hot-deck. Sin embargo, el método de imputación cold-deck tiene algunas limitaciones, como la posibilidad de que los datos externos no sean representativos de la muestra de estudio y la posibilidad de introducir errores en los datos imputados si la fuente de datos externa no es fiable o no es adecuada para la variable de interés.

*Imputación por hot-deck*

Iniciamos los ejemplos en esta sección con el método hot-deck. A continuación, se presenta un código computacional que ejemplifica, para los datos que estamos usando en el capítulo, el uso del método hot-deck.

donante <- which(!is.na(encuesta$Income\_missin))  
receptor <- which(is.na(encuesta$Income\_missin))  
encuesta$Income\_imp <- encuesta$Income\_missin  
set.seed(1234)  
for(ii in receptor){  
don\_ii <- sample(x = donante, size = 1)  
encuesta$Income\_imp[ii] <- encuesta$Income\_missin[don\_ii]}  
sum(is.na(encuesta$Income\_imp))

## [1] 0

El código mostrado anteriormente se describe a continuación:

* La primera línea del código selecciona las observaciones que no tienen valores faltantes en la variable “Income\_missin” utilizando la función *which* y el operador *!* (not) *donante <- which(!is.na(encuesta$Income\_missin))*
* La segunda línea selecciona las observaciones que tienen valores faltantes en la variable “Income\_missin” utilizando la función *which* y el operador *is.na*. *receptor <- which(is.na(encuesta$Income\_missin))*
* La tercera línea crea una nueva variable “Income\_imp” en la encuesta, que se utilizará para almacenar los valores imputados *encuestaIncome\_missin*
* La cuarta línea utiliza la función *set.seed* para establecer una semilla aleatoria para asegurar la reproducibilidad del proceso de imputación. *set.seed(1234)*
* Se utiliza un bucle *for* para iterar a través de cada observación receptora. Dentro del bucle, se utiliza la función *sample* para seleccionar una observación donante aleatoria de entre las observaciones que no tienen valores faltantes en la variable “Income\_missin”.

Una vez realizada la imputación, se calcula la media y la desviación de los datos completos e imputados:

encuesta %>% summarise(  
 Income\_ = mean(Income),  
 Income\_sd = sd(Income),  
 Income\_imp\_ = mean(Income\_imp),  
 Income\_imp\_sd = sd(Income\_imp))

| Income\_ | Income\_sd | Income\_imp\_ | Income\_imp\_sd |
| --- | --- | --- | --- |
| 604.2494 | 513.1078 | 618.2937 | 528.2157 |

Como en los métodos de imputación anterior, calculemos el sesgo relativo de la imputación el cual fue de 2.3%:

100\*(604.2494-618.2937)/604.2494

## [1] -2.324256

Ahora bien, haciendo el mismo ejercicio, pero esta vez desagregada por zona tenemos:

encuesta %>%group\_by(Zone) %>% summarise(  
 Income\_ = mean(Income),  
 Income\_sd = sd(Income),  
 Income\_imp\_ = mean(Income\_imp),  
 Income\_imp\_sd = sd(Income\_imp))

| Zone | Income\_ | Income\_sd | Income\_imp\_ | Income\_imp\_sd |
| --- | --- | --- | --- | --- |
| Rural | 469.1217 | 336.5861 | 503.7127 | 368.9137 |
| Urban | 730.8793 | 609.0304 | 725.6691 | 623.9875 |

El sesgo relativo de la estimación en la zona rural es de 7.4% y en al zona urbana es de 0.7%.

100\*(469.1217-503.7127)/469.1217

## [1] -7.373566

100\*(730.8793-725.6691)/730.8793

## [1] 0.7128674

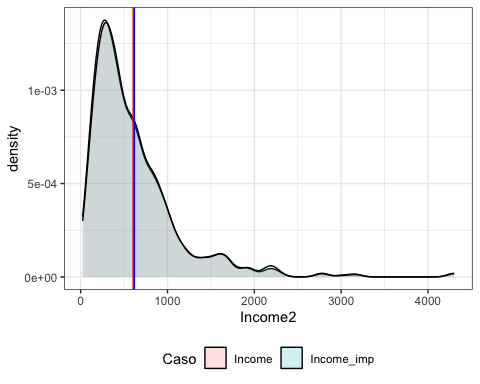
El mismo ejercicio se puede realizar por sexo:

encuesta %>%group\_by(Sex) %>% summarise(  
 Income\_ = mean(Income),  
 Income\_sd = sd(Income),  
 Income\_imp\_ = mean(Income\_imp),  
 Income\_imp\_sd = sd(Income\_imp))

| Sex | Income\_ | Income\_sd | Income\_imp\_ | Income\_imp\_sd |
| --- | --- | --- | --- | --- |
| Female | 589.2330 | 504.3041 | 602.8075 | 503.0951 |
| Male | 621.7771 | 522.9428 | 636.3699 | 555.8522 |

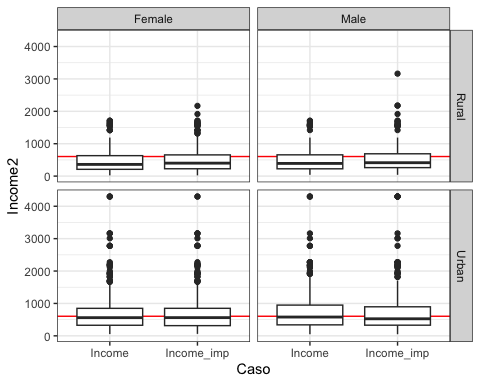
Como en los ejercicios anteriores, a continuación, se muestra la gráfica de la distribución de los datos tanto los completos como los imputados observándose que la distribución de los datos imputados es muy similar a la de los datos no imputados:

dat\_plot6 <- tidyr::gather(  
 encuesta %>% dplyr::select(Zone,Sex,Income, Income\_imp),  
 key = "Caso", value = "Income2", -Zone,-Sex)  
  
p1 <- ggplot(dat\_plot6, aes(x = Income2, fill = Caso)) +   
 geom\_density(alpha = 0.2) + theme\_bw() +  
 theme(legend.position = "bottom") +  
 geom\_vline(  
 xintercept = mean(encuesta$Income),   
 col = "red") +  
 geom\_vline(  
 xintercept = mean(encuesta$Income\_imp),   
 col = "blue")  
p1



El mismo ejercicio anterior se realiza por zona y sexo obteniendo resultados similares a los abtenidos en el gráfico anterior:

p1 <- ggplot(dat\_plot6, aes(x= Caso, y = Income2)) +   
 geom\_hline(yintercept = mean(encuesta$Income),  
 col = "red") + geom\_boxplot() +  
 facet\_grid(Zone~Sex) + theme\_bw()  
p1



A continuación, se implementa el método de imputación pero para la variable empleado. Los códigos computacionales son similares a los empleados con la variable ingresos:

donante <- which(!is.na(encuesta$Income\_missin))  
receptor <- which(is.na(encuesta$Income\_missin))  
encuesta$Employment\_imp <- encuesta$Employment\_missin  
(prop <- prop.table(  
 table(na.omit(encuesta$Employment\_missin))))

| Unemployed | Inactive | Employed |
| --- | --- | --- |
| 0.0425815 | 0.3739188 | 0.5834997 |

set.seed(1234)  
imp <- sample(size = length(receptor),  
 c("Unemployed", "Inactive","Employed"),  
 prob = prop, replace = TRUE )  
encuesta$Employment\_imp[receptor] <- imp   
sum(is.na(encuesta$Employment\_imp))

## [1] 0

Se le deja al lector realizar los ejercicios de comprobación gráfica, como se mostró a lo largo de esta sección.

## 10.4 Imputación por regresión

La imputación por regresión es una técnica de análisis de datos que se utiliza para imputar valores faltantes en un conjunto de datos. Esta técnica se basa en la construcción de un modelo de regresión a partir de las variables disponibles en el conjunto de datos, que se utiliza para predecir los valores faltantes . Cuando se habla de predicción no se refiere a dar un valor futuro, se refiere a dar un valor a la información faltante.

Para llevar a cabo la imputación por regresión, se selecciona una variable objetivo que tenga valores faltantes y se identifican las variables predictoras que tienen una correlación significativa con la variable objetivo. Se ajusta un modelo de regresión utilizando las variables predictoras y la variable objetivo disponible, y se utilizan los coeficientes del modelo para predecir los valores faltantes de la variable objetivo.

Es importante destacar que la imputación por regresión es una técnica estadística avanzada que requiere conocimientos sólidos de análisis de datos y modelado estadístico. Además, su aplicación puede verse limitada por la calidad y la cantidad de los datos disponibles y por la distribución de los valores faltantes en el conjunto de datos. Por lo tanto, es importante utilizarla con precaución y tener en cuenta sus limitaciones.

Para ejemplificar, imputemos la variable ingreso y la variable empleados tomadno como covariables las variable zona, sexo y empleamiento. Para la primera variable se utiliza un modelo de regresión líneal múltiple y para el segundo. se utiliza un modelo multinomial (dada la naturaleza de la variable) como se muestra a continución:

require(nnet)

## Loading required package: nnet

encuesta$Income\_imp <- encuesta$Income\_missin  
encuesta$Employment\_imp <- encuesta$Employment\_missin  
encuesta\_obs <- filter(encuesta, !is.na(Income\_missin))  
encuesta\_no\_obs <- filter(encuesta, is.na(Income\_missin))  
mod <- lm(Income~Zone + Sex +Expenditure, data = encuesta\_obs)  
mod.mult <- multinom(Employment~Zone + Sex +Expenditure, data = encuesta\_obs)

## # weights: 15 (8 variable)  
## initial value 1651.214270   
## iter 10 value 1182.110114  
## final value 1132.682019   
## converged

Una vez ajustado los modelos tanto para las variable ingreso como para empleados, se realiza el proceso de predicción como se muestra a continuación:

imp <- predict(mod, encuesta\_no\_obs)  
imp.mult <- predict(mod.mult, encuesta\_no\_obs, type = "class")  
encuesta\_no\_obs$Income\_imp <- imp  
encuesta\_no\_obs$Employment\_imp <- imp.mult  
encuesta <- bind\_rows(encuesta\_obs,encuesta\_no\_obs)

A continuación, se presenta el porcentaje de datos faltantes en la variable empleado:

prop.table(table(encuesta$Employment\_missin, useNA = "a"))

| Unemployed | Inactive | Employed | NA |
| --- | --- | --- | --- |
| 0.0340607 | 0.2990953 | 0.4667376 | 0.2001064 |

Se puede observar que hay un 20% de datos faltantes. Una vez se realiza la imputación, se redistribuyen esas observaciones en las demás categorías arrojando los siguientes resultados:

prop.table(table(encuesta$Employment\_imp, useNA = "a"))

| Unemployed | Inactive | Employed | NA |
| --- | --- | --- | --- |
| 0.0340607 | 0.3858435 | 0.5800958 | 0 |

A modo de ejercicio, se realiza el cálculo del porcentaje de los valores faltante para la variable empleados por zona, antes y después de imputar, reconociendo que, los porcentajes marginales por zona no varían:

library(printr)  
library(kableExtra)  
  
kable(prop.table( table(encuesta$Zone, encuesta$Employment\_missin, useNA = "a")) %>% addmargins())

|  | Unemployed | Inactive | Employed | NA | Sum |
| --- | --- | --- | --- | --- | --- |
| Rural | 0.0117084 | 0.1506120 | 0.2208622 | 0.1005854 | 0.483768 |
| Urban | 0.0223523 | 0.1484832 | 0.2458755 | 0.0995210 | 0.516232 |
| NA | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.000000 |
| Sum | 0.0340607 | 0.2990953 | 0.4667376 | 0.2001064 | 1.000000 |

kable(  
prop.table( table(encuesta$Zone, encuesta$Employment\_imp,useNA = "a")) %>% addmargins()  
)

|  | Unemployed | Inactive | Employed | NA | Sum |
| --- | --- | --- | --- | --- | --- |
| Rural | 0.0117084 | 0.2006386 | 0.2714210 | 0 | 0.483768 |
| Urban | 0.0223523 | 0.1852049 | 0.3086748 | 0 | 0.516232 |
| NA | 0.0000000 | 0.0000000 | 0.0000000 | 0 | 0.000000 |
| Sum | 0.0340607 | 0.3858435 | 0.5800958 | 0 | 1.000000 |

El mismo ejercicio anterior se realiza por sexo arrojando los siguientes resultados:

kable(  
 prop.table( table(encuesta$Sex, encuesta$Employment\_missin, useNA = "a")) %>% addmargins()  
)

|  | Unemployed | Inactive | Employed | NA | Sum |
| --- | --- | --- | --- | --- | --- |
| Female | 0.0106440 | 0.2277807 | 0.2011708 | 0.0989888 | 0.5385844 |
| Male | 0.0234167 | 0.0713145 | 0.2655668 | 0.1011176 | 0.4614156 |
| NA | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| Sum | 0.0340607 | 0.2990953 | 0.4667376 | 0.2001064 | 1.0000000 |

kable(  
prop.table( table(encuesta$Sex, encuesta$Employment\_imp,useNA = "a")) %>% addmargins()  
)

|  | Unemployed | Inactive | Employed | NA | Sum |
| --- | --- | --- | --- | --- | --- |
| Female | 0.0106440 | 0.3145290 | 0.2134114 | 0 | 0.5385844 |
| Male | 0.0234167 | 0.0713145 | 0.3666844 | 0 | 0.4614156 |
| NA | 0.0000000 | 0.0000000 | 0.0000000 | 0 | 0.0000000 |
| Sum | 0.0340607 | 0.3858435 | 0.5800958 | 0 | 1.0000000 |

Para finalizar y como se ha realizado con los métodos anteriores, se hace el cálculo de la variable ingreso completa e imputada:

encuesta %>% summarise(  
 Income\_ = mean(Income),  
 Income\_sd = sd(Income),  
 Income\_imp\_ = mean(Income\_imp),  
 Income\_imp\_sd = sd(Income\_imp))

| Income\_ | Income\_sd | Income\_imp\_ | Income\_imp\_sd |
| --- | --- | --- | --- |
| 604.2494 | 513.1078 | 611.7477 | 498.3293 |

Teniendo un sesgo relativo de 1.2%.

100\*(604.2494 - 611.7477)/604.2494

## [1] -1.240928

Haciendo el mismo ejercicio por zona tenemos:

encuesta %>%group\_by(Zone) %>% summarise(  
 Income\_ = mean(Income),  
 Income\_sd = sd(Income),  
 Income\_imp\_ = mean(Income\_imp),  
 Income\_imp\_sd = sd(Income\_imp))

| Zone | Income\_ | Income\_sd | Income\_imp\_ | Income\_imp\_sd |
| --- | --- | --- | --- | --- |
| Rural | 469.1217 | 336.5861 | 476.1361 | 317.3999 |
| Urban | 730.8793 | 609.0304 | 738.8311 | 594.5319 |

Con sesgos relativos para la zona rural de 1.5% y para urbano de 1%.

100\*(469.1217 - 476.1361)/469.1217

## [1] -1.49522

100\*(730.8793 - 738.8311)/730.8793

## [1] -1.087977

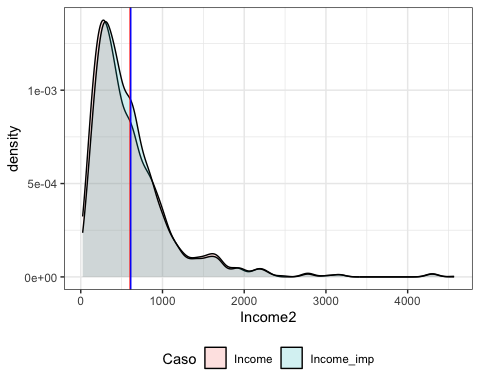
Para la variable sexo, se puede realizar el mismo ejercicio anterior. Se le deja al lector hacer los cálculos pertinentes:

encuesta %>%group\_by(Sex) %>% summarise(  
 Income\_ = mean(Income),  
 Income\_sd = sd(Income),  
 Income\_imp\_ = mean(Income\_imp),  
 Income\_imp\_sd = sd(Income\_imp))

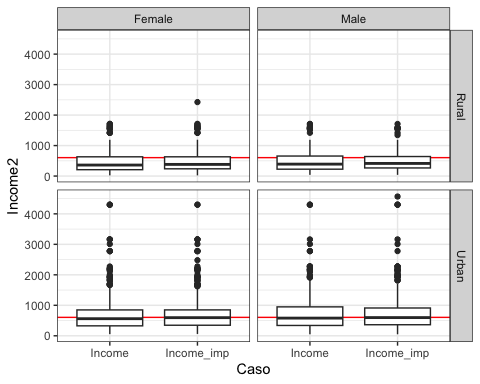
| Sex | Income\_ | Income\_sd | Income\_imp\_ | Income\_imp\_sd |
| --- | --- | --- | --- | --- |
| Female | 589.2330 | 504.3041 | 598.6515 | 488.3917 |
| Male | 621.7771 | 522.9428 | 627.0341 | 509.5410 |

Por último, los ejercicios gráficos se realizan a continuación:

dat\_plot7 <- tidyr::gather(  
 encuesta %>% dplyr::select(Zone,Sex,Income, Income\_imp),  
 key = "Caso", value = "Income2", -Zone,-Sex)  
  
p1 <- ggplot(dat\_plot7, aes(x = Income2, fill = Caso)) +   
 geom\_density(alpha = 0.2) + theme\_bw() +  
 theme(legend.position = "bottom") +  
 geom\_vline(  
 xintercept = mean(encuesta$Income),   
 col = "red") +  
 geom\_vline(  
 xintercept = mean(encuesta$Income\_imp),   
 col = "blue")  
p1



p1 <- ggplot(dat\_plot7, aes(x= Caso, y = Income2)) +   
 geom\_hline(yintercept = mean(encuesta$Income),  
 col = "red") + geom\_boxplot() +  
 facet\_grid(Zone~Sex) + theme\_bw()  
p1



Obteniendo buenos resultados en el proceso de imputación como se pudo observar en las gráficas anteriores.

## 10.5 Imputación por el vecino más cercano

La imputación por el vecino más cercano (K-nearest neighbor imputation en inglés) es una técnica de análisis de datos utilizada para estimar los valores faltantes en un conjunto de datos. En esta técnica, los valores faltantes se reemplazan por valores de otras observaciones que son similares a la observación con valores faltantes.

La imputación por el vecino más cercano se basa en la idea de que los registros similares tienden a tener valores similares para una determinada variable. La técnica consiste en encontrar los registros más similares a la observación con valores faltantes en función de las variables disponibles en el conjunto de datos y utilizar los valores de estas observaciones para estimar el valor faltante.

Para calcular la similitud entre observaciones, se pueden utilizar diferentes medidas de distancia, como la distancia euclidiana o la distancia de Manhattan. La técnica también permite ajustar el valor de , que representa el número de vecinos más cercanos utilizados para estimar el valor faltante.

Es importante destacar que la imputación por el vecino más cercano es una técnica relativamente simple y fácil de implementar. Sin embargo, su eficacia puede verse limitada por la cantidad y la calidad de los datos disponibles, así como por la elección de los parámetros (como el valor de y la medida de distancia) que pueden afectar significativamente los resultados obtenidos. Por lo tanto, es importante evaluar cuidadosamente la calidad de los datos y los resultados obtenidos antes de utilizar esta técnica.

Teniendo en cuenta lo anterior, se presentan 3 pasos a tener en cuenta al momento de utilizar esta técnica:

* **Paso 1**: Definir una magnitud de distancia (Distancia euclidiana, k-media, K-Medioides, entre otras).
* **Paso 2**: Para la -ésimo elemento identificar el donante, cual será el más cercano al receptor según la magnitud de distancia previamente definida.
* **Paso 3**: Se imputa el valor faltante con la información del donante identificado previamente.

Para ejemplificar esta metología, se va a imputar la variable ingresos y empleado utilizando como variable de apoyo los gastos del individuo. Se utilizará como distancia la euclidiana.

encuesta$Income\_imp <- encuesta$Income\_missin  
encuesta$Employment\_imp <- encuesta$Employment\_missin  
encuesta\_obs <- filter(encuesta, !is.na(Income\_missin))  
encuesta\_no\_obs <- filter(encuesta, is.na(Income\_missin))  
  
for(ii in 1:nrow(encuesta\_no\_obs)){  
  
Expen\_ii <- encuesta\_no\_obs$Expenditure[[ii]]  
don\_ii <- which.min(abs(Expen\_ii - encuesta\_obs$Expenditure))  
encuesta\_no\_obs$Income\_imp[[ii]] <- encuesta\_obs$Income\_missin[[don\_ii]]  
encuesta\_no\_obs$Employment\_imp[[ii]] <- encuesta\_obs$Employment\_missin[[don\_ii]]  
}  
  
encuesta <- bind\_rows(encuesta\_obs,encuesta\_no\_obs)

Como se hizo la revisión anterior, se calculan los porcentajes de datos faltantes y se revisa nuevamente la distribución de la imputación en las categorías de la variable empleado.

prop.table(table(encuesta$Employment\_missin, useNA = "a"))

| Unemployed | Inactive | Employed | NA |
| --- | --- | --- | --- |
| 0.0340607 | 0.2990953 | 0.4667376 | 0.2001064 |

prop.table(table(encuesta$Employment\_imp, useNA = "a"))

| Unemployed | Inactive | Employed | NA |
| --- | --- | --- | --- |
| 0.0436402 | 0.3650878 | 0.591272 | 0 |

Haciendo el mismo ejercicio por zona se tiene:

kable(  
prop.table( table(encuesta$Zone, encuesta$Employment\_missin, useNA = "a")) %>% addmargins()  
)

|  | Unemployed | Inactive | Employed | NA | Sum |
| --- | --- | --- | --- | --- | --- |
| Rural | 0.0117084 | 0.1506120 | 0.2208622 | 0.1005854 | 0.483768 |
| Urban | 0.0223523 | 0.1484832 | 0.2458755 | 0.0995210 | 0.516232 |
| NA | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.000000 |
| Sum | 0.0340607 | 0.2990953 | 0.4667376 | 0.2001064 | 1.000000 |

kable(  
prop.table( table(encuesta$Zone, encuesta$Employment\_imp,useNA = "a")) %>% addmargins()  
)

|  | Unemployed | Inactive | Employed | NA | Sum |
| --- | --- | --- | --- | --- | --- |
| Rural | 0.0127728 | 0.1873337 | 0.2836615 | 0 | 0.483768 |
| Urban | 0.0308675 | 0.1777541 | 0.3076104 | 0 | 0.516232 |
| NA | 0.0000000 | 0.0000000 | 0.0000000 | 0 | 0.000000 |
| Sum | 0.0436402 | 0.3650878 | 0.5912720 | 0 | 1.000000 |

Al igual que en el caso anterior, la distribución marginal por zona no se altera. Ahora por sexo la distribución es la siguiente:

kable(  
prop.table( table(encuesta$Sex, encuesta$Employment\_missin, useNA = "a")) %>% addmargins()  
)

|  | Unemployed | Inactive | Employed | NA | Sum |
| --- | --- | --- | --- | --- | --- |
| Female | 0.0106440 | 0.2277807 | 0.2011708 | 0.0989888 | 0.5385844 |
| Male | 0.0234167 | 0.0713145 | 0.2655668 | 0.1011176 | 0.4614156 |
| NA | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| Sum | 0.0340607 | 0.2990953 | 0.4667376 | 0.2001064 | 1.0000000 |

kable(  
prop.table( table(encuesta$Sex, encuesta$Employment\_imp,useNA = "a")) %>% addmargins()  
)

|  | Unemployed | Inactive | Employed | NA | Sum |
| --- | --- | --- | --- | --- | --- |
| Female | 0.0159659 | 0.2527940 | 0.2698244 | 0 | 0.5385844 |
| Male | 0.0276743 | 0.1122938 | 0.3214476 | 0 | 0.4614156 |
| NA | 0.0000000 | 0.0000000 | 0.0000000 | 0 | 0.0000000 |
| Sum | 0.0436402 | 0.3650878 | 0.5912720 | 0 | 1.0000000 |

Ahora, haciendo el cálculo del promedio de los ingresos para los datos completos y estimados se tienen los siguientes resultados:

encuesta %>% summarise(  
 Income\_ = mean(Income),  
 Income\_sd = sd(Income),  
 Income\_imp\_ = mean(Income\_imp),  
 Income\_imp\_sd = sd(Income\_imp))

| Income\_ | Income\_sd | Income\_imp\_ | Income\_imp\_sd |
| --- | --- | --- | --- |
| 604.2494 | 513.1078 | 610.505 | 513.6812 |

Se observa que hay una diferencia de 6 unidades monetarias entre el promedio real y el estimado. Realizando este mismo ejercicio por zona tenemos:

encuesta %>%group\_by(Zone) %>% summarise(  
 Income\_ = mean(Income),  
 Income\_sd = sd(Income),  
 Income\_imp\_ = mean(Income\_imp),  
 Income\_imp\_sd = sd(Income\_imp))

| Zone | Income\_ | Income\_sd | Income\_imp\_ | Income\_imp\_sd |
| --- | --- | --- | --- | --- |
| Rural | 469.1217 | 336.5861 | 477.9160 | 344.1316 |
| Urban | 730.8793 | 609.0304 | 734.7559 | 607.0266 |

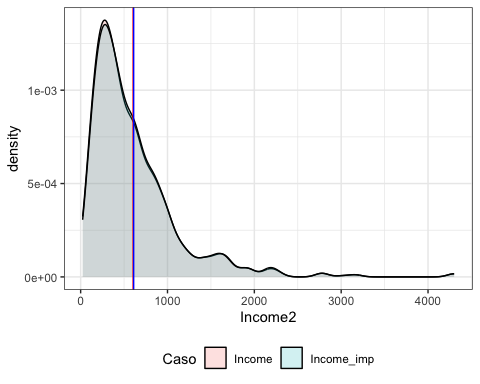
Obteniéndose diferencias de 7 unidades monetarias en el ingreso para la zona rural y de 4 para la urbana. Este mismo ejercicio se realiza por sexo teniendo los siguientes resultados:

encuesta %>%group\_by(Sex) %>% summarise(  
 Income\_ = mean(Income),  
 Income\_sd = sd(Income),  
 Income\_imp\_ = mean(Income\_imp),  
 Income\_imp\_sd = sd(Income\_imp))

| Sex | Income\_ | Income\_sd | Income\_imp\_ | Income\_imp\_sd |
| --- | --- | --- | --- | --- |
| Female | 589.2330 | 504.3041 | 597.8052 | 504.5971 |
| Male | 621.7771 | 522.9428 | 625.3287 | 523.9882 |

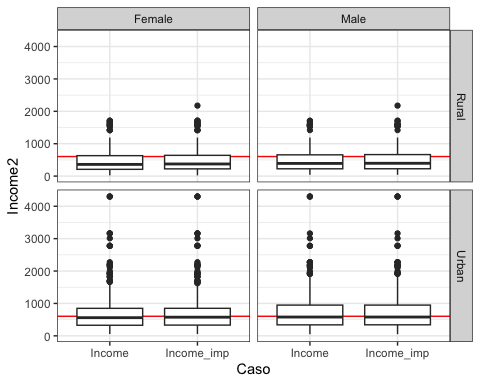
Obteniéndose diferencias pequeñas entre el ingreso real y el estimado en ambos sexos.

dat\_plot8 <- tidyr::gather(  
 encuesta %>% dplyr::select(Zone,Sex,Income, Income\_imp),  
 key = "Caso", value = "Income2", -Zone,-Sex)  
  
p1 <- ggplot(dat\_plot8, aes(x = Income2, fill = Caso)) +   
 geom\_density(alpha = 0.2) + theme\_bw() +  
 theme(legend.position = "bottom") +  
 geom\_vline(  
 xintercept = mean(encuesta$Income),   
 col = "red") +  
 geom\_vline(  
 xintercept = mean(encuesta$Income\_imp),   
 col = "blue")  
p1



Se puede observar que la distribución de los datos imputados son muy próximos que los datos reales. Haciendo este mismo ejercicio pero por zona y sexo se obtienen resultados similares a los anteriores:

p1 <- ggplot(dat\_plot8, aes(x= Caso, y = Income2)) +   
 geom\_hline(yintercept = mean(encuesta$Income),  
 col = "red") + geom\_boxplot() +  
 facet\_grid(Zone~Sex) + theme\_bw()  
p1



## 10.6 Imputación por el vecino más cercano con regresión

An las secciones anteriores se realizaron las descripciones delas técnicas: vecino más cercano e imputación vía regresión, a continuación, se presentan los pasos que se deben tener en cuenta para realizar la imputación utilizando el vecino más cernado mediante una regresión:

* **Paso 1**: Ajustar un modelo de regresión.
* **Paso 2**: Realizar la predicción de los valores observados y no observados.
* **Paso 3**: Comparar las predicciones obtenidas para los valores observados y no observados.
* **Paso 4**: Para la -ésima observación identificar el donante con la menor distancia al receptor.
* **Paso 5**: Reemplazar el valor faltante con la información proveniente del donante.

*NOTA* Se toma es la información observada en el donante.

A continuación, se ejemplifica la técnica imputando los ingresos en los hogares realizando un modelo en el cual se toman como covariables el sexo, la zona y los gastos:

encuesta$Income\_imp <- encuesta$Income\_missin  
encuesta$Employment\_imp <- encuesta$Employment\_missin  
encuesta\_obs <- filter(encuesta, !is.na(Income\_missin))  
encuesta\_no\_obs <- filter(encuesta, is.na(Income\_missin))  
mod <- lm(Income ~ Zone + Sex + Expenditure, data = encuesta\_obs)

Luego, se predicen los valores observados y no observados con el modelo ajustado anteriormente y se imputa el valor faltante calculando las diferencias entre las predicciones de los datos observados y no observados:

pred\_Obs <- predict(mod, encuesta\_obs)  
pred\_no\_Obs <- predict(mod, encuesta\_no\_obs)  
  
for(ii in 1:nrow(encuesta\_no\_obs)){  
  
don\_ii <- which.min(abs(pred\_no\_Obs[ii] - pred\_Obs))  
encuesta\_no\_obs$Income\_imp[[ii]] <- encuesta\_obs$Income\_missin[[don\_ii]]  
encuesta\_no\_obs$Employment\_imp[[ii]] <- encuesta\_obs$Employment\_missin[[don\_ii]]  
}  
  
encuesta <- bind\_rows(encuesta\_obs,encuesta\_no\_obs)

Una vez imputada la información, se puede chequear el pocentaje de datos faltantes que habían y una vez impuatado, cómo cambia la distribución:

kable(  
prop.table(table(encuesta$Employment\_missin, useNA = "a"))  
)

| Var1 | Freq |
| --- | --- |
| Unemployed | 0.0340607 |
| Inactive | 0.2990953 |
| Employed | 0.4667376 |
| NA | 0.2001064 |

kable(  
prop.table(table(encuesta$Employment\_imp, useNA = "a"))  
)

| Var1 | Freq |
| --- | --- |
| Unemployed | 0.0399148 |
| Inactive | 0.3736030 |
| Employed | 0.5864822 |
| NA | 0.0000000 |

El mismo ejercicio se puede realizar por zona:

kable(  
prop.table( table(encuesta$Zone, encuesta$Employment\_missin, useNA = "a")) %>% addmargins()  
)

|  | Unemployed | Inactive | Employed | NA | Sum |
| --- | --- | --- | --- | --- | --- |
| Rural | 0.0117084 | 0.1506120 | 0.2208622 | 0.1005854 | 0.483768 |
| Urban | 0.0223523 | 0.1484832 | 0.2458755 | 0.0995210 | 0.516232 |
| NA | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.000000 |
| Sum | 0.0340607 | 0.2990953 | 0.4667376 | 0.2001064 | 1.000000 |

kable(  
prop.table( table(encuesta$Zone, encuesta$Employment\_imp,useNA = "a")) %>% addmargins()  
)

|  | Unemployed | Inactive | Employed | NA | Sum |
| --- | --- | --- | --- | --- | --- |
| Rural | 0.0138371 | 0.1905269 | 0.2794039 | 0 | 0.483768 |
| Urban | 0.0260777 | 0.1830761 | 0.3070782 | 0 | 0.516232 |
| NA | 0.0000000 | 0.0000000 | 0.0000000 | 0 | 0.000000 |
| Sum | 0.0399148 | 0.3736030 | 0.5864822 | 0 | 1.000000 |

Y por sexo:

kable(  
prop.table( table(encuesta$Sex, encuesta$Employment\_missin, useNA = "a")) %>% addmargins()  
)

|  | Unemployed | Inactive | Employed | NA | Sum |
| --- | --- | --- | --- | --- | --- |
| Female | 0.0106440 | 0.2277807 | 0.2011708 | 0.0989888 | 0.5385844 |
| Male | 0.0234167 | 0.0713145 | 0.2655668 | 0.1011176 | 0.4614156 |
| NA | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| Sum | 0.0340607 | 0.2990953 | 0.4667376 | 0.2001064 | 1.0000000 |

kable(  
prop.table( table(encuesta$Sex, encuesta$Employment\_imp,useNA = "a")) %>% addmargins()  
)

|  | Unemployed | Inactive | Employed | NA | Sum |
| --- | --- | --- | --- | --- | --- |
| Female | 0.0122406 | 0.2730176 | 0.2533262 | 0 | 0.5385844 |
| Male | 0.0276743 | 0.1005854 | 0.3331559 | 0 | 0.4614156 |
| NA | 0.0000000 | 0.0000000 | 0.0000000 | 0 | 0.0000000 |
| Sum | 0.0399148 | 0.3736030 | 0.5864822 | 0 | 1.0000000 |

Por último, se calcula la media de los ingresos y su desviación estándar para los datos completos e imputado como se ha realizado anteriormente:

encuesta %>% summarise(  
 Income\_ = mean(Income),  
 Income\_sd = sd(Income),  
 Income\_imp\_ = mean(Income\_imp),  
 Income\_imp\_sd = sd(Income\_imp))

| Income\_ | Income\_sd | Income\_imp\_ | Income\_imp\_sd |
| --- | --- | --- | --- |
| 604.2494 | 513.1078 | 608.2746 | 515.5829 |

De la anterior imputación se puede observar que, la diferencia entre los datos reales y los imputados es cercano a 4 unidades monetarias.

El mismo ejercicio realizado por zona arroja los siguientes resultados:

encuesta %>%group\_by(Zone) %>% summarise(  
 Income\_ = mean(Income),  
 Income\_sd = sd(Income),  
 Income\_imp\_ = mean(Income\_imp),  
 Income\_imp\_sd = sd(Income\_imp))

| Zone | Income\_ | Income\_sd | Income\_imp\_ | Income\_imp\_sd |
| --- | --- | --- | --- | --- |
| Rural | 469.1217 | 336.5861 | 476.0558 | 342.6870 |
| Urban | 730.8793 | 609.0304 | 732.1786 | 611.0504 |

y se puede observar también que la diferencia entre los ingresos reales y los estimados en las dos zonas con inferiores a 6 unidades monetarias.

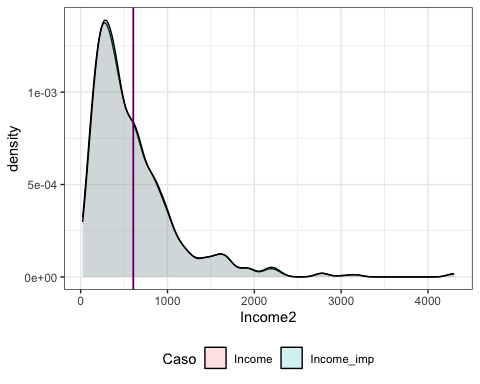
Por sexo, los resultados son los siguientes:

encuesta %>%group\_by(Sex) %>% summarise(  
 Income\_ = mean(Income),  
 Income\_sd = sd(Income),  
 Income\_imp\_ = mean(Income\_imp),  
 Income\_imp\_sd = sd(Income\_imp))

| Sex | Income\_ | Income\_sd | Income\_imp\_ | Income\_imp\_sd |
| --- | --- | --- | --- | --- |
| Female | 589.2330 | 504.3041 | 592.9922 | 508.4371 |
| Male | 621.7771 | 522.9428 | 626.1128 | 523.5303 |

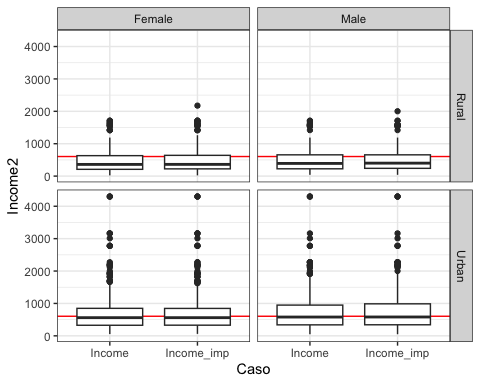
Teniendo como diferencia máxima para la variable ingreso de 5 unidades monetarias. A continuación, se presentan la gráfica distribucional del ingreso real y del ingreso imputado por el método del vecino más cercano mediante un modelo. Se puede observar que, las dos distribuciones con muy similares.

dat\_plot9 <- tidyr::gather(  
 encuesta %>% dplyr::select(Zone,Sex,Income, Income\_imp),  
 key = "Caso", value = "Income2", -Zone,-Sex)  
  
p1 <- ggplot(dat\_plot9, aes(x = Income2, fill = Caso)) +   
 geom\_density(alpha = 0.2) + theme\_bw() +  
 theme(legend.position = "bottom") +  
 geom\_vline(  
 xintercept = mean(encuesta$Income),   
 col = "red") +  
 geom\_vline(  
 xintercept = mean(encuesta$Income\_imp),   
 col = "blue")  
p1



Realizando ahora unos boxplot por sexo y zona para la variable ingreso tanto la completa como la imputada se tiene:

p1 <- ggplot(dat\_plot9, aes(x= Caso, y = Income2)) +   
 geom\_hline(yintercept = mean(encuesta$Income),  
 col = "red") + geom\_boxplot() +  
 facet\_grid(Zone~Sex) + theme\_bw()  
p1

 Se puede observar que los boxplot son muy similares.

## 10.7 Introducción a la imputación múltiple.

La imputación múltiple, también conocida como “multiple imputations” en inglés, es una técnica estadística utilizada para tratar los datos faltantes o incompletos en un conjunto de datos. La imputación múltiple consiste en crear múltiples copias del conjunto de datos, donde los valores faltantes en cada copia son imputados utilizando modelos estadísticos. Estos modelos se basan en las relaciones entre las variables en el conjunto de datos y se utilizan para estimar los valores faltantes de manera plausible.

Luego de que se han creado múltiples copias completas del conjunto de datos, se realizan análisis separados en cada copia para generar resultados. Los resultados de cada análisis se combinan para obtener un único resultado final que refleje la incertidumbre causada por la imputación de los valores faltantes.

La imputación múltiple es una técnica poderosa para lidiar con los datos faltantes, ya que proporciona resultados más precisos y menos sesgados en comparación con otros métodos que simplemente eliminan las observaciones con valores faltantes. Sin embargo, la imputación múltiple es un proceso computacionalmente intensivo y requiere un conocimiento sólido de la teoría estadística para su implementación efectiva.

En este sentido, suponga entonces que existe un conjunto de datos que relaciona dos variables , , a través del siguiente modelo de regresión simple:

Para todo individuo , de tal manera que los errores tienen distribución normal con y .

Ahora bien, se debe tener en cuenta, para utilizar la metodología, los siguientes atributos:

* Sea los valores observados para un conjunto de individuos de tamaño .
* Sea los valores **NO** observados de la variable de tamaño , es decir, .
* Suponga que sí fue posible observar los valores de la covariable para todos los individuos en la muestra.

A modo de ejemplificar la teoría, realicemos un ejercicio de simulación de la siguiente manera. Simular un conjunto de datos con una pendiente y con una dispersión de . A su vez, el conjunto de datos tendrá valores faltantes en la variable respuesta.

El algoritmo de simulación es el siguiente:

generar <- function(n = 500, n\_0 = 200,   
 beta = 10, sigma = 2){  
 x <- runif(n)  
 mu <- beta \* x  
 y <- mu + rnorm(n, mean = 0, sd = sigma)  
 datos <- data.frame(x = x, y = y)  
 faltantes <- sample(n, n\_0)  
 datos$faltantes <- "No"  
 datos$faltantes[faltantes] <- "Si"  
 datos$y.per <- y  
 datos$y.per[faltantes] <- NA  
 return(datos)  
}

El código anterior realiza lo siguiente:

* *generar <- function(n = 500, n\_0 = 200, beta = 10, sigma = 2){*:Esta línea de código define la función “generar” con cuatro argumentos opcionales: n, n\_0, beta y sigma.
* *x <- runif(n)*: Esta línea genera una secuencia de n números aleatorios uniformemente distribuidos en el intervalo [0,1] y los asigna a la variable “x”.
* **mu <- beta \* x**: Esta línea de código calcula la media condicional de “y” (la variable de respuesta) dada “x” utilizando el parámetro de pendiente “beta” y lo asigna a la variable “mu”.
* *y <- mu + rnorm(n, mean = 0, sd = sigma)*: Esta línea genera una variable de respuesta “y” a partir de la media condicional “mu” y agrega un error aleatorio generado a partir de una distribución normal con media cero y desviación estándar “sigma”.
* *datos <- data.frame(x = x, y = y)*: Esta línea combina las variables “x” e “y” en un data frame llamado “datos”.
* *faltantes <- sample(n, n\_0)*: Esta línea genera una muestra aleatoria de “n\_0” valores únicos desde un rango de 1 hasta “n” (la longitud de los datos) y los asigna a la variable “faltantes”.
* *datos$faltantes <- "No"\*, \*datos$faltantes[faltantes] <- “Si”*: Esta línea crea una nueva columna en el data frame “datos” llamada “faltantes” y la inicializa con valores “No” para todas las observaciones. Luego, establece los valores de esta columna en “Si” para las filas seleccionadas por la muestra aleatoria anterior.
* *datosy.per[faltantes] <- NA*: Esta línea crea una nueva columna en el data frame “datos” llamada “y.per” y la inicializa con los mismos valores que “y”. Luego, establece los valores de esta columna en “NA” para las filas seleccionadas por la muestra aleatoria anterior, lo que simula valores faltantes en los datos de “y”.

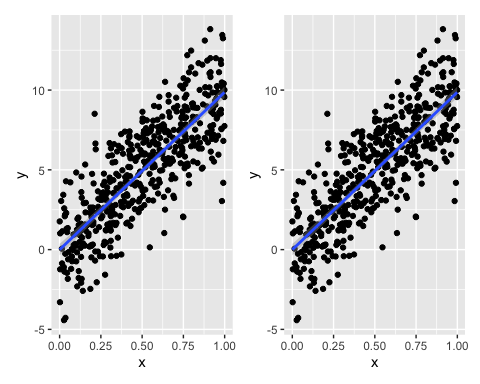
Una vez creada la función anterior, se genera la población usando una semilla para poder reproducirla:

set.seed(1234)  
datos <- generar()  
head(datos,12)

| x | y | faltantes | y.per |
| --- | --- | --- | --- |
| 0.1137034 | 2.0108953 | No | 2.010895 |
| 0.6222994 | 8.3432419 | No | 8.343242 |
| 0.6092747 | 6.9971281 | No | 6.997128 |
| 0.6233794 | 7.5601916 | Si | NA |
| 0.8609154 | 6.3364067 | No | 6.336407 |
| 0.6403106 | 5.6621110 | No | 5.662111 |
| 0.0094958 | 3.0488967 | No | 3.048897 |
| 0.2325505 | -0.1223024 | Si | NA |
| 0.6660838 | 7.1769744 | Si | NA |
| 0.5142511 | 5.9525170 | No | 5.952517 |
| 0.6935913 | 8.8875196 | No | 8.887520 |
| 0.5449748 | 4.7519949 | No | 4.751995 |

A continuación, se grafican la relación entre “x” y “y” y “x” y “y.per” notando que, la distribución entre las dos relaciones es muy similar.

library(patchwork)  
p1 <- ggplot(data = datos, aes(x = x, y = y)) +  
 geom\_point() + geom\_smooth(formula = y~x , method = "lm")  
  
p2 <- ggplot(data = datos, aes(x = x, y = y.per)) +  
 geom\_point() + geom\_smooth(formula = y~x , method = "lm")  
   
p1 | p1



Ahora, dado el 40% de valores faltantes, es necesario imputar dichos valores. Para esto, utilizaremos la técnica de imputación múltiple propuesta por **Rubin (1987)**. La idea consiste en generar conjuntos de valores para los datos faltantes. Al final, el valor *imputado* corresponderá al promedio de esos valores.

Hay varias maneras de realizar la imputación, a continuación, se hace un listado:

* **Ingenua**: Esta clase de imputación carece de aleatoriedad y por tanto, la varianza de va a ser subestimada.
* **Bootstrap**: Se seleccionan muestras bootstrap, y para cada una se estiman los parámetros y para generar . Al final se promedian los valores y se imputa el valor faltante.
* **Bayesiana**: Se definen las distribuciones posteriores de y para generar valores de estos parámetros y por tanto valores de . Al final se promedian los valores y se imputa el valor faltante.

Dado que el interés es la estimación de la pendiente de la regresión lineal simple , entonces la esperanza estimada al utilizar la metodología de imputación múltiple está dada por:

Esta expresión es estimada por el promedio de las estimaciones puntuales de sobre las imputaciones, dado por:

La varianza estimada al utilizar la metodología de imputación múltiple está dada por la siguiente expresión:

La primera parte de la anterior expresión se estima como el promedio de las varianzas muestrales de sobre las imputaciones, dado por:

El segundo término se estima como la varianza muestral de las estimaciones puntuales de sobre las imputaciones, dada por:

Es necesario tener en cuenta un factor de corrección (puesto que es finito). Por tanto, la estimación del segundo término viene dada por la siguiente expresión:

Por tanto, la varianza estimada es igual a:

A modo de ejemplo, a continuación, se crea una función que haga la estimación de los valores faltantes usando bootstrap:

im.bootstrap <- function(datos, M = 15){  
 library(dplyr)  
 n <- nrow(datos)  
 datos1 <- na.omit(datos)  
 n1 <- nrow(datos1)  
 n0 <- n - n1  
 Ind <- is.na(datos$y.per)  
 faltantes.boot <- NULL  
 beta1 <- NULL  
 sigma1 <- NULL  
   
 for (m in 1:M){  
 datos.m <- dplyr::sample\_n(datos1, n1, replace = TRUE)  
 model1 <- lm(y ~ 0 + x, data = datos.m)  
 beta <- model1$coeff  
 sigma <- sqrt(anova(model1)[["Mean Sq"]][2])  
 faltantes.boot <- rnorm(n0, datos$x[Ind] \* beta, sd = sigma)  
 datos$y.per[Ind] <- faltantes.boot  
 model.input <- lm(y.per ~ 0 + x, data = datos)  
 beta1[m] <- model.input$coeff  
 sigma1[m] <- summary(model.input)$coeff[2]  
 }  
 beta.input <- mean(beta1)  
 u.bar <- mean(sigma1 ^ 2)  
 B <- var(beta1)  
 beta.sd <- sqrt(u.bar + B + B/M)  
 result <- list(new = datos, beta = beta.input, sd = beta.sd)  
}

La función anterior realiza internamente lo siguiente:

* En la primera sección del código, desde *library(dplyr)* hasta *sigma1 <- NULL* se calcula el número total de observaciones “n”, se eliminan las filas con valores faltantes y se calculan las longitudes de los datos “datos1” sin valores faltantes, “n1”, y los datos faltantes “n0”. También se crea una variable de índice “Ind” que indica qué observaciones tienen valores faltantes, y se inicializan varias variables que se usarán más adelante en la función.
* Luego, en la sección del bucle *for* se realiza el proceso de imputación múltiple. Se itera “M” veces, donde en cada iteración se muestran “n1” filas de los datos sin valores faltantes “datos1” con reemplazo, y se ajusta un modelo de regresión lineal con “y” como respuesta y “x” como variable explicativa. Luego se calculan los coeficientes de la regresión y la desviación estándar de los residuos. A continuación, se generan valores de imputación aleatorios para las filas faltantes, utilizando la media condicional y la desviación estándar estimadas en la muestra. Estos valores de imputación se asignan a las filas faltantes en el data frame “datos”. Finalmente, se ajusta un modelo de regresión lineal con la variable de respuesta imputada “y.per” como respuesta y “x” como variable explicativa. Se guardan los coeficientes y las desviaciones estándar de los residuos para cada iteración en las variables “beta1” y “sigma1”, respectivamente.
* En la última sección del código se calcula el coeficiente de regresión promedio para las iteraciones, “beta.input”, y se calcula la desviación estándar de “beta1” y por último, la estimación de la desviación estándar del beta.

Al aplicar la función sobre el conjunto de datos creado, se obtienen las siguientes salidas:

datos <- generar()  
im.bootstrap(datos)$beta

## [1] 10.29741

im.bootstrap(datos)$sd

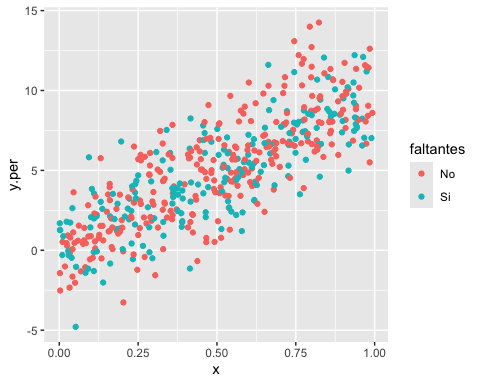
## [1] 0.2222246

head(im.bootstrap(datos)$new)

| x | y | faltantes | y.per |
| --- | --- | --- | --- |
| 0.2173207 | 0.2872457 | Si | 0.4423490 |
| 0.2953090 | 1.5860882 | Si | 1.8694104 |
| 0.9609104 | 11.2309918 | Si | 11.8596299 |
| 0.3119669 | 5.0508837 | No | 5.0508837 |
| 0.0520595 | -0.7714404 | Si | -0.5380997 |
| 0.1597426 | 3.9523667 | No | 3.9523667 |

Nótese que existe una buena dispersión en los valores imputados.

nuevos <- im.bootstrap(datos)$new  
ggplot(data = nuevos, aes(x = x, y = y.per, color = faltantes)) + geom\_point()



Por otro lado, se ejemplificará la técnica de imputación múltiple para los datos de la encuesta que se utiliza de ejemplo en este texto:

encuesta$Income\_imp <- encuesta$Income\_missin  
encuesta$Employment\_imp <- encuesta$Employment\_missin  
encuesta\_obs <- filter(encuesta, !is.na(Income\_missin))  
encuesta\_no\_obs <- filter(encuesta, is.na(Income\_missin))  
n0 <- nrow(encuesta\_no\_obs)  
n1 <- nrow(encuesta\_obs)

Inicialmente, se extraen los datos a imputar y se calculan los tamaños de los datos observados y no observados.

M = 10   
set.seed(1234)  
for (ii in 1:M) {  
vp <- paste0("Income\_vp\_",ii)  
vp2 <- paste0("Employment\_vp\_",ii)  
  
encuesta\_temp <- encuesta\_obs %>% sample\_n(size = n1, replace = TRUE)  
  
mod <- lm(Income~ Zone + Sex + Expenditure, data = encuesta\_temp)  
mod.mult <- multinom(Employment~Zone + Sex +Expenditure,data = encuesta\_temp)  
  
encuesta\_no\_obs[[vp]] <- predict(mod, encuesta\_no\_obs)  
encuesta\_obs[[vp]] <- encuesta\_obs$Income  
  
encuesta\_no\_obs[[vp2]] <- predict(mod.mult, encuesta\_no\_obs,type = "class")  
encuesta\_obs[[vp2]] <- encuesta\_obs$Employment  
}

## # weights: 15 (8 variable)  
## initial value 1651.214270   
## iter 10 value 1147.241384  
## final value 1127.485787   
## converged  
## # weights: 15 (8 variable)  
## initial value 1651.214270   
## iter 10 value 1177.374646  
## final value 1161.652154   
## converged  
## # weights: 15 (8 variable)  
## initial value 1651.214270   
## iter 10 value 1104.676334  
## final value 1090.350507   
## converged  
## # weights: 15 (8 variable)  
## initial value 1651.214270   
## iter 10 value 1157.667055  
## final value 1138.773189   
## converged  
## # weights: 15 (8 variable)  
## initial value 1651.214270   
## iter 10 value 1145.090973  
## final value 1094.863541   
## converged  
## # weights: 15 (8 variable)  
## initial value 1651.214270   
## iter 10 value 1139.350146  
## final value 1127.134266   
## converged  
## # weights: 15 (8 variable)  
## initial value 1651.214270   
## iter 10 value 1169.810267  
## final value 1141.798357   
## converged  
## # weights: 15 (8 variable)  
## initial value 1651.214270   
## iter 10 value 1165.750154  
## final value 1125.933563   
## converged  
## # weights: 15 (8 variable)  
## initial value 1651.214270   
## iter 10 value 1117.370321  
## final value 1093.345603   
## converged  
## # weights: 15 (8 variable)  
## initial value 1651.214270   
## iter 10 value 1143.115259  
## final value 1129.053716   
## converged

El código anterior realiza lo siguiente:

* Se crea una cadena de caracteres “vp” y “vp2” mediante la función *paste0()* concatenando la cadena “Income\_vp\_” y “Employment\_vp\_” respectivamente con el número de la iteración actual “ii”.
* Se crea una muestra aleatoria con reemplazo de tamaño n1 a partir de la base de datos “encuesta\_obs” utilizando la función *sample\_n()* del paquete *dplyr*.
* Se ajusta un modelo de regresión lineal con la variable “Income” como respuesta y las variables “Zone”, “Sex” y “Expenditure” como covariables utilizando la función *lm()*.
* Se ajusta un modelo de regresión multinomial con la variable “Employment” como respuesta y las variables “Zone”, “Sex” y “Expenditure” como covariables utilizando la función *multinom()* del paquete *nnet*.
* Se utiliza el modelo ajustado en el punto 3 para predecir los valores de la variable “Income” en la base de datos “encuesta\_no\_obs” mediante la función *predict()*.
* Se guarda en la base de datos “encuesta\_obs” los valores verdaderos de la variable “Income”.
* Se utiliza el modelo ajustado en el punto 4 para predecir los valores de la variable “Employment” en la base de datos “encuesta\_no\_obs” mediante la función *predict()*.
* Se guarda en la base de datos “encuesta\_obs” los valores verdaderos de la variable “Employment”.
* Se repite el proceso para las siguientes iteraciones.
* Al finalizar el bucle se obtendrán 20 nuevas variables en las bases de datos “encuesta\_no\_obs” y “encuesta\_obs” correspondientes a las predicciones de “Income” y “Employment” respectivamente, para cada una de las 10 iteraciones del bucle.

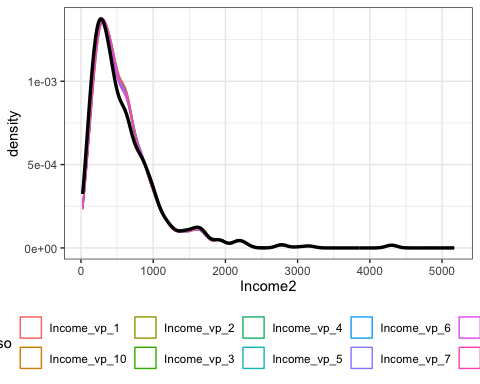
Una vez corrido el código anterior, se seleccionan las variables de ingresos y sus 10 valores plausibles como se muestra a continuación:

dplyr::select(encuesta\_no\_obs, Income, matches("Income\_vp\_"))[1:10,1:4]

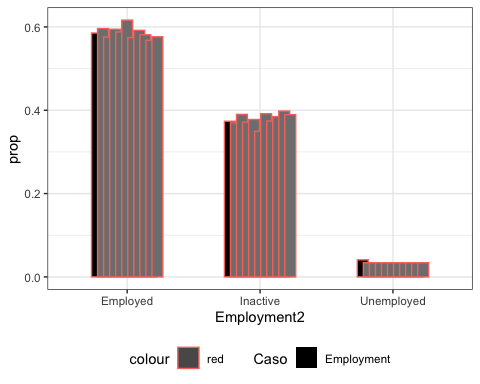
| Income | Income\_vp\_1 | Income\_vp\_2 | Income\_vp\_3 |
| --- | --- | --- | --- |
| 409.87 | 550.2195 | 566.0432 | 567.8296 |
| 409.87 | 561.1194 | 529.3457 | 541.8310 |
| 90.92 | 210.5682 | 225.7715 | 163.9913 |
| 90.92 | 221.4682 | 189.0739 | 137.9927 |
| 90.92 | 210.5682 | 225.7715 | 163.9913 |
| 135.33 | 222.6937 | 237.9191 | 178.4083 |
| 135.33 | 222.6937 | 237.9191 | 178.4083 |
| 1539.75 | 784.8579 | 801.1103 | 846.8098 |
| 336.00 | 507.8955 | 472.7913 | 439.7467 |
| 685.48 | 593.0111 | 558.0623 | 540.9473 |

A continuación, se grafica la distribución de los ingresos y los 10 valores plausibles observándose que, las distribuciones son muy similares:

encuesta <- bind\_rows(encuesta\_obs, encuesta\_no\_obs)  
  
dat\_plot10 <- tidyr::gather(  
 encuesta %>% dplyr::select(Zone,Sex,matches("Income\_vp\_")),  
 key = "Caso", value = "Income2", -Zone,-Sex)  
  
p1 <- ggplot(dat\_plot10, aes(x = Income2, col = Caso)) +   
 geom\_density(alpha = 0.2) + theme\_bw() +  
 theme(legend.position = "bottom") + geom\_density(data = encuesta ,aes(x = Income), col = "black", size = 1.2)   
  
p1

 En el siguiente gráfico se presentan las frecuencias de los valores plausibles para la variable empleado. Se observa también que son muy aproximadas a la variable completa:

dat\_plot11 <- tidyr::gather(  
 encuesta %>%   
 dplyr::select(Zone,Sex, Employment,matches("Employment\_vp\_")),  
 key = "Caso", value = "Employment2", -Zone,-Sex) %>%  
 group\_by(Caso,Employment2) %>% tally() %>%   
 group\_by(Caso) %>% mutate(prop = n/sum(n))  
  
p1 <- ggplot(dat\_plot11,   
 aes(x = Employment2, y = prop,  
 fill = Caso, color="red")) +   
 geom\_bar(stat="identity",  
 position = position\_dodge(width = 0.5)) +  
 theme\_bw() +  
 theme(legend.position = "bottom") +  
 scale\_fill\_manual(values = c("Employment" = "black"))  
p1

 Con los valores plausibles enocntrados anteriormente, se procede a definir el diseño muestral utilizado en este ejemplo y así poder hacer la estimación de los parámetros. A continuación, se define el diseño muestral:

library(srvyr)  
  
diseno <- encuesta %>%  
 as\_survey\_design(  
 strata = Stratum,  
 ids = PSU,  
 weights = wk,  
 nest = T  
 )

Con el diseño anterior, se estiman los ingresos medios para cada valor plausible junto ocn su varianza, como se muestra a continuación:

estimacion\_vp <- diseno %>%   
 summarise(  
 vp1 = survey\_mean(Income\_vp\_1, vartype = c("var")),  
 vp2 = survey\_mean(Income\_vp\_2, vartype = c("var")),  
 vp3 = survey\_mean(Income\_vp\_3, vartype = c("var")),  
 vp4 = survey\_mean(Income\_vp\_4, vartype = c("var")),  
 vp5 = survey\_mean(Income\_vp\_5, vartype = c("var")),  
 vp6 = survey\_mean(Income\_vp\_6, vartype = c("var")),  
 vp7 = survey\_mean(Income\_vp\_7, vartype = c("var")),  
 vp8 = survey\_mean(Income\_vp\_8, vartype = c("var")),  
 vp9 = survey\_mean(Income\_vp\_9, vartype = c("var")),  
 vp10 =survey\_mean(Income\_vp\_10, vartype = c("var")))  
estimacion\_vp

| vp1 | vp1\_var | vp2 | vp2\_var | vp3 | vp3\_var | vp4 | vp4\_var | vp5 | vp5\_var | vp6 | vp6\_var | vp7 | vp7\_var | vp8 | vp8\_var | vp9 | vp9\_var | vp10 | vp10\_var |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 619.3525 | 845.706 | 617.6058 | 844.5521 | 617.4322 | 867.5133 | 617.748 | 856.7732 | 619.9729 | 857.805 | 617.1098 | 852.6605 | 618.3378 | 860.7981 | 619.2779 | 867.9485 | 616.9401 | 850.131 | 615.7853 | 870.0434 |

A continuación se presentan los datos anteriores discriminado por promedio y varianza:

require(tidyr)  
(estimacion\_vp %<>% tidyr::gather() %>% separate(key, c("vp", "estimacion")) %>%   
mutate(estimacion = ifelse(is.na(estimacion), "promedio","var" )) %>% spread(estimacion,value) %>% mutate(vp = 1:10))

| vp | promedio | var |
| --- | --- | --- |
| 1 | 619.3525 | 845.7060 |
| 2 | 615.7853 | 870.0434 |
| 3 | 617.6058 | 844.5521 |
| 4 | 617.4322 | 867.5133 |
| 5 | 617.7480 | 856.7732 |
| 6 | 619.9729 | 857.8050 |
| 7 | 617.1098 | 852.6605 |
| 8 | 618.3378 | 860.7981 |
| 9 | 619.2779 | 867.9485 |
| 10 | 616.9401 | 850.1310 |

Por último, para obtener la estimación de la media y su varianza utilizando la imputación múltiple, se realizan los siguientes cálculos que se derivan de las expresiones matemáticas antes mostradas:

Media\_vp = mean(estimacion\_vp$promedio)  
(Ubar = mean(estimacion\_vp$var))

## [1] 857.3931

(B = var(estimacion\_vp$promedio))

## [1] 1.645758

var\_vp = Ubar + (1 + 1/M)   
(resultado <- data.frame(Media\_vp,   
 Media\_vp\_se = sqrt(var\_vp)))

| Media\_vp | Media\_vp\_se |
| --- | --- |
| 617.9562 | 29.30005 |

estimacion\_var\_vp <- diseno %>%   
 summarise\_at(vars(matches("Income\_vp")),   
 survey\_var, vartype = "var" )

Por otro lado, otro parámetro de interés es la varianza de los ingresos. Este parámetro permite medir la variabilidad de los ingresos de los ciudadanos de la base de datos de ejemplo. La forma de estimarla es la misma que para el promedio de los ingresos y se utilizarán los mismo códigos mostrados anteriormente, cambiando el parámetro a estimar:

(estimacion\_var\_vp %<>% tidyr::gather() %>%  
 separate(key, c("A", "B","vp", "estimacion")) %>%   
mutate(estimacion = ifelse(is.na(estimacion), "promedio","var" ),  
 A = NULL, B = NULL, vp = as.numeric(vp)) %>%   
 spread(estimacion,value))

| vp | promedio | var |
| --- | --- | --- |
| 1 | 262689.8 | 3074674460 |
| 2 | 263092.0 | 3079895581 |
| 3 | 274370.1 | 3237991671 |
| 4 | 269127.4 | 3164842634 |
| 5 | 270450.0 | 3165285304 |
| 6 | 264992.6 | 3106582542 |
| 7 | 270916.0 | 3175865575 |
| 8 | 276069.5 | 3251836316 |
| 9 | 265060.8 | 3111426636 |
| 10 | 275462.0 | 3258214990 |

Por último, se utilizan las ecuaciones mostradas anteriormente:

Media\_var\_vp = mean(estimacion\_var\_vp$promedio)  
Ubar = mean(estimacion\_var\_vp$var)  
B = var(estimacion\_var\_vp$promedio)  
var\_var\_vp = Ubar + (1 + 1/M)\*B   
resultado$var\_vp <- Media\_var\_vp  
resultado$var\_vp\_se <- sqrt(var\_var\_vp)  
cbind(Media\_var\_vp, var\_var\_vp)

| Media\_var\_vp | var\_var\_vp |
| --- | --- |
| 269223 | 3191037330 |

Otro parámetro de interés a estimar es la proporción. A continuación, se realizará la estimación de la proporción utilizando valores plausibles. Para ello se estimará la variable empleado, como se muestra a continuación:

estimacion\_prop\_vp <- lapply(paste0("Employment\_vp\_",1:10),  
 function(vp){diseno %>% group\_by\_at(vars(Employment = vp)) %>%   
 summarise(prop = survey\_mean(vartype = "var"),.groups = "drop") %>%  
 mutate(vp = vp)}) %>% bind\_rows()

Se presenta la estimación de la proporción para cada uno de los 10 valores plausibles en cada categoría de la variable:

(estimacion\_prop\_vp %<>% separate(vp, c("A", "B","vp")) %>%   
mutate(A = NULL, B = NULL, vp = as.numeric(vp)) %>%  
 dplyr::select(vp,Employment:prop\_var)) %>% slice(1:12L)

| vp | Employment | prop | prop\_var |
| --- | --- | --- | --- |
| 1 | Unemployed | 0.0355675 | 0.0000375 |
| 1 | Inactive | 0.3753767 | 0.0002041 |
| 1 | Employed | 0.5890558 | 0.0001669 |
| 2 | Unemployed | 0.0355675 | 0.0000375 |
| 2 | Inactive | 0.3771091 | 0.0002035 |
| 2 | Employed | 0.5873234 | 0.0001661 |
| 3 | Unemployed | 0.0355675 | 0.0000375 |
| 3 | Inactive | 0.3867965 | 0.0002056 |
| 3 | Employed | 0.5776360 | 0.0001790 |
| 4 | Unemployed | 0.0355675 | 0.0000375 |
| 4 | Inactive | 0.3549487 | 0.0001915 |
| 4 | Employed | 0.6094838 | 0.0001524 |

Por último, utilizando las ecuaciones de Rubin se obtiene la varianza estimada:

resultado = estimacion\_prop\_vp %>%   
 group\_by(Employment) %>%   
 summarise(prop\_pv = mean(prop),  
 Ubar = mean(prop\_var),  
 B = var(prop)) %>%   
 mutate(prop\_pv\_var = Ubar + (1 + 1/M)\*B) |>   
 dplyr::select(Employment, prop\_pv, prop\_pv\_var)  
resultado

| Employment | prop\_pv | prop\_pv\_var |
| --- | --- | --- |
| Unemployed | 0.0355675 | 0.0000375 |
| Inactive | 0.3868270 | 0.0004665 |
| Employed | 0.5776055 | 0.0004328 |

# 11 Referencias

* Asparouhov, T., & Muthen, B. (2006, August). Multilevel modeling of complex survey data. In Proceedings of the joint statistical meeting in Seattle (pp. 2718-2726).
* Asparouhov, T. (2006). General multi-level modeling with sampling weights. Communications in Statistics—Theory and Methods, 35(3), 439-460.
* Pfeffermann, D., Skinner, C. J., Holmes, D. J., Goldstein, H., & Rasbash, J. (1998). Weighting for unequal selection probabilities in multilevel models. Journal of the Royal Statistical Society: series B (statistical methodology), 60(1), 23-40.
* Cai, T. (2013). Investigation of ways to handle sampling weights for multilevel model analyses. Sociological Methodology, 43(1), 178-219.
* Finch, W. H., Bolin, J. E., & Kelley, K. (2019). Multilevel modeling using R. Crc Press.
* Merlo, J., Chaix, B., Ohlsson, H., Beckman, A., Johnell, K., Hjerpe, P., … & Larsen, K. (2006). A brief conceptual tutorial of multilevel analysis in social epidemiology: using measures of clustering in multilevel logistic regression to investigate contextual phenomena. Journal of Epidemiology & Community Health, 60(4), 290-297.
* Sarndal, C., Swensson, B. &Wretman, J. (1992), Model Assisted Survey Sampling, Springer, New York.
* Rojas, H. A. G. (2016). Estrategias de muestreo: diseño de encuestas y estimación de parámetros. Ediciones de la U.
* Santana Sepúlveda, S., & Mateos Farfán, E. (2014). El arte de programar en R: un lenguaje para la estadística.
* Lumley, T. (2011). Complex surveys: a guide to analysis using R. John Wiley & Sons.
* Bache, S. M., Wickham, H., Henry, L., & Henry, M. L. (2022). Package ‘magrittr’.
* Tellez Piñerez, C. F., & Lemus Polanía, D. F. (2015). Estadística Descriptiva y Probabilidad con aplicaciones en R. Fundación Universitaria Los Libertadore.
* Groves, R. M., Fowler Jr, F. J., Couper, M. P., Lepkowski, J. M., Singer, E., & Tourangeau, R. (2011). Survey methodology. John Wiley & Sons.
* Tille, Y. & Ardilly, P. (2006), Sampling Methods: Exercises and Solutions, Springer.
* Gambino, J. G., & do Nascimento Silva, P. L. (2009). Sampling and estimation in household surveys. In Handbook of Statistics (Vol. 29, pp. 407-439). Elsevier.
* Cochran, W. G. (1977) *Sampling Techniques*. John Wiley and Sons.
* Gutiérrez, H. A. (2017) TeachingSampling. *R package*.
* Wickham, H., Chang, W., & Wickham, M. H. (2016). Package ‘ggplot2’. Create elegant data visualisations using the grammar of graphics. Version, 2(1), 1-189.
* Lumley, T. (2020). Package ‘survey’. Available at the following link: <https://cran>. r-project. org.
* Hansen, M. H., & Steinberg, J. (1956). Control of errors in surveys. Biometrics, 12(4), 462-474.
* Heeringa, S. G., West, B. T., & Berglund, P. A. (2017). Applied survey data analysis. chapman and hall/CRC.
* Valliant, R., Dever, J.A., and Kreuter, F., Practical Tools for Designing and Weighting Survey Samples, Springer, New York, 2013.
* Valliant, R., Dorfman, A.H., and Royall, R.M., Finite Population Sampling and Inference: A Prediction Approach, John Wiley & Sons, New York, 2000.
* Loomis, D., Richardson, D.B., and Elliott, L., Poisson regression analysis of ungrouped data, Occupational and Environmental Medicine, 62, 325–329, 2005.
* Kovar, J.G., Rao, J.N.K., and Wu, C.F.J., Bootstrap and other methods to measure errors in survey estimates, Canadian Journal of Statistics, 16(Suppl.), 25–45, 1988.
* Binder, D.A. and Kovacevic, M.S., Estimating some measures of income inequality from survey data: An application of the estimating equations approach, Survey Methodology, 21(2), 137–145, 1995.
* Kovacevic, M. S., & Binder, D. A. (1997). Variance estimation for measures of income inequality and polarization-the estimating equations approach. Journal of Official Statistics, 13(1), 41.
* Bautista, J. (1998), Diseños de muestreo estadístico, Universidad Nacional de Colombia.
* Monroy, L. G. D., Rivera, M. A. M., & Dávila, L. R. L. (2018). Análisis estadístico de datos categóricos. Universidad Nacional de Colombia.
* Kish, L. and Frankel, M.R., Inference from complex samples, Journal of the Royal Statistical Society, Series B, 36, 1–37, 1974.
* Fuller, W.A., Regression analysis for sample survey, Sankyha, Series C, 37, 117–132, 1975.
* Shah, B.V., Holt, M.M., and Folsom, R.F., Inference about regression models from sample survey data, Bulletin of the International Statistical Institute, 41(3), 43–57, 1977.
* Skinner, C.J., Holt, D., and Smith, T.M.F., Analysis of Complex Surveys, John Wiley & Sons, New York, 1989.
* Binder, D.A., On the variances of asymptotically normal estimators from complex surveys, International Statistical Review, 51, 279–292, 1983.
* Fuller, W.A., Regression estimation for survey samples (with discussion), Survey Methodology, 28(1), 5–23, 2002.
* Pfeffermann, D., Modelling of complex survey data: Why model? Why is it a problem? How can we approach it? Survey Methodology, 37(2), 115–136, 2011.
* Wolter, K.M., Introduction to Variance Estimation (2nd ed.), Springer-Verlag, New York, 2007.
* Tellez, C. F., & Morales, M. A. (2016). Modelos Estadísticos lineales con aplicaciones en R. Ediciones de la U.
* Fay, R.E., On adjusting the Pearson Chi-square statistic for cluster sampling, In Proceedings of the Social Statistics Section, American Statistical Association, Washington, DC, 402–405, 1979.
* Fay, R.E., A jack-knifed chi-squared test for complex samples, Journal of the American Statistical Association, 80, 148–157, 1985.
* Fellegi, I.P., Approximate tests of independence and goodness of fit based on stratified multistage samples, Journal of the American Statistical Association, 75, 261–268, 1980.
* Thomas, D.R. and Rao, J.N.K., Small-sample comparisons of level and power for simple goodness-of-fit statistics under cluster sampling, Journal of the American Statistical Association, 82, 630–636, 1987.
* Rao, J.N.K. and Scott, A.J., On chi-squared test for multiway contingency tables with cell proportions estimated from survey data, The Annals of Statistics, 12, 46–60, 1984.
* Van Buuren, S., Flexible Imputation of Missing Data, Chapman & Hall, Boca Raton, FL, 2012.
* Carpenter, J.R. and Kenward, M.G., Multiple Imputation and Its Application, John Wiley & Sons, Chichester, West Sussex, UK, 2013.
* Berglund, P.A. and Heeringa, S.G., Multiple Imputation of Missing Data Using SAS®, SAS Institute Inc., Cary, NC, 2014.
* Chambers, R.L., Steel, D.G., Wang, S., and Welsh, A.H., Maximum Likelihood Estimation for Sample Surveys, Chapman & Hall, Boca Raton, FL, 2012.
* Zhou, H., Elliott, M.R., and Raghunathan, T.E., Multiple imputation in two-stage cluster samples using the weighted finite population Bayesian Bootstrap, Journal of Survey Statistics and Methodology, 4, 139–170, 2016a.
* Zhou, H., Elliott, M.R., and Raghunathan, T.E., Synthetic multiple-imputation procedure for multistage complex samples, Journal of Official Statistics, 32(1), 231–256, 2016b.
* Kim, J.K. and Shao, J., Statistical Methods for Handling Incomplete Data, Chapman & Hall, Boca Raton, FL, 2014.
* Kim, J.K. and Fuller, W.A., Fractional Hotdeck imputation, Biometrika, 89, 470–477, 2004.
* StataCorp., Release 14, P Manual, STATA Survey Data Manual, Stata Press, College Station, TX, 2015.
* Raghunathan, T.E., Missing Data Analysis in Practice, Chapman & Hall/CRC Interdisciplinary Statistics, Boca Raton, FL, 2016.
* Rubin, D. B. (1987). Multiple imputation for survey nonresponse.
* Goldstein, H. (2011). Multilevel statistical models (Vol. 922). John Wiley & Sons.
* Data analysis using regression and multilevel/hierarchical models” de Andrew Gelman y Jennifer Hill (2006)
* Sophia, R. H., & Skrondal, A. (2012). Multilevel and longitudinal modeling using Stata. STATA press.
* Browne, W. J., & Draper, D. (2006). A comparison of Bayesian and likelihood-based methods for fitting multilevel models.

1. Experto Regional en Estadísticas Sociales - Comisión Económica para América Latina y el Caribe (CEPAL) - [andres.gutierrez@cepal.org](mailto:andres.gutierrez@cepal.org) [↑](#footnote-ref-20)
2. Consultor - Comisión Económica para América Latina y el Caribe (CEPAL), [guerrerostalyn@gmail.com](mailto:guerrerostalyn@gmail.com) [↑](#footnote-ref-22)
3. Consultor - Comisión Económica para América Latina y el Caribe (CEPAL), [cftellezp@unal.edu.co](mailto:cftellezp@unal.edu.co) [↑](#footnote-ref-24)
4. Consultor - Comisión Económica para América Latina y el Caribe (CEPAL), [gbabativam@gmail.com](mailto:gbabativam@gmail.com) [↑](#footnote-ref-26)
5. <https://tech.popdata.org/pma-data-hub/posts/2021-08-15-covid-analysis/> [↑](#footnote-ref-341)
6. Cai, T. (2013). Investigation of ways to handle sampling weights for multilevel model analyses. Sociological Methodology, 43(1), 178-219. [↑](#footnote-ref-357)