Household Surveys Analysis with R

Andrés Gutiérrez[1], Stalyn Guerrero[2], Giovany Babativa-Márquez[3], Cristian Téllez[4]

2026-06-12

[1]Regional Adviser on Social Statistics - Economic Commission for Latin America and the Caribbean (ECLAC) - andres.gutierrez@cepal.org

[2]Consultant - Economic Commission for Latin America and the Caribbean (ECLAC), guerrerostalyn@gmail.com

[3]Associate Professor - Department of Statistics, Faculty of Sciences, National University of Colombia, Bogotá Campus - jgbabativam@unal.edu.co

[4]Consultant - Economic Commission for Latin America and the Caribbean (ECLAC), cftellezp@unal.edu.co

Table of Contents

List of Figures

List of Tables

Preface

You can also download the PDF version here: Download the book as PDF.

This book brings together practical international experience acquired by the authors while advising Latin American and Caribbean countries through the Statistics Division of ECLAC.

Summary

Household surveys are the main source of quantitative information for measuring social and economic indicators, such as poverty, inequality, employment, education, and many others, in Latin America and worldwide. They are used to estimate indicators that guide the design, evaluation, and monitoring of public policies, and they are also a primary source of information for monitoring the Sustainable Development Goals within the framework of the United Nations 2030 Agenda. The Economic Commission for Latin America and the Caribbean (ECLAC) manages microdata repositories, including the Household Survey Data Bank (BADEHOG), which compile, harmonize, and document surveys from more than 18 countries in the region since 2000, precisely because the information produced by these sources is essential for evidence-based decision-making.

However, behind this apparently democratized access to data lies a structural technical issue that often goes unnoticed: most household surveys do not come from simple random samples. On the contrary, they are based on complex sampling designs that include stratification, multiple stages, unequal selection probabilities, and calibration-based weighting adjustments. When an analyst applies traditional statistical methods designed for independent and identically distributed samples to data that do not have those characteristics, the consequences include biased estimates, underestimated standard errors, overly narrow confidence intervals, and hypothesis tests with inflated type I error rates.

As an illustrative example, consider a population with two regions of very different sizes and very different average incomes. A survey that selects the same number of people in each region and analyzes the data without weights will produce a structurally biased estimator of national mean income, because units in the smaller region have greater empirical weight in the sample than their actual weight in the population. This error also propagates to the variance. An analyst who computes the variance while ignoring the cluster structure will treat households in the same census sector as independent observations, when in fact they share common characteristics that reduce the effective information in the sample.

This document seeks to close that gap and provide researchers with the theoretical foundations and computational tools needed to work properly with surveys drawn from complex designs. Throughout the text, practical aspects are addressed in an integrated way, ranging from the definition of the sampling design to the estimation of indicators, such as totals, counts, means, and proportions, the construction of their confidence intervals, the formulation of hypothesis tests, modeling through advanced regression techniques, and the treatment of common imputation problems.

Chapter 1

Introduction

The analysis of surveys with complex designs has a well-established tradition in the statistical literature, grounded in the design-based inference approach. As noted by Särndal et al. [2003] and Gutiérrez [2016], this approach recognizes the probability distribution induced by the sample selection process as the only probabilistic mechanism governing inference. Under this paradigm, the statistical properties of estimators, such as unbiasedness, precision, and consistency, are evaluated with respect to the distribution of all possible samples that the design can generate, without imposing distributional assumptions on the variable of interest.

Within this framework, variance estimation has played a central role and has evolved around three broad complementary methodological families. The first, the ultimate cluster method [Hansen et al., 1953], simplifies the multilevel structure by concentrating variability in the primary sampling units (PSUs), treating selection as if it were carried out with replacement. The second, Taylor linearization [Binder, 1983], extends this principle to nonlinear parameters, such as means, proportions, and ratios, through a first-order approximation that reduces the variance estimation problem to that of a total. Finally, the third family, replication methods such as jackknife, bootstrap, and balanced repeated replication, estimates variance from the dispersion among estimates obtained through systematic sample replicates.

In parallel, in the field of statistical modeling, foundational contributions such as those by Kish and Frankel [1974], Fuller [1975], and Binder [1983] laid the groundwork for weighted regression and variance estimation for coefficients under stratified and multistage designs. Building on this development, Pfeffermann [1993] formalized the maximum pseudo-likelihood (MPL) method, which incorporates sampling weights into the log-likelihood function to produce design-consistent estimators, thereby extending the generalized linear models of Nelder and Wedderburn [1972] to the context of surveys with complex designs.

More recently, Heeringa et al. [2017] have synthesized these contributions within a unified analytical framework that coherently integrates the sampling design, weighting, and modeling, establishing their work as a standard reference for contemporary practice in complex survey analysis.

This document is intended for readers with training in statistics or data science, or with equivalent professional experience. It is especially designed for those who have knowledge of the foundations of sampling theory, such as inclusion probabilities, estimators, and confidence intervals, of linear and logistic regression models, and of introductory programming in R.

In this context, the document is positioned as an intermediate-to-advanced text aimed at

deepening and consolidating this knowledge, facilitating its rigorous application in complex and technically demanding settings. Its purpose is to accompany researchers in the transition from theoretical foundations to their proper implementation in the production of official statistics, academic research in the social sciences, public policy evaluation, and data analysis under international quality standards.

I Rationale

Despite the methodological richness available in the specialized literature, a significant gap persists between the state of the art and the everyday practice of survey analysis in the region. One of the main manifestations of this gap is related to access to and use of statistical software. Several widely recognized reference texts in survey analysis, such as Heeringa et al. [2017], Cochran [1977], and Särndal et al. [2003], are fundamental from a theoretical perspective; however, they are mainly oriented toward licensed software such as `SAS`, `Stata`, or `SPSS`, or they do not fully integrate with the `R` computational ecosystem. In contrast, many online tutorials on survey analysis in `R` tend to privilege the operational aspects of the software, but rarely reach the conceptual depth needed for appropriate and responsible use in the production of official statistics.

In addition, there is substantial methodological fragmentation among the different components involved in complex survey analysis. The foundations of sampling design, computational data management, estimation of descriptive and inferential parameters with design correction, and treatment of nonresponse are often addressed separately, making it difficult to build integrated and reproducible workflows. In this context, the present text proposes a coherent integration of these elements within a single analytical framework implemented entirely in `R`.

Another relevant aspect is the regional contextualization of the examples and applications. A considerable share of reference texts use databases and examples from surveys conducted in English-speaking countries, which partially limits their immediate applicability in Latin America. To bring the content closer to the statistical realities of the region, this document uses a native `R` database designed to replicate the traditional structure of Latin American household surveys and therefore to provide examples closer to the problems faced by national statistical offices and technical teams in the region.

Finally, this document is itself an exercise in reproducible research. All of its content, including tables, figures, equations, and numerical results, is generated entirely through R code embedded directly in the chapters. In this way, readers can not only study the methods presented, but also replicate, modify, and adapt them to their own contexts and databases, thereby strengthening analytical capacity and transparency in statistical production.

II Use of R Software for Survey Analysis

The choice of R as the computational environment for this document is not arbitrary. R has at least three characteristics that make it an ideal option for household survey analysis in the Latin American context.

First, it is freely accessible. Unlike other programs with licensing costs that may be prohibitive for many national statistical offices in developing countries, R is distributed under the GNU license,

ensuring that the methods presented in this document are fully accessible and replicable regardless of institutional budget constraints.

Second, its specialized ecosystem stands out. The `survey` package [Lumley, 2024], available since 2004 and constantly evolving, implements a wide range of estimation methods for complex designs: Horvitz-Thompson estimators, calibration techniques, Taylor linearization, replication methods, weighted regression models, and design-adjusted hypothesis tests such as Rao-Scott tests, among others. In turn, the `srvyr` package [Freedman Ellis and Schneider, 2024] extends these capabilities by incorporating the syntax of the `tidyverse` ecosystem, facilitating a smooth transition between exploratory and inferential analysis. Complementary packages such as `TeachingSampling` [Gutiérrez, 2020] and `convey` [Jacob et al., 2024] consolidate an analytical environment that is difficult to match in other languages.

Finally, R offers strong integration with tools for scientific reproducibility. In particular, its combination with R Markdown and the `bookdown` system makes it possible to generate documents in which results, tables, and figures are produced directly from the source code. This capability is especially relevant in the field of official statistics, where traceability and methodological transparency are fundamental requirements.

III Structure of the Document

This document is organized according to a logical progression that moves from the conceptual foundations of sampling to more advanced applications of statistical modeling and treatment of nonresponse. This sequence is not arbitrary: each level of knowledge is a necessary prerequisite for understanding the next, allowing the reader to move coherently from basic principles to more sophisticated analytical tools.

First, the foundations of sampling design are addressed. Proper survey analysis necessarily begins with understanding its probability design. Before estimating any parameter, the analyst must become familiar with three constitutive elements: the target population, understood as the set of units about which inferences are to be made; the sampling frame, which corresponds to the operational enumeration of those units; and the sampling units at each stage of the design. These components determine both the validity and the scope of subsequent inferences.

Next, the computational elements that make it possible to materialize these concepts in an applied environment are introduced. The document incorporates variable management and transformation through a programming environment suitable for incorporating the sampling design, which constitutes the central axis of the analytical workflow and ensures that all statistical operations respect that structure.

Once the design has been correctly specified, the document moves on to the estimation of descriptive parameters. This section covers the most commonly used estimators in survey analysis for continuous variables, such as income, expenditure, or assets, including totals, means, and ratios. It also presents their corresponding precision measures, including variances, standard errors, and confidence intervals, adjusted for the complexity of the sampling design.

The document then develops statistical modeling, extending the analysis beyond descriptive statistics toward an analytical and inferential approach. In particular, it addresses linear regression models under complex designs, where the explicit incorporation of sampling weights

is fundamental for obtaining design-unbiased estimators. This context highlights the discussion between the design-based approach, which prioritizes statistical properties at the population level, and the model-based approach, which depends on correct functional specification. Consequently, the importance of assessing the sensitivity of results through comparisons between weighted and unweighted estimates is emphasized. In addition, generalized linear models and multilevel models are introduced, broadening the scope of analysis to nonnormal response variables, such as binary variables (logistic regression), counts (Poisson regression), and proportions.

The organization of the chapters follows the conceptual architecture described above, moving from foundations to more complex applications. Chapter 2 presents the basic concepts for household survey analysis, emphasizing that the validity of estimates depends on properly incorporating the sampling design. It explains the central elements of a survey, such as the target population, sampling frame, sampling units, stratification, clustering, and sampling weights. The chapter also introduces the main sampling estimators for totals, means, and other population parameters, and develops the foundations for estimating variance and confidence intervals.

Chapter 3 addresses the rigorous estimation of descriptive parameters for numerical variables under complex designs and develops techniques for estimating totals and means, together with variance estimation through the ultimate cluster method and the construction of confidence intervals with adjusted degrees of freedom. It also presents alternative approaches to variance estimation, such as Taylor linearization, estimating equations, and replication methods, including bootstrap and jackknife. The chapter also covers weighted quantile estimation, inequality analysis using indicators such as the Gini index and the Lorenz curve, estimation in domains and subpopulations, and visualization of results with confidence intervals.

Chapter 4 addresses the analysis of categorical variables under complex designs, including the assessment of associations between categories using methods that properly incorporate the sampling design. It develops techniques for estimating population counts and proportions from weights, together with estimation of their variability through linearization and construction of confidence intervals, including robust approaches for extreme proportions. It also presents weighted contingency tables with their corresponding frequencies and proportions, as well as independence tests that adjust the classical statistic to reflect the design effect. The chapter also covers the estimation and interpretation of association measures such as odds ratios, the analysis of differences in proportions between subpopulations through contrasts, and various visualization strategies, including bar charts with confidence intervals and maps.

Chapter 5 focuses on analyzing relationships between variables and building predictive or explanatory models that are valid under complex designs, when the classical assumptions of independence and identical distribution are not met. It presents the evolution of the problem from early contributions to more recent developments and discusses the implications of working with survey data in the context of regression models, including violations of the assumptions of the classical linear model. It contrasts the design-based approach with the model-based approach. The chapter also addresses hypothesis testing with appropriate adjustments, residual diagnostics, and identification of influential observations.

Chapter 6 extends statistical modeling to variables that do not follow a normal distribution, such as binary, multinomial, and strictly positive variables, in the context of complex sampling designs. To do so, it builds a unified framework based on generalized linear models, incorporating their fundamental components, and discusses inferential duality in finite populations, where the probability measure of the model and that of the sampling design coexist. It also contrasts

the classical maximum likelihood approach with maximum pseudo-likelihood, based on weighted estimating equations that incorporate sampling weights, and studies variance estimation through linearization techniques.

Chapter 7 reviews multilevel models, also called hierarchical or mixed-effects models, for analyzing data with a nested structure by simultaneously incorporating individual-level and contextual variables. Through regression examples, with intercepts and slopes that vary by stratum, it shows that ignoring the hierarchical structure can lead to incorrect inferences, whereas modeling it explicitly makes it possible to capture heterogeneity between groups and borrow information across them. The approach is formalized with the null model, which decomposes variance into within-stratum and between-stratum components and allows computation of the intraclass correlation coefficient (ICC), a key indicator for quantifying within-group dependence.

In addition, the first appendix introduces data management in `R` using the `tidyverse` environment, with emphasis on the `dplyr` package. Using the `BigCity` database, it presents a basic workflow that includes loading libraries, initial inspection of the database, use of the chaining operator `%>%`, and application of the main `dplyr` verbs: `filter()`, `select()`, `arrange()`, `mutate()`, `summarise()`, and `group_by()`. The chapter shows how to select records and variables, sort data, create new variables, and produce descriptive summaries, explaining both the logic of the code and the interpretation of its output.

Finally, the second appendix addresses inference in finite populations and the difference between model-based inference and design-based inference. Through simulation examples, it shows that in complex surveys randomness mainly comes from sample selection and that, therefore, estimates must incorporate inclusion probabilities and sampling weights. It presents the maximum likelihood method as a starting point for statistical models under assumptions of independent observations and then presents maximum pseudo-likelihood as an appropriate extension for survey data, where each individual contribution is weighted according to the design.

Readers who go through the chapters in order will have built the capacity to take any Latin American household survey, understand its design, process its data, estimate indicators of interest with the correct statistical precision, model relationships between variables validly under the complex design, and handle nonresponse with appropriate methods. That capacity is, in essence, what rigorous production of social statistics requires.

Chapter 2

Basic concepts in household surveys

Household surveys are one of the main tools for understanding the social and economic reality of a population. They provide essential information on living conditions, employment, education, health, and other aspects that guide public policy formulation and evidence-based decision-making. Rigorous analysis of these surveys is fundamental for studying the social, economic, and demographic reality of a population. However, the validity of the results depends not only on sample size or the quality of data collection, but also on how the survey was designed and how that design is incorporated into the statistical analysis. Ignoring the sample design and applying traditional analysis methods that assume a simple random sample can lead to biased estimates, inferential errors, and incorrect conclusions. For this reason, the analysis of household surveys cannot be separated from their selection design, since that design is the foundation that supports the validity and representativeness of the information produced.

In practice, household surveys often use complex designs that combine stratification with the selection of informants in several stages. These designs seek to optimize resources and improve the precision of estimates, although they may generate unequal inclusion probabilities. Ignoring these characteristics can lead to biased estimates, incorrect standard errors, and erroneous conclusions. As Särndal et al. [2003] and Gutiérrez [2016] point out, every survey starts from three fundamental concepts: the target population, the sampling frame, and the selected sample. These elements form the basis of the survey design and are essential for ensuring the validity of inferences and the representativeness of the results.

In this context, a fundamental aspect is the proper consideration of the sample design. To obtain valid conclusions about the population, it is necessary to adopt a design-based inference approach, which recognizes that the sample does not come from just any random selection, but from a carefully defined probability plan. In this scheme, each unit in the population has a known and nonzero probability of being selected, which is the main guarantee that the results can be generalized to the reference population.

Under this approach, it is possible to show that the estimates are unbiased (or approximately unbiased) with respect to the sampling design, without needing to assume specific distributions for the variable of interest. This feature gives design-based inference a robust and widely accepted character in the analysis of household surveys. A central component of this process is the sampling weights, which indicate how many population units are represented by each selected unit. These weights make it possible to adjust estimates to the particular features of the design, ensuring that the results adequately reflect the structure of the population.

As Korn and Graubard [1995] show, weighted and unweighted estimates can differ substantially, which demonstrates the importance of using analysis methods that are consistent with the survey design. Consequently, properly accounting for sampling weights, stratification, and selection units is not a secondary technical issue, but an essential requirement for ensuring the credibility of the results.

United Nations Statistics Division [2026] mentions the following example. Suppose a country is made up of two regions: Region A, with 100 inhabitants and an average income of $10,000, and Region B, with 900 inhabitants and an average income of $2,000. The true average income of the population is:

$$\theta = \frac{(100 \times 10.000) + (900 \times 2.000)}{100 + 900} = 2.800$$

If 50 people are selected in each region and the sampling design is ignored, assigning the same weight to all observations, the estimated average would be:

$$\hat{\theta} = \frac{(50 \times 10.000) + (50 \times 2.000)}{100} = 6.000$$

In this case, the estimate considerably overestimates national income, since Region A, which represents only 10% of the population, ends up having as much influence as Region B, which contains 90%. By contrast, when weights proportional to population size are applied (1 for Region A and 9 for Region B), the following estimate is obtained:

$$\hat{\theta} = \frac{(1 \times 50 \times 10.000) + (9 \times 50 \times 2.000)}{(1 \times 50) + (9 \times 50)} = 2.800$$

This result matches the true population value and shows how the proper use of weights corrects the bias introduced by an analysis that ignores the sample design.

I Target population and sampling units

Gambino and do Nascimento Silva [2009] mentions that, since household surveys became widespread in the 1940s, several trends associated with technological advances have emerged both in statistical agencies and in society in general, and these trends accelerated with the introduction of the computer. In this context, sampling arises as a response to the need to obtain precise statistical information about a target population without conducting a complete census. As Gutiérrez [2016] notes, sampling consists of conducting partial investigations of a population in order to infer results for the whole.

In recent decades, this methodology has become consolidated in different fields, especially in the government sector, through the production of official statistics that make it possible to monitor public policies and the Sustainable Development Goals. Likewise, the use of sampling has spread to academia, the private sector, and the media, where it is a fundamental tool for generating and analyzing information.

Every survey is associated with a finite population composed of individuals or elements about which information is desired. The set of units for which estimates and results will be produced is called the target population. Surveys also define units of analysis, which correspond to the different levels of disaggregation for which statistics of interest are presented, such as persons, households, or dwellings.

In household surveys in Latin America, it is common to use multistage sampling designs. To do this, different sampling units are selected that ultimately make it possible to reach the households that will be part of the sample. For example, primary sampling units (UPM) may correspond to census sectors built from the most recent population census, while secondary sampling units (USM) may be the dwellings located within those sectors.

To carry out a systematic household selection process, it is essential to have a sampling frame that serves as a link between the sampling units and the households that make up the target population. This frame is the set in which all the elements that make up the study population are identified and from which the sample is selected. In household surveys with complex sample designs, sampling frames are usually based on geographic areas. That is, they are built from spatial structures that link households or persons to delimited areas of the territory. This type of frame facilitates territorial organization into more manageable units and makes it possible to implement multistage selection processes while maintaining the principles of probability sampling.

Another fundamental characteristic of household surveys is clustering. In practice, many surveys first select primary sampling units, such as census sectors or enumeration areas, and then select households within each of them. This type of design reduces operational costs and facilitates fieldwork; however, it also has important implications for the precision of estimates. When households belonging to the same cluster share similar characteristics, the additional information contributed by each new observation within the cluster decreases.

For example, suppose a survey selects 100 clusters and, within each one, 10 households, resulting in a total sample of 1,000 households. If households in the same cluster display very similar behaviors (for example, if all have access to electricity), the effective variability of the sample is considerably reduced. In practical terms, the precision obtained could be equivalent to that of a simple random sample of only 100 households. Analyzing the 1,000 households as if they were completely independent observations, while ignoring the clustering structure, leads to an underestimation of standard errors and, consequently, to artificially narrow confidence intervals and potentially erroneous conclusions about the statistical significance of the results.

II Sampling estimators

When analyzing survey data, it is also essential to define the parameter of interest: a fixed numerical value that describes a characteristic of the whole population, denoted as U. The most common parameters include proportions, sizes, totals, means, and ratios, among others. Since, in practice, it is not possible to observe the entire population, sample surveys make it possible to infer these parameters from a sample, denoted as s.

In probability sampling, each unit in the population has a known inclusion probability greater than zero of being selected in the sample. These probabilities are the basis for calculating the basic sampling weights, which are then used to estimate population parameters through weighted sums

of the data collected in the survey. When the design and selection are implemented correctly, the resulting estimates are unbiased, meaning that on average they coincide with the true population value if the sampling process were repeated under the same conditions.

Nevertheless, basic sampling weights often require additional adjustments in order to improve the precision and robustness of the estimates. One of the most common adjustments is the treatment of nonresponse, through which the weights of the units that did respond are increased so that they also represent selected units that did not participate in the survey, thereby helping to reduce possible biases. Another widely used adjustment is calibration, which consists of modifying the weights to ensure that the weighted sums of certain auxiliary variables, such as age or sex, match known population totals from censuses or demographic projections. In addition to improving the consistency of estimates, calibration is a useful tool for detecting coverage problems or nonresponse patterns.

Weight adjustments, and calibration in particular, play a fundamental role in the analysis of surveys with complex designs. These procedures not only correct imperfections arising from nonresponse or incomplete coverage, but also strengthen the coherence between the sample and the target population, ensuring that inferences are statistically valid and comparable with other official sources [Kalton and Flores-Cervantes, 2003].

Calibration is especially relevant because it makes it possible to compare the survey's initial estimates with external reference values, usually obtained from censuses, administrative records, or demographic projections. This comparison is a powerful tool for detecting inconsistencies in the composition of the sample, identifying potential biases associated with nonresponse, and assessing the quality of fieldwork.

In addition, calibrated weights tend to reduce the variance of estimates, improving their precision without altering known population totals [Särndal and Lundström, 2005]. Consequently, calibration not only corrects but also optimizes the use of available information, making it possible to obtain more precise results.

The estimation of totals in surveys is a central step in statistical analysis applied to finite populations. Many indicators of interest for public policy formulation (such as, for example, the number of people living in poverty, the total number of employed persons, or aggregate household expenditure) are derived from a population total. For this reason, understanding how totals are defined and estimated is fundamental for ensuring the quality and relevance of the information produced.

In formal terms, if y_k denotes the value of a variable of interest for unit $k \in U$, the population total is defined as

$$t_y = \sum_U y_k,$$

Letting N be the size of the population, the population mean is defined as $\bar{y} = \frac{t_y}{N}$. Since in practice only a sample $s \subset U$ is observed, it is necessary to use estimators that incorporate the sampling design. For example, the Horvitz-Thompson (HT) estimator is expressed as

$$\hat{t}_y = \sum_s d_k y_k$$

where $d_k = 1/\pi_k$ are the basic weights from the sampling design and $\pi_k = \Pr(k \in s)$ are the first-order inclusion probabilities.

In practice, design weights are often modified to reflect additional processes such as nonresponse adjustment or calibration to known population totals, thus obtaining new adjusted weights, denoted as w_k for all $k \in s$. Replacing d_k with w_k makes it possible to improve precision and reduce bias in the estimates, especially when reliable auxiliary information sources are available.

Nevertheless, any estimate from a sample involves uncertainty. Even when the estimator is unbiased, results vary from one sample to another because of the randomness property of the design. This variability is quantified through the variance of the estimator, from which the standard error (se) or the coefficient of variation (cv) can be calculated. These indicators are indispensable tools for assessing the precision of estimated totals and, therefore, for interpreting statistical information properly. Under the design-based approach, the unbiased variance of the Horvitz-Thompson estimator can be expressed as:

$$\widehat{Var}_p(\hat{t}_y) = \sum_{k \in s} \sum_{l \in s} (d_k d_l - d_{kl}) y_k y_l,$$

where $d_{kl} = 1/\pi_{kl}$ and $\pi_{kl} = \Pr(k, l \in s)$ represent the joint inclusion probabilities. This expression requires the sampling design to satisfy $\pi_{kl} > 0$ for every pair of units $k, l \in U$.

To understand more concretely the relevance of considering the sample design when estimating totals and their variances, let us analyze an example from United Nations Statistics Division [2026] that shows how the estimation method adjusts to the selection scheme adopted. Suppose there is a finite population of size $N = 6$, from which a simple random sample (MAS) without replacement of size $n = 3$ is selected. In that sample, the values $(y_1 = 10, y_2 = 14, y_3 = 18)$ are observed. Under this design, the Horvitz-Thompson estimator of the population total is defined as the sum of the observed values divided by their inclusion probabilities, and its estimated variance is obtained from the covariances induced by the selection process. Thus, the estimated variance of the Horvitz-Thompson estimator is calculated as:

$$\widehat{Var}_{MAS}(\hat{t}_y) = \frac{N^2}{n} \left(1 - \frac{n}{N}\right) S^2_{y_s}$$

where $S^2_{y_s}$ corresponds to the sample variance of the observed values. Substituting into the expression gives

$$\widehat{Var}_p(\hat{t}_y) = \frac{36}{3} \left(1 - \frac{3}{6}\right) 16 = 96$$

By contrast, if the sampling design is ignored, an inexperienced analyst might incorrectly calculate the variance using the simplified formula:

$$\frac{N^2}{n} S^2_{y_s} = 192,$$

which would lead to an overestimation of the variance because the characteristics of the sample design are not considered. Likewise, the estimate of the population total is $\hat{t}_y = 84$. The standard error calculated according to the sampling design is

$$\sqrt{\widehat{Var}_p(\hat{t}_y)} = \sqrt{96} \approx 9.80.$$

By contrast, if the variance is estimated using a naive method that ignores the sampling design, the confidence interval would be wider and clearly biased, which could lead to erroneous inferences.

Finally, according to Kish [1965, p. 258], the design effect (DEFF) is defined as the ratio between the variance of an estimator obtained under a complex sampling design and the variance of the same estimator under simple random sampling (MAS) with the same sample size. Its estimate is expressed as:

$$\widehat{DEFF} = \frac{\widehat{Var}_p(\hat{\theta})}{\widehat{Var}_{MAS}(\hat{\theta})}$$

Here, $\widehat{Var}_p(\hat{\theta})$ corresponds to the estimated variance of $\hat{\theta}$ under the complex design $p(\cdot)$, while $\widehat{Var}_{MAS}(\hat{\theta})$ represents the estimated variance of the same estimator under a MAS design with the same sample size.

This indicator makes it possible to quantify how much the variance increases due to clustering and other characteristics of complex designs compared with simple sampling. According to Naciones Unidas [2009, p. 40], the DEFF can be understood in three ways: as the variance inflation factor relative to MAS, as a measure of the relative loss of precision, or as an indication of the increase in sample size that would be necessary in a complex design to achieve the same variance level as in a MAS. According to Park et al. [2003], the design effect of a survey may be due to the following three factors:

- Unequal weighting: the presence of nonuniform sample weights usually increases the variance slightly; therefore, the use of uniform weights is advantageous and explains why self-weighting designs are preferred in household surveys.
- Stratification: when applied correctly, it can reduce the variance, although in practice its variance-reducing effect is usually moderate.
- Multistage sampling: this generally increases the variance, since units within the same cluster tend to be more homogeneous among themselves than compared with units in other clusters.

In survey analysis, the design effect (DEFF) is an essential indicator for assessing the precision and efficiency of estimates, as well as for guiding the planning of future studies. A high value shows that the complex design introduces an increase in variance, reducing the precision of the results. By contrast, a value close to one indicates that the design has a minimal effect on the variance. This information allows researchers to identify whether it is necessary to adjust weighting, optimize stratification, or modify the subsampling size to increase efficiency in future survey operations.

The interpretation of a high DEFF should be done with caution, since it does not always imply that the sample design is inadequate. It is essential to consider the survey context: a value greater than three might seem alarming, but it is often due to practical limitations such as budget constraints, logistical difficulties, or the need to guarantee respondent participation. In certain household surveys, it may be essential to select only a fraction of eligible individuals in each household. In addition, coverage problems or nonresponse rates can increase the variability of sampling weights and, consequently, raise DEFF values. Gutiérrez and Babativa-Márquez [2023] provide a detailed analysis of design effects in household surveys in Latin America using BADEHOG data.

III Theoretical foundations

As Heeringa et al. [2017] emphasize, the calculation of population totals and means, together with their variances, has been essential for the development of probability sampling theory and the proper interpretation of household survey results. Determining population totals is one of the fundamental pillars of survey analysis. Means, proportions, and ratios are all derived from these totals. A total is defined as the sum of a specific variable (for example, income or expenditure) across the entire population.

With household surveys, the analysis of numerical data frequently involves calculating descriptive statistics such as means, totals, and ratios, since these summarize the main characteristics of the population and serve as a basis for decision-making. These estimates can be calculated for the population as a whole or for specific subgroups, depending on the objectives of the research.

A Point estimation

Once the objective of the sample design has been defined, the process of estimating the parameters of interest is carried out. For the purposes of this document, we begin with the estimation of total household income. For the estimation of totals in complex sample designs, strata will be denoted by the letter h; primary sampling units, which are contained within strata, by the letter i; while observed units (households or persons) will be denoted by the letter k. Thus, the estimator of the total can be expressed as:

$$\hat{t}_y = \sum_h \sum_i \sum_k w_{hik}\, y_{hik}$$

Here, w_{hik} corresponds to the expansion factor of unit k, while y_{hik} corresponds to the observation of the variable of interest for that same unit. Calculating the estimate of the total and its associated variance can be complex, so it is necessary to consider different methodological approaches.

In terms of notation, and in order to simplify the writing throughout this document, unless otherwise indicated, $\sum_h$ will denote the sum over all strata in the population; $\sum_i$ will represent the sum over all primary sampling units selected in the sample within stratum h; and $\sum_k$ will correspond to the sum over all elements observed in the sample of UPM i, belonging to stratum h of the population of interest.

B Variance estimation

When working with household surveys, it is essential not only to obtain point estimates but also to quantify the uncertainty associated with those estimates. Understanding and estimating this uncertainty is an essential part of the analysis of household survey data. By applying appropriate methods, users can assess the precision of their estimates. There are several methods for estimating this precision and, with the support of modern software, these approaches can be implemented efficiently to support rigorous and reliable analyses. The main methods include:

- Estimating equations: provide a flexible framework for estimating totals, means, ratios, and other parameters, as well as their corresponding variances, integrating a unified view of sampling theory [Binder, 1983].
- Taylor linearization: consists of approximating complex nonlinear statistics through linear expressions and then estimating the variance of that approximation.
- Ultimate cluster method: frequently used in surveys based on stratified multistage sampling. It is based on calculating variance from the differences between estimates obtained at the level of the primary sampling units (UPM). This method is often combined with Taylor linearization to estimate the variance of nonlinear statistics, such as means or ratios.
- Bootstrap and other replication methods: are based on repeatedly taking subsamples from the observed data set, calculating estimates for each replicate, and then using the variability among these replicated estimates to infer the variance of the main estimator.

In the context of household surveys, many complex parameters can be formulated as linear combinations of other parameters $f(\theta_1, ..., \theta_J)$. Specifically, let:

$$f = f(\theta_1, ..., \theta_J) = \sum_{j=1}^{J} a_j \theta_j,$$

where the coefficients a_j are known constants and θ_j represent population parameters. If $\hat{\theta}_j$ is an estimator of θ_j, then a sample estimator of f is defined as:

$$\hat{f} = \sum_{j=1}^{J} a_j \hat{\theta}_j,$$

Thus, its variance can be expressed as:

$$Var(\hat{f}) = \sum_{j=1}^{J} a_j^2 Var(\hat{\theta}_j) \; + \; 2 \sum_{j=1}^{J-1} \sum_{k>j}^{J} a_j a_k \, Cov(\hat{\theta}_j, \hat{\theta}_k).$$

This result is particularly important because it includes numerous cases of practical interest that will be developed throughout this document.

B.1 Estimating equations

Many population parameters can be expressed as solutions to estimating equations involving population totals. Although the technical details can be complex, the fundamental idea is that the same principles used to estimate totals can also be applied to variance estimation. This general framework makes the method simple and flexible, facilitating its implementation in specialized statistical software. A generic population estimating equation can be expressed as follows:

$$\sum_{k \in U} z_k(\theta) = 0,$$

where $z_k(\cdot)$ is an estimating function evaluated for unit k and θ represents the population parameter of interest. These equations provide a general framework for defining and calculating various population parameters, such as totals, means, and ratios. For example, for the population total, $z_k(\theta) = y_k - \theta/N$ is defined, so that the estimating equation is $\sum_{k\in U}(y_k - \theta/N) = 0$. The solution to this equation leads to $\theta = \sum_{k\in U} y_k = Y$, that is, to the population total.

Similarly, for the population mean, $z_k(\theta) = y_k - \theta$ is used, and the equation $\sum_{k\in U}(y_k - \theta) = 0$ has as its solution $\theta = \left(\sum_{k\in U} y_k\right)/N = \overline{Y}$, corresponding to the population mean. Likewise, in the case of ratios of totals, $z_k(\theta) = y_k - \theta x_k$ is defined. Thus, the equation $\sum_{k\in U}(y_k - \theta x_k) = 0$ leads to the solution $\theta = \dfrac{\sum_{k\in U} y_k}{\sum_{k\in U} x_k} = R$, which corresponds to the population ratio between the totals of variables y and x. These formulations show how different parameters of interest can be expressed through estimating equations that share a common structure.

The idea of defining population parameters as solutions to estimating equations at the population level naturally leads to a general method for obtaining sample estimators. In this case, equations of the following form are used:

$$\sum_{k\in s} w_k\, z_k(\theta) = 0,$$

where w_k are the expansion factors and $z_k(\theta)$ is the estimating function evaluated for each unit in the sample. Under probability sampling and assuming complete response, the sample sum $\sum_{k\in s} d_k\, z_k(\theta)$ is unbiased with respect to its population analogue, which ensures that the solutions to these equations are consistent estimators of the population parameters.

B.2 Taylor linearization

This technique makes it possible to approximate the variance of nonlinear estimators. The procedure consists of applying a first-order Taylor expansion around the estimated parameter in order to replace the nonlinear estimator with a linear expression. This facilitates the calculation of variances in situations where exact formulas do not exist or their derivation is too complex. A consistent estimator of the variance, derived through Taylor linearization for solutions to sample estimating equations, can be expressed as:

$$\widehat{Var}(\hat{\theta}) \approx [\hat{J}(\hat{\theta})]^{-1}\, \widehat{Var}_p\left[\sum_{k\in s} w_k\, z_k(\hat{\theta})\right] [\hat{J}(\hat{\theta})]^{-1}$$

where $\hat{J}(\hat{\theta}) = \sum_{k\in s} w_k \left[\frac{\partial z_k(\theta)}{\partial \theta}\right]_{\theta=\hat{\theta}}$. This result shows how Taylor linearization converts variance estimation for complex parameters into a problem of total estimation, which explains its widespread adoption in specialized software for survey analysis.

B.3 Ultimate cluster

The ultimate cluster method is a direct and robust approach for estimating the variance of totals in surveys that use stratified multistage cluster sampling designs. Proposed by Hansen et al. [1953],

this method simplifies the complexity of multilevel designs by focusing only on the variation among primary sampling units (UPM). It is assumed that, within each sampling stratum, the UPM were selected independently with replacement (possibly with unequal probabilities), although in practice selection is usually carried out without replacement.

The method is based on the variation among statistics calculated at the UPM level. When applied correctly, it implicitly reflects any subsampling carried out within the UPM, allowing simpler but reliable variance estimates. It is especially useful in complex designs that include stratification and unequal selection probabilities for both UPM and lower-level units (households and individuals). The requirements for applying this method include the availability of unbiased estimates of totals for the variables of interest in each selected UPM. In addition, if the sample is stratified at the first stage, at least two UPM must be available per stratum, since this makes it possible to estimate adequately the variability among clusters within each stratum.

Thus, consider a multistage sampling design where n_h UPM are selected in stratum h, $(h = 1, \dots, H)$. An estimate of the total of the variable of interest in UPM i of stratum h is:

$$\hat{t}_{y_i} = \sum_{k \in s_{hi}} w_{hik}\, y_{hik}$$

Here, s_{hi} corresponds to the sample of units in UPM i of stratum h; therefore, an unbiased estimator of the population total would be expressed as

$$\hat{t}_y = \sum_h \sum_i \hat{t}_{y_i}$$

According to the ultimate cluster method, assuming that $\hat{\bar{t}}_{y_h} = (1/n_h) \sum_i \hat{t}_{y_i}$ is the sample mean of the estimated UPM totals in stratum h, an approximate estimator of the variance of $\hat{t}_y$ is obtained through the following expression:

$$\widehat{Var}(\hat{t}_y) = \sum_h \frac{n_h}{n_h - 1} \sum_i (\hat{t}_{y_i} - \hat{\bar{t}}_{y_h})^2$$

For more details, see Hansen, Hansen et al. [1953, p. 258] or Wolter and Wolter [2007]. Although this method was originally proposed for calculating variances of total estimators, it can be combined with Taylor linearization or estimating equations to derive variances of other population parameters that can be formulated as solutions to estimating equations. This flexibility makes the method applicable to various contexts of household survey analysis.

A fundamental assumption of this technique is that, within each stratum, the UPM are selected independently and with replacement. In practice, most surveys select UPM without replacement, generating more efficient designs. Consequently, the variances calculated under the independence assumption are approximations of the true sampling variances. When the sampling fraction is small (for example, less than 5%), these approximations are usually sufficiently precise for use by national statistical offices or other analysts.

This method stands out for its simplicity and robustness, making it very attractive in practice. Although more sophisticated methods that consider all stages of the design may offer slightly more precise variance estimates, their application requires more detailed information and greater

computational complexity. By contrast, the ultimate cluster method provides a reliable and efficient approximation, especially useful when estimating totals or means in household surveys. For a detailed analysis of the precision of this approximation and possible alternatives, see Särndal et al. [2003].

B.4 Bootstrap

In many cases, public survey microdata omit essential design information, such as identifiers for strata or primary sampling units (UPM), to protect respondent confidentiality. This omission limits users' ability to calculate valid variances. In such situations, it is recommended that national statistical offices provide replication weights, which allow analysts to estimate standard errors correctly. Without these data, secondary users cannot reproduce the published standard errors or adequately account for the complex survey design.

Replication methods estimate variance by generating subsets of the original sample, calculating estimates for each one, and using the observed variability among these estimates to approximate the variance of the main estimator. They are particularly useful when information on strata or UPM is not available, a situation in which the ultimate cluster method cannot be applied.

For example, Bootstrap is a robust and versatile replication tool. Originally introduced by Efron [1979] for data that did not come from surveys, its most widely used adaptation for household surveys is the Rao-Wu-Yue Rescaling Bootstrap [Rao et al., 1992]. This method fits optimally with stratified and multistage sampling designs and is widely used for variance estimation in complex surveys.

The procedure consists of generating many replicates of the original sample, simulating repeated draws from the population. Each replicate is constructed by creating additional columns of replication weights in the database, following this process:

- For each stratum, UPM are randomly selected with replacement; some may be repeated and others may not appear. Each selected UPM is included with all its observations. If the first-stage sample size in stratum h is greater than two ($n_h > 2$), the number of UPM selected per replicate is $n_h - 1$.
- This process is repeated many times, usually hundreds, generating a large number of replicates. The number of times that a UPM i from stratum h appears in replicate r is denoted $n_{hi}^{(r)}$, varying between 0 and $n_h - 1$.
- From each replicate, new bootstrap weights are calculated for all units, reflecting how many times their UPM was selected. The weight of unit k in replicate r is calculated as:

$$w_{hik}^{(r)} = w_{hik} \times \frac{n_h}{n_h - 1} \times n_{hi}^{(r)}$$

If the original weights include nonresponse or calibration adjustments, these must also be applied to each set of bootstrap weights.

When only Bootstrap replication weights are provided, analysts can estimate standard errors correctly, even without strata or UPM identifiers. For each replicate r, the parameter of interest

$\hat{\theta}^{(r)}$ is calculated using the bootstrap weights $w_{hik}^{(r)}$. The variance of the original estimator is approximated through the variability among all replicates:

$$\widehat{Var}_B(\hat{\theta}) = \frac{1}{R}\sum_{r=1}^{R}\left(\hat{\theta}^{(r)} - \tilde{\theta}\right)^2, \quad \tilde{\theta} = \frac{1}{R}\sum_{r=1}^{R}\hat{\theta}^{(r)}$$

This approach ensures that the dispersion among replicates faithfully captures the uncertainty of the parameter. Bootstrap offers multiple advantages. Although it requires greater computational processing, it is effective for complex survey designs and makes it possible to estimate parameters that are difficult to calculate with traditional methods, such as medians or other nonlinear statistics. It is especially useful for analysts working with databases that lack strata and UPM identifiers but include replication weights.

The simplicity of the method facilitates its application even without specialized statistical software. However, most modern statistical packages already include procedures for applying Bootstrap and calculating variances, expanding its availability and robustness. Nevertheless, its use is not recommended in repeated surveys with overlapping samples or in situations with large sampling fractions and small sample sizes [Bruch et al., 2011].

IV Confidence intervals

In addition to producing point estimates, one of the fundamental objectives of a survey is to quantify the uncertainty associated with those estimates. In this context, confidence intervals make it possible to construct a range of plausible values for the population parameter of interest, explicitly incorporating the variability introduced by the sample design. The $(1-\alpha)100\%$ confidence interval for the population total Y is calculated as:

$$\hat{\theta} \pm t_{1-\alpha/2,df} \times \sqrt{\widehat{Var}(\hat{\theta})}$$

where $\hat{\theta}$ corresponds to the sampling estimator of the population parameter θ; $\widehat{Var}(\hat{\theta})$ represents the variance estimator under the complex survey design; and $t_{1-\alpha/2,df}$ is the quantile of Student's t distribution with df degrees of freedom. In complex surveys, the degrees of freedom are usually related to the number of primary sampling units (UPM) and the strata considered in the design. A common approximation is to calculate them as:

$$df = m - H$$

where m represents the total number of UPM observed in the sample and H the number of strata. This approximation reflects the effective amount of independent information available to estimate sampling variability. As the degrees of freedom increase, Student's t distribution converges to the standard normal distribution. This property explains why normal approximations are often used to report confidence intervals, especially in large-scale household surveys. However, when the number of clusters is small or there are strata with few UPM, the normal approximation may underestimate uncertainty and produce excessively narrow intervals.

Chapter 3

Analysis of continuous variables

The statistical analysis of surveys has constantly evolved, driven by the development of new methodologies and approaches that seek to improve the precision and representativeness of the results. In general, these advances emerge in the academic field and, over time, are adopted by public institutions, private companies and statistical organizations, which demand their incorporation into the main data analysis programs.

This chapter presents the main procedures for the statistical analysis of continuous variables using `R`, appropriately incorporating the characteristics of the sampling design, such as expansion weights, stratification and conglomeration. Throughout the chapter, methods are illustrated to estimate population parameters, measure their precision, and make valid inferences from data from complex surveys. Likewise, functions and tools of the `survey` [Lumley, 2024] package are introduced, one of the most used resources in `R` for survey analysis.

Databases used in survey analysis can be found in a wide variety of formats (.xlsx, .dat, .csv, .parquet, .sas7bdat, .sav, .txt, among others). However, a good practice is to import the information from any of these formats and immediately save it in a file with the extension `.rds`, which is the native format of `R`.

`.rds` files allow you to store any object or information in `R`, such as data frames, vectors, matrices or lists, and are characterized by their complete compatibility with the R work environment. In addition, they facilitate efficient reading and writing of data, reducing loading times and avoiding compatibility problems between sessions. Another `R` format is `.Rdata`, which allows multiple objects to be saved in a single file. In contrast, `.rds` files store only one object, making their use more controlled and reproducible. For this reason, it is recommended to preferably work with the `.rds` format when documenting projects or reproducing analyses.

To illustrate the computational syntax that will be used throughout the chapter, we use the same database as in the previous chapter, which contains a sample of 2,605 records from a complex sample design. The following shows how to load a file with the `.rds` extension:

```
library(tidyverse)

survey_data <- readRDS("Data/encuesta.rds")
```

According to Naciones Unidas [2009, section 7.8], it is essential that the structure of the sampling design be taken into account in the inference process when estimating official statistics based on

household surveys. As emphasized in this document, ignoring this aspect can generate biased estimates and underestimated sampling errors. In this sense, statistical programs incorporate specific functionalities to handle data from surveys with complex designs [Heeringa et al., 2017, appendix A].

Once the sample of households has been loaded into R, the next step is to define the sample design from which said sample comes. For this, the `srvyr` [Freedman Ellis and Schneider, 2024] package will be used, which appears as a complement to the `survey` [Lumley, 2024] package. These libraries allow defining `survey.design` type objects, to which the estimation and survey analysis functions are applied, and which can be combined with the programming of the `tidyverse` [Wickham et al., 2026a] package. Here is an example of defining an object of type `survey_design` for the survey:

```
options(survey.lonely.UPM = "adjust")

library(survey)
library(srvyr)

survey_design <- survey_data %>%
  as_survey_design(
    strata = Stratum,
    ids = PSU,
    weights = wk,
    nest = TRUE
  )
```

In this code, `survey_data` corresponds to the database that contains the selected sample. From this data, the function `as_survey_design()` converts the information into an object of class `survey_design`, which stores all the elements necessary to describe the sample design of the survey. These include the stratification variable (`strata`), which identifies the strata defined in the design, and the variable specified in `ids`, which represents the primary sampling units (PSUs) or clusters selected in the first stage of sampling. Likewise, the `weights` argument defines the variable that contains the expansion factors associated with each observation. Finally, the `nest = TRUE` option indicates that PSUs are nested within strata, a necessary characteristic to correctly represent the hierarchical structure of the sampling design. ## Estimation of descriptive parameters {#estimacion-de-parametros-descriptivos}

In household surveys it is common to estimate descriptive parameters associated with numerical variables, such as totals, means, ratios and standard deviations. These measures allow us to characterize different social and economic phenomena of the population, for example income, expenditure, hours worked or household size. This section presents the logic behind the syntax used for this type of estimation, along with its corresponding precision measures. ### Total estimate {#estimacion-del-total}

In R, the `survey_total()` function is used to estimate totals from the design object. This feature automatically incorporates expansion factors and design structure, ensuring that estimates correctly reflect the inference model.

```
survey_total(
  x,
  na.rm = FALSE,
  vartype = c("se", "ci", "var", "cv"),
  level = 0.95,
  deff = FALSE
)
```

The main arguments of the `survey_total()` function are `x`, `na.rm`, `vartype`, `level`, and `deff`. The `x` argument corresponds to the variable on which you want to calculate the total; It can also be left empty when seeking to obtain only the weighted total of cases in the sample. Meanwhile, `na.rm` indicates whether missing values (`NA`) should be removed before performing the calculation, with `FALSE` being the default value.

The `vartype` argument allows you to specify the type of uncertainty measure that you want to estimate, and one or more results can be requested simultaneously, such as the standard error (`"se"`), the confidence interval (`"ci"`), the variance (`"var"`) or the coefficient of variation (`"cv"`). Likewise, `level` defines the confidence level used to calculate the confidence intervals, whose default value is 0.95. On the other hand, the confidence interval can be obtained directly by including the argument `vartype = "ci"` within `survey_total()`, or by using the function `confint()` on the resulting object, specifying the required confidence level. Finally, `deff` is a logical value that indicates whether to return the design effect (DEFF).

The following example presents how to estimate totals and their confidence intervals for different variables of interest in `R`, using the function `survey_total()`, and presented in Table 3.1:

```
survey_design %>%
  summarise(total = survey_total(Income, vartype = c("se", "ci"), deff = TRUE))
```

Table 3.1: Estimation of total household income with 95% confidence interval

total	total_se	total_low	total_upp	total_deff
85793667	4778674	76331414	95255920	11

From the table, it is obtained that the estimate of the total population for the total income is 85,793,667. This value corresponds to the point estimator obtained from the survey, incorporating the expansion factors and the characteristics of the sample design. The standard error associated with this estimate is 4,778,674, which reflects the sample variability of the estimator. From this value, a 95% confidence interval is constructed, whose lower and upper limits are 76,331,414 and 95,255,920, respectively. The design effect (`deff`) is equal to 11, suggesting a significant loss of accuracy mainly associated with complex design characteristics. Consequently, although the nominal sample size may be large, the effective sample size would be considerably smaller due to the correlation between observations within the clusters.

To continue illustrating the use of the `survey_total()` function within the `srvyr` approach, let us estimate total household expenditures, but now calculating the 90% confidence interval. The following code performs this estimation, presented in Table 3.2:

```
survey_design %>%
  summarise(total = survey_total(Expenditure,
                                 vartype = "ci", level = 0.9,
                                 deff = TRUE))
```

Table 3.2: Estimation of total household expenses with 90% confidence interval

total	total_low	total_upp	total_deff
55677504	51360469	59994539	10.2

If the objective is to estimate total household income, but disaggregated by sex, the function `group_by()` can be used to group by the variable of interest, together with `cascade()` from the `srvyr` library, which allows adding a row with the general total at the end of the table. In this way, category totals and the overall total can be easily obtained within the same summary framework, as shown in Table 3.3.

```
survey_design %>%
  group_by(Sex) %>%
  cascade(total = survey_total(Income,
                               vartype = c("se", "ci"),
                               level = 0.95),
          .fill = "Total income")
```

Table 3.3: Total household income disaggregated by sex

Sex	total	total_se	total_low	total_upp
Female	44153820	2324452	39551172	48756467
Male	41639847	2870194	35956576	47323118
Total income	85793667	4778674	76331414	95255920

As seen in the previous codes, a practical way to obtain the estimates of the total, its standard error and its confidence interval is through the argument `vartype`, specifying the options `"se"` and `"ci"` respectively. ### Estimation of the mean {#estimacion-de-la-media}

The estimation of the population average occupies a central place in household surveys, since many of the indicators used for socioeconomic analysis correspond to average values, such as average household income, per capita spending or average hours worked. This type of parameters allows us to summarize the general behavior of the variables of interest and describe the central tendencies of the target population. According to Gutiérrez [2016], the population mean estimator can be expressed as a non-linear ratio between two population totals, which are estimated as follows:

$$\hat{\bar{y}} = \frac{\hat{t}_y}{\hat{N}} = \frac{\sum_h \sum_i \sum_k w_{hik}\, y_{hik}}{\sum_h \sum_i \sum_k w_{hik}}$$

Since $\hat{\bar{y}}$ is not a linear statistic, there is no closed formula for the variance of this estimator. For this reason, it is necessary to resort to resampling methods or Taylor linearization to approximate its variance. In this particular case, using Taylor series, the variance is defined as:

$$\widehat{Var}\left(\hat{\bar{y}}\right) \approx \frac{\widehat{Var}\left(\hat{t}_y\right) + \hat{\bar{y}}^2 \ \widehat{Var}\left(\hat{N}\right) - 2 \ \hat{\bar{y}} \ \widehat{Cov}\left(\hat{t}_y, \hat{N}\right)}{\hat{N}^2}$$

As can be seen, the calculation of the variance of the population mean involves complex analytical components, such as the covariance between the estimated total and the estimated population size. However, the `srvyr` package in `R` facilitates these calculations by incorporating functions that directly estimate the mean, its standard error, the confidence interval, and the design effect. Below is the syntax to estimate the average household income, along with its 95% confidence interval, presented in Table 3.4:

```
survey_design %>%
  summarise(mean = survey_mean(Income,
                               vartype = c("se", "ci"),
                               level = 0.95,
                               deff = TRUE))
```

Table 3.4: Estimation of average household income

mean	mean_se	mean_low	mean_upp	mean_deff
571	28.5	515	627	8.82

As can be seen, the arguments used in the `survey_mean()` function are similar to those of `survey_total()`. In this case, `vartype` allows you to obtain the standard error (`"se"`) and the confidence interval (`"ci"`), while the argument `deff = TRUE` requests the calculation of the design effect. Similarly, it is possible to estimate the average household expenses, using the same code structure, presented in Table 3.5:

```
survey_design %>%
  summarise(mean = survey_mean(Expenditure,
                               vartype = c("se", "ci"),
                               level = 0.95,
                               deff = TRUE))
```

Table 3.5: Estimation of average household expenses

mean	mean_se	mean_low	mean_upp	mean_deff
371	13.3	344	397	6.02

Estimates of the mean by subgroups can also be made following the same scheme shown previously. Particularly, household expenses discriminated by sex are presented in Table 3.6:

```
survey_design %>%
  group_by(Sex) %>%
  cascade(mean = survey_mean(Expenditure,
                             level = 0.95,
                             vartype = c("se", "ci")),
          .fill = "Mean expenditure") %>%
  arrange(desc(Sex))
```

Table 3.6: Average household expenses disaggregated by sex

Sex	mean	mean_se	mean_low	mean_upp
Mean expenditure	371	13.3	344	397
Male	374	16.1	343	406
Female	367	12.3	343	391

In an analogous manner, the average expenditure by area can be calculated, which is presented in Table 3.7:

```
survey_design %>%
  group_by(Zone) %>%
  cascade(mean = survey_mean(Expenditure,
                             level = 0.95,
                             vartype = c("se", "ci")),
          .fill = "Mean expenditure") %>%
  arrange(desc(Zone))
```

Table 3.7: Average household expenditure disaggregated by area

Zone	mean	mean_se	mean_low	mean_upp
Urban	460	22.2	416	504
Rural	274	10.3	254	294
Mean expenditure	371	13.3	344	397

Likewise, it is possible to make a combined estimate by sex and area, presented in Table 3.8:

```
survey_design %>%
  group_by(Zone, Sex) %>%
  cascade(mean = survey_mean(Expenditure,
                             level = 0.95,
                             vartype = c("se", "ci")),
          .fill = "Mean expenditure") %>%
  arrange(desc(Zone), desc(Sex))
```

Table 3.8: Average household expenses disaggregated by area and sex

Zone	Sex	mean	mean_se	mean_low	mean_upp
Urban	Mean expenditure	460	22.2	416	504
Urban	Male	470	27.0	416	523
Urban	Female	451	20.1	411	491
Rural	Mean expenditure	274	10.3	254	294
Rural	Male	275	10.2	255	296
Rural	Female	273	11.6	250	296
Mean expenditure	Mean expenditure	371	13.3	344	397

A Ratio estimation

A particular case of nonlinear functions of totals is the population ratio, defined as the quotient between two population totals associated with characteristics of interest. Ratios allow the relationship between two variables to be expressed, which is especially useful for building comparative and monitoring indicators.

This type of parameter is widely used in household surveys to construct relative indicators, for example, the number of men for each woman, the proportion of employed people compared to the working-age population or the average number of pets per household. Ratios are also used in international frameworks. For example, Indicator 2.1.1 of the Sustainable Development Goals (SDGs) (prevalence of undernourishment) is calculated from the ratio between food consumption, measured in calories ingested, and the minimum energy requirements of the diet, determined according to age, sex and level of physical activity.

Since the ratio corresponds to the quotient between two unknown population parameters, both the numerator and the denominator must be estimated from the sample [Bautista, 1998]. Being Y the total of the variable y and X the total of the variable x, the population ratio is defined as $R = \frac{t_y}{t_x}$ and its point estimator in the framework of complex sampling is expressed as:

$$\hat{R} = \frac{\hat{t}_y}{\hat{t}_x} = \frac{\sum_h \sum_i \sum_k w_{hik}\ y_{hik}}{\sum_h \sum_i \sum_k w_{hik}\ x_{hik}}$$

However, since $\hat{R}$ is a quotient between two estimators, that is, two random variables, the calculation of its variance is not trivial. For this, Taylor linearization is used, as shown by Gutiérrez [2016], or resampling methods. In particular, the `survey_ratio` function implements the estimation of ratios and their variances in the framework of complex surveys.

A direct example in household surveys is the **expenditure/income** ratio, which helps identify consumption patterns and levels of economic sustainability of households. The following illustrates how to estimate the ratio between household expenditure and household income, presented in Table 3.9:

```
survey_design %>%
  summarise(ratio = survey_ratio(numerator = Expenditure,
                                 denominator = Income,
                                 level = 0.95,
                                 vartype = c("se", "ci"))
            )
```

Table 3.9: Household expenditure/income ratio

ratio	ratio_se	ratio_low	ratio_upp
0.649	0.023	0.603	0.695

In this case, the expenditure variable was specified as the numerator (`numerator`), the income variable as the denominator (`denominator`), the confidence level (`level`) used to construct the confidence intervals, and the precision measures required by the argument (`vartype`). As can be seen, the ratio between expenditure and income is, approximately, 0.65. Which implies that for every 100 monetary units that enter the home, 65 units are spent, achieving a 95% confidence interval of 0.60 and 0.69.

If now the objective is to estimate the ratio between women and men in the example base, it is done as follows, presented in Table 3.10:

```
survey_design %>%
  summarise(ratio = survey_ratio(numerator = (Sex == "Female"),
                                 denominator = (Sex == "Male"),
                                 level = 0.95,
                                 vartype = c("se", "ci"))
            )
```

Table 3.10: Ratio of women to men in the population

ratio	ratio_se	ratio_low	ratio_upp
1.11	0.035	1.04	1.18

Given that the sex variable in the database is categorical, it was necessary to use indicator variables to calculate the ratio, using `Sex == "Female"` to identify women and `Sex == "Male"` for men. The results show that in the analyzed population there are a greater number of women than men, obtaining an estimated ratio of 1.11. This indicates that, for every 100 men, there are approximately 111 women. Furthermore, the 95% confidence interval suggests that this ratio could vary between 1.05 and 1.18. If you want to make the ratio of women and men but in rural areas, it would be done as follows, presented in Table 3.11:

```
rural_subset <- survey_design %>%
  filter(Zone == "Rural")
rural_subset %>%
  summarise(ratio = survey_ratio(numerator = (Sex == "Female"),
                                 denominator = (Sex == "Male"),
                                 level = 0.95,
                                 vartype = c("se", "ci"))
            )
```

Table 3.11: Ratio of women to men in rural areas

ratio	ratio_se	ratio_low	ratio_upp
1.07	0.035	0.997	1.14

Now, another analysis of interest is to estimate the expense ratio but only in the female population. The computational codes are presented below, as shown in Table 3.12.

```
female_subset <- survey_design %>%
  filter(Sex == "Female")

female_subset %>%
  summarise(ratio = survey_ratio(numerator = Expenditure,
                                 denominator = Income,
                                 level = 0.95,
                                 vartype = c("se", "ci"))
            )
```

Table 3.12: Expenditure/income ratio in households headed by women

ratio	ratio_se	ratio_low	ratio_upp
0.658	0.02	0.619	0.698

The result is that for every 100 monetary units that women receive, 66 are spent with a confidence interval between 0.62 and 0.69. Finally, similarly for men, the expense ratio is very similar to that for women, as shown in Table 3.13.

```
male_subset <- survey_design %>%
  filter(Sex == "Male")

male_subset %>%
  summarise(ratio = survey_ratio(numerator = Expenditure,
                                 denominator = Income,
```

```
                                level = 0.95,
                                vartype = c("se", "ci"))
            )
```

Table 3.13: Expenditure/income ratio in households headed by men

ratio	ratio_se	ratio_low	ratio_upp
0.639	0.029	0.582	0.696

I Estimation of distribution and inequality parameters

A Dispersion estimation

In household surveys it is also essential to estimate measures of dispersion of the variables studied, since these allow us to evaluate the degree of heterogeneity existing in the population. For example, knowing how dispersed household incomes are is key to analyzing economic inequalities and guiding the design of public policies. Among the most used measures of dispersion is the standard deviation, which quantifies how far the observed values are from their mean. The estimator of this parameter is presented below:

$$\hat{s}_y = \sqrt{\frac{\sum_h \sum_i \sum_k w_{hik} \left(y_{hik} - \hat{\bar{y}}\right)^2}{\sum_h \sum_i \sum_k w_{hik} - 1}}$$

To estimate the standard deviation of numerical variables in household surveys using `R`, the function `survey_var` can be used. Below is an example of its use applied to the estimation of the standard deviation of income, the results of which are shown in Table 3.14.

```
survey_design %>%
  group_by(Zone) %>%
  summarise(
    Var = survey_var(
      Income,
      level = 0.95,
      vartype = c("se", "ci"),
      na.rm = TRUE
    )
  ) %>%
  mutate(
    Sd     = sqrt(Var),
    Sd_low = sqrt(Var_low),
    Sd_upp = sqrt(Var_upp)
  )
```

Table 3.14: Standard deviation of income disaggregated by area

Zone	Var	Var_se	Var_low	Var_upp	Sd	Sd_low	Sd_upp
Rural	96274	13794	68960	123588	310	263	352
Urban	338628	81241	177763	499494	582	422	707

As observed in the previous example, the standard deviation of income by area was estimated, also reporting its corresponding 95% confidence interval. The arguments of the `survey_var()` function are equivalent to those previously used for the estimation of means and totals. If the interest is to estimate the standard deviation of income simultaneously disaggregating by sex and area, the corresponding computational codes are presented in Table 3.15.

```
survey_design %>%
  group_by(Zone, Sex) %>%
  summarise(
    Var = survey_var(
      Income,
      level = 0.95,
      vartype = c("se", "ci"),
      na.rm = TRUE
    )
  ) %>%
  mutate(
    Sd     = sqrt(Var),
    Sd_low = sqrt(Var_low),
    Sd_upp = sqrt(Var_upp)
  )
```

Table 3.15: Standard deviation of income disaggregated by area and sex

Zone	Sex	Var	Var_se	Var_low	Var_upp	Sd	Sd_low	Sd_upp
Rural	Female	86947	12459	62277	111618	295	250	334
Rural	Male	106119	15616	75197	137040	326	274	370
Urban	Female	323069	82058	160585	485553	568	401	697
Urban	Male	356141	83488	190826	521456	597	437	722

B Percentile estimation

Non-central position measures, such as percentiles and medians, allow us to identify specific points in the distribution of a variable of interest other than its average value. In household surveys, this type of measure is especially useful to characterize the distribution of economic and social variables, such as income, expenditure or hours worked. The median, for example, corresponds to

the value that divides the population into two equal parts: 50% of the observations are below said value and the other 50% above it. Unlike the mean, the median is not very sensitive to extreme values, which is why it is usually considered a robust measure of central tendency.

Similarly, percentiles allow you to identify specific segments of the population distribution. For example, income percentiles can be used to identify the population in the top 10% of the distribution for tax purposes, or to target subsidies to households in the lowest percentiles. According to Loomis et al. [2005], the estimation of quantiles in complex surveys is based on the use of weighted estimators and the estimation of the cumulative distribution function (CDF) of the population. For a finite population of size N, an estimator of this distribution is given by:

$$\hat{F}(x) = \frac{\sum_h \sum_i \sum_k w_{hik} \, I(y_k \leq x)}{\sum_h \sum_i \sum_k w_{hik}}$$

where the function $I(y_k \leq x)$ is an indicator variable that takes the value one if $y_i \leq x$ and zero otherwise.

Once the CDF is estimated using the sample design weights, the q-th quantile of a variable y is defined as the smallest value of y for which the CDF is greater than or equal to q. In particular, the median corresponds to the value for which the CDF reaches or exceeds 0.5; therefore, the estimated median is the value where the estimated CDF is greater than or equal to 0.5. In this way, following the recommendations of Heeringa et al. [2017] for estimating quantiles in complex surveys, the observations are ordered in ascending order (order statistics) so that $y_{(1)} \geq y_{(2)} \geq ... \geq y_{(n)}$ and the value of j $(j = 1, ..., n)$ is identified such that:

$$\hat{F}\left(y_{(j)}\right) \leq q \leq \hat{F}\left(y_{(j+1)}\right)$$

Under this condition, the estimator of the q-th quantile is obtained through linear interpolation, which allows obtaining more stable and precise estimates of the quantiles, especially when the CDF presents important jumps associated with the use of sampling weights. Regarding the estimation of the variance of percentiles and medians, as well as the construction of confidence intervals, Kovar et al. [1988] carried out a simulation study in the context of complex designs and recommend the use of replication methods due to their good performance for this type of non-linear parameters.

Both the quantile estimators and the methods to estimate their variances are implemented in `R`. In particular, the `survey_median()` function allows us to obtain the median along with its standard error and confidence interval. The syntax for estimating median household expenditure using the example database is presented below, the results of which are shown in Table 3.16.

```
survey_design %>%
  summarise(median = survey_median(Expenditure,
                                   level = 0.95,
                                   vartype = c("se", "ci")
                                   )
            )
```

Table 3.16: Estimation of median household expenditures

median	median_se	median_low	median_upp
298	8.82	282	317

As can be seen, the arguments of the `survey_median()` function are equivalent to those previously used to estimate totals and means. As with other population parameters, the median can also be estimated for different study domains, such as geographic area, sex or age groups. Table 3.17 shows the results by geographic area.

```
survey_design %>%
  group_by(Zone) %>%
  summarise(median = survey_median(Expenditure,
                                   level = 0.95,
                                   vartype = c("se", "ci")
                                   )
            )
```

Table 3.17: Estimation of median expenses disaggregated by area

Zone	median	median_se	median_low	median_upp
Rural	241	11.0	214	258
Urban	381	19.8	337	416

If the goal is to estimate a specific percentile other than the median, for example the 25th percentile (first quartile), the `survey_quantile()` function can be used as follows. The corresponding results are presented in Table 3.18.

```
survey_design %>%
  summarise(quantile = survey_quantile(Expenditure,
                                   quantiles = 0.25,
                                   level = 0.95,
                                   vartype = c("se", "ci"),
                                   interval_type = "score"
                                   )
            )
```

Table 3.18: Estimation of the first quartile of household expenditures

quantile_q25	quantile_q25_se	quantile_q25_low	quantile_q25_upp
200	13.8	163	218

Note that the `interval_type = "score"` argument was added, which indicates that the confidence intervals are constructed using the *score* method based on the percentile influence function. This approach is recommended by Lumley [2010] for the analysis of data from complex sampling designs. Similarly, the estimate of the 25th percentile can be obtained for different subgroups of the population, such as gender or geographic area, using `group_by()`, as shown in Table 3.19.

```
survey_design %>%
  group_by(Zone, Sex) %>%
  summarise(quantile = survey_quantile(Expenditure,
                                       quantiles = 0.25,
                                       level = 0.95,
                                       vartype = c("se", "ci"),
                                       interval_type = "score"
                                       )
            )
```

Table 3.19: First quartile of expenses disaggregated by area and sex

Zone	Sex	quantile_q25	quantile_q25_se	quantile_q25_low	quantile_q25_upp
Rural	Female	160	7.18	135	163
Rural	Male	163	3.45	150	163
Urban	Female	258	10.40	249	291
Urban	Male	259	10.03	257	297

C Estimation of the Gini coefficient

The issue of economic inequality transcends the strictly statistical field and constitutes one of the central themes of contemporary social analysis. There is a broad consensus regarding the importance of understanding how economic inequalities condition people's opportunities, well-being and quality of life. In this sense, the rigorous measurement of inequality represents a fundamental input for the design, monitoring and evaluation of public policies aimed at social equity.

Among the most widely used indicators to quantify economic inequality is the Gini coefficient (G), which measures the degree of income concentration by comparing the observed distribution with a hypothetical situation of perfect equality. This coefficient takes values between 0 and 1, where $G = 0$ represents perfect equality and $G = 1$ corresponds to the maximum possible level

of inequality. Higher values indicate a greater concentration of income in a small group of the population.

In the context of household surveys, the estimation of the Gini coefficient must adequately incorporate the characteristics of the sample design, including expansion factors, stratification and conglomeration. In many cases, the sample weights are previously normalized in order to simplify the estimation procedures and facilitate computational processing. According to Binder and Kovacevic [1995], the Gini coefficient estimator can be expressed as:

$$\hat{G} = \frac{2\sum_h \sum_i \sum_k w^*_{hik}\, \hat{F}_{hik}\, y_{hik} - 1}{\hat{\bar{y}}}$$

where $w^*_{hik} = \dfrac{w_{hik}}{\sum_h \sum_i \sum_k w_{hik}}$ corresponds to the normalized sample weight, $\hat{F}_{hik}$ represents the estimated cumulative distribution function (CDF) for observation k within the cluster i of the h stratum, and $\hat{\bar{y}}$ denotes the weighted average of the income variable in the population.

Authors such as Osier [2009] and Langel and Tillé [2013] delve into additional technical aspects, especially as related to the estimation of the variance of the Gini coefficient under complex designs. The `convey` [Jacob et al., 2024] package implements the recommended procedures for calculating the Gini coefficient and its variance in household surveys. First, the sample design is prepared using `convey_prep()`. The `svygini()` function then calculates the Gini index using the input variable and the specified complex survey design. Below is the syntax to estimate it in the example database, presented in Table 3.20:

```
library(convey)
gini_design <- convey_prep(survey_design)
gini <- svygini(~Income, design = gini_design)

data.frame(Gini = coef(gini),
           standard_error = SE(gini),
           ci_lower = confint(gini)[1],
           ci_upper = confint(gini)[2])
```

Table 3.20: Estimation of the Gini coefficient for household income

	Gini	Income	ci_lower	ci_upper
Income	0.413	0.019	0.377	0.45

If the interest focuses on the estimation of the Lorenz curve, it is important to remember that, according to Kovacevic and Binder [1997], this represents the relationship between the accumulated percentage of the population (ordered from households with the lowest income to those with the highest income) and the accumulated proportion of the total income concentrated in said population. The 45-degree diagonal line corresponds to a hypothetical situation of perfect equality, in which all individuals share proportionally in the total income.

The area between the Lorenz curve and the diagonal of equality is known as the Lorenz area, and the Gini coefficient can be interpreted as twice this relative area. Consequently, the closer the Lorenz curve is to the diagonal, the greater the level of equity in the distribution of income; On the contrary, further curves reflect higher levels of economic inequality.

To create the Lorenz curve in R, the function `svylorenz()` is used, presented in Figure 3.1:

```
clorenz <- svylorenz(
  formula = ~Income,
  design = gini_design,
  quantiles = seq(0, 1, .05),
  alpha = .01
)
```

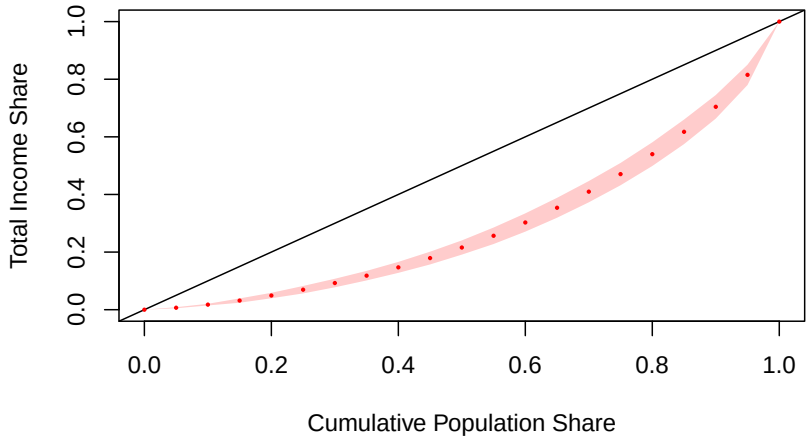

Figure 3.1: Estimated Lorenz curve for household income

The `formula = ~Income` argument specifies the income variable on which the cumulative distribution will be calculated. The argument `design = gini_design` indicates the previously defined sample design object, which contains the information related to weights, strata and clusters of the survey. For its part, `quantiles = seq(0, 1,.05)` defines the distribution points at which the Lorenz curve will be evaluated, generating percentiles from 0 to 1 in increments of 0.05. Finally, the `alpha =.01` argument establishes a significance level of 1%, which is equivalent to constructing 99% confidence intervals for the estimates associated with the curve. ### Correlation estimation {#estimacion-de-la-correlacion}

In the study of household surveys, in addition to describing variables individually, it is essential to analyze how they relate to each other. One of the most used tools for this purpose is the Pearson correlation coefficient, which measures the strength and direction of the linear relationship between two numerical variables. This coefficient takes values between -1 and 1. A positive value indicates that both variables tend to increase simultaneously, while a negative value indicates that when one variable increases, the other tends to decrease. For their part, values close to zero suggest the absence of a strong linear relationship between the analyzed variables.

For example, in a household survey it may be of interest to study whether there is an association between household income and their level of expenditure, as well as evaluate the magnitude of said relationship. This type of analysis allows us to better understand the economic and social patterns observed in the population.

In complex surveys, it is necessary to incorporate sampling weights so that the estimate is representative of the population. This adjustment takes into account stratification, clustering, and unequal selection probabilities. The weighted calculation involves evaluating the covariance between the two variables and dividing it by the product of their weighted standard deviations, thus eliminating the influence of the measurement units. Assuming that $\hat{\bar{x}}$ is an estimate of the population mean of the x variable, then the weight-adjusted Pearson correlation coefficient is expressed as:

$$\hat{\rho}_{xy} = \frac{\sum_h \sum_i \sum_k w_{hik}(y_{hik} - \hat{\bar{y}})(x_{hik} - \hat{\bar{x}})}{\sqrt{\sum_h \sum_i \sum_k w_{hik}(y_{hik} - \hat{\bar{y}})^2} \sqrt{\sum_h \sum_i \sum_k w_{hik}(x_{hik} - \hat{\bar{x}})^2}}$$

The `survey` package has the `svyvar()` function that allows obtaining weighted covariance matrices, from which the correlation can be calculated. Table 3.21 presents an example of estimation for the variance covariance matrix between household income and expenditure.

```
cov_matrix <- svyvar(~Income + Expenditure, design = survey_design)
```

Table 3.21: Weighted variance and covariance matrix between income and expenditure

	Variable	Income	Expenditure
Income	Income	243719	98337
Expenditure	Expenditure	98337	77887

The `svyvar` function estimates the variance and covariance matrix considering the sample design. From this matrix the correlation can be obtained by applying the following instructions:

```
cov_xy <- cov_matrix[1, 2]
sd_x <- sqrt(cov_matrix[1, 1])
sd_y <- sqrt(cov_matrix[2, 2])

weighted_cor <- cov_xy / (sd_x * sd_y)
weighted_cor
```

```
## [1] 0.714
```

If you also want to perform statistical inference on the correlation, for example obtaining standard errors or confidence intervals, you can use replication methods (bootstrap/jackknife) compatible with the survey package. Another option is to use the `svycor` function from the `jtools` package to directly estimate the correlation:

```r
library(jtools)

svycor(~Income + Expenditure, design = survey_design)
```

```
##              Income Expenditure
## Income         1.00        0.71
## Expenditure    0.71        1.00
```

II Hypothesis testing

In the analysis of data from household surveys, it is not enough to study each variable in isolation, such as estimating the average income of men and women in a country. It is also essential to compare groups and evaluate whether the differences observed between them reflect real inequalities in the population or if they could be explained simply by sampling error. For example, a frequently asked question is whether there are statistically significant differences in median income between male-headed households and female-headed households. This type of analysis makes it possible to identify social and economic gaps, in addition to generating useful evidence for the design and evaluation of public policies.

To answer these types of questions, hypothesis tests are used, that is, statistical procedures that allow statements about population parameters to be contrasted using the information observed in the sample. In the context of household surveys, these tests must adequately incorporate the characteristics of the sampling design in order to guarantee valid inferences and reliable results.

Hypothesis testing is a fundamental tool for evaluating statements about population parameters based on the information observed in a sample. Generally speaking, any hypothesis test is based on the contrast between two opposing propositions: the null hypothesis, denoted by H_0, and the alternative hypothesis, denoted by H_1. In the most common case of a two-sided test, these hypotheses are expressed as:

$$\begin{cases} H_0: & \theta = \theta_0 \\ H_1: & \theta \neq \theta_0 \end{cases}$$

where θ represents the population parameter of interest and θ_0 a reference value specified under the null hypothesis. Depending on the objective of the analysis, the alternative hypothesis can also be formulated unilaterally, for example, when you want to evaluate whether $\theta > \theta_0$ or whether $\theta < \theta_0$. The purpose of the test is to determine whether the evidence contained in the sample is strong enough to reject the null hypothesis in favor of the alternative hypothesis.

Suppose $\bar{y}_1$ and $\bar{y}_2$ represent the population means of a variable y in two different domains, for example, the mean income of male-headed households and the mean income of female-headed households. So, the parameter of interest then corresponds to the difference:

$$\Delta = \bar{y}_1 - \bar{y}_2$$

Its sample estimator is defined as:

$$\hat{\Delta} = \hat{\bar{y}}_1 - \hat{\bar{y}}_2$$

For which, its standard error is estimated by the following expression:

$$\widehat{se}(\hat{\Delta}) = \sqrt{\widehat{Var}(\hat{\bar{y}}_1) + \widehat{Var}(\hat{\bar{y}}_2) - 2\widehat{Cov}(\hat{\bar{y}}_1, \hat{\bar{y}}_2)}$$

The presence of the covariance term is important because the estimates of both domains usually come from the same sample and, therefore, are not necessarily independent. Once the estimator and its standard error have been calculated, the hypothesis contrast is carried out using the following test statistic:

$$t = \frac{\hat{\Delta}}{\widehat{se}(\hat{\Delta})} \sim t_{df}$$

which approximately follows a Student t distribution with (df) degrees of freedom. In practice, these degrees of freedom are usually approximated by $df = n_I - H$, where n_I represents the number of primary sampling units and H the number of strata. Under the null hypothesis, high absolute values of the t statistic constitute evidence against H_0. In a complementary manner, a confidence interval can also be constructed for the parameter Δ. For a confidence level of $(1-\alpha)100\%$, said interval is given by:

$$\hat{\Delta} \pm t_{(1-\alpha/2,df)}\, \hat{se}(\hat{\Delta}).$$

This interval provides a range of plausible values for the population difference and is a useful tool for evaluating both the magnitude and statistical significance of the observed differences between groups.

In `R`, these tests can be implemented using the `svyttest()` function of the `survey` package, which automatically incorporates the adjustments associated with the sample design. From the results presented in Table 3.22 it can be concluded that, with a confidence level of 95%, there is not sufficient statistical evidence to affirm that average incomes differ by sex.

```
ttest_sex <- svyttest(Income ~ Sex, design = survey_design, level = 0.95)
```

Table 3.22: Test of difference in average income by sex (total population)

statistic	p_value	gl	ci_lower	ci_upper	estimated_difference
1.36	0.176	118	-12.8	69.4	28.3

Another hypothesis of interest consists of evaluating whether the average household income differs depending on the sex of the head within the urban area, as presented in Table 3.23. The results indicate that the null hypothesis H_0 is not rejected, so there is not enough statistical evidence to affirm that average income differs by sex in the urban area.

```
urban_subset <- survey_design %>%
  filter(Zone == "Urban")

ttest_urban <- svyttest(Income ~ Sex, design = urban_subset, level = 0.95)
```

Table 3.23: Test of difference in average income by sex in urban areas

statistic	p_value	gl	ci_lower	ci_upper	estimated_difference
1.57	0.122	63	-12.3	102	44.7

The procedure described is not limited to means, but can also be applied to proportions, totals, ratios or any function differentiable from totals. In all cases, the contrast is based on the point estimate, its variance (including covariances when applicable) and the comparison with the t distribution adjusted to the sample design. ## Contrast estimation {#estimacion-de-contrastes}

In the analysis of household surveys it is common that the interest is not limited to comparing only two populations, but several simultaneously. For example, it may be necessary to compare the average household income between different regions, geographic areas or population groups, in order to identify differences and establish patterns of inequality between them. In these types of situations, the mean difference tests presented above are limited, since they are designed to only compare pairs of populations.

To address more general comparisons, contrasts are used, which constitute a flexible tool to evaluate linear combinations of population parameters. In general terms, a contrast is defined as:

$$f\left(\theta_1, \theta_2, \ldots, \theta_J\right) = \sum_{j=1}^{J} a_j \theta_j,$$

where the coefficients a_j are known constants and θ_j represent the parameters of interest. This approach allows a wide variety of comparisons between groups to be formulated and evaluated, including differences between means, multiple comparisons, and more complex combinations of parameters.

In `R`, methodological procedures for implementing contrasts in complex sampling designs are available through the `svycontrast()` function of the `survey` package. For example, if you want to compare the average income of two particular subpopulations (the Northern and Southern regions) you can use the contrast $\bar{y}_{Norte} - \bar{y}_{Sur}$. Since the sample covers five regions in total, the contrast must be constructed by assigning coefficients only to the regions involved in the comparison and leaving coefficients equal to zero for the other regions. In this way, the contrast is defined as:

$$1 \times \hat{\bar{y}}_{Norte} + (-1) \times \hat{\bar{y}}_{Sur} + 0 \times \hat{\bar{y}}_{Centro} + 0 \times \hat{\bar{y}}_{Occidente} + 0 \times \hat{\bar{y}}_{Oriente}.$$

This expression indicates that the contrast corresponds to the difference between the estimated means of the North and South regions, while the other regions do not participate in the comparison. In matrix form, the contrast can be written as:

$$[1, -1, 0, 0, 0] \times \begin{bmatrix} \hat{\bar{y}}_{Norte} \\ \hat{\bar{y}}_{Sur} \\ \hat{\bar{y}}_{Centro} \\ \hat{\bar{y}}_{Occidente} \\ \hat{\bar{y}}_{Oriente} \end{bmatrix}.$$

Consequently, the contrast vector associated with this comparison is [1, −1, 0, 0, 0], where the positive and negative coefficients indicate the populations to be compared and the zeros represent the populations excluded from the contrast. The first step is to calculate the estimated means for each region using `group_by()` and `survey_mean()`, as presented in Table 3.24. The visible results object is constructed with `srvyr`; Additionally, an auxiliary object with a covariance matrix is created for subsequent contrasts. As a result, the estimated averages of income by region are obtained.

```
region_mean <- survey_design %>%
  group_by(Region) %>%
  summarise(mean = survey_mean(Income,
                               na.rm = TRUE,
                               vartype = c("se", "ci")))

region_mean_contrast <- svyby(formula = ~Income,
                              by = ~Region,
                              design = survey_design,
                              FUN = svymean,
                              na.rm = TRUE,
                              covmat = TRUE,
                              vartype = c("se", "ci"))
```

Table 3.24: Estimated average income by region

Region	mean	mean_se	mean_low	mean_upp
Norte	552	55.4	443	662
Sur	626	62.4	502	749
Centro	651	61.5	529	772
Occidente	517	46.2	425	609
Oriente	542	71.7	400	684

The `svycontrast()` function returns the estimated contrast and its standard error. The arguments of this function are the averages of the estimated revenues (`stat`) and the contrast constants (`contrasts`), as shown in Table 3.25.

```
contrast_region_ns <- svycontrast(
  stat = region_mean_contrast,
  contrasts = list(north_south_difference = c(1, -1, 0, 0, 0))
)
```

Table 3.25: Contrast of average income: Northern region minus Southern region

contrast	estimate	variance	standard_error
north_south_difference	-73.4	6959	83.4

Note that these same results could be obtained by explicitly carrying out the process of constructing the contrast estimator and its estimated variance. If only the estimated average incomes of the North and South regions are considered, their difference is:

```
# Step 1: difference in estimates (North - South)
552.4 - 625.8
```

```
## [1] -73.4
```

The next step is to calculate the variance and covariance matrix and from there extract the variances and covariances of the North and South regions, presented in Table 3.26:

```
# Step 2: variance and covariance matrix
region_vcov <- vcov(region_mean_contrast)
```

Table 3.26: Variance and covariance matrix of average income by region

Region	Norte	Sur	Centro	Occidente	Oriente
Norte	3065	0	0	0	0
Sur	0	3894	0	0	0
Centro	0	0	3778	0	0
Occidente	0	0	0	2136	0
Oriente	0	0	0	0	5136

To calculate the standard error of the difference (contrast), the properties of the variance will be used, as follows:

$$\widehat{se}\left(\hat{\bar{y}}_{Norte} - \hat{\bar{y}}_{Sur}\right) = \sqrt{\widehat{Var}\left(\hat{\bar{y}}_{Norte}\right) + \widehat{Var}\left(\hat{\bar{y}}_{Sur}\right) - 2\,\widehat{Cov}\left(\hat{\bar{y}}_{Norte}, \hat{\bar{y}}_{Sur}\right)}$$

Therefore:

```
sqrt(2650 + 3460 - 2 * 0)
```

```
## [1] 78.2
```

Which corresponds to the same result obtained previously with the `svycontrast()` function. In addition to comparing only two populations, contrasts allow several comparisons of interest to be evaluated simultaneously between different groups.

On the other hand, suppose you want to compare average incomes between different geographic regions. In particular, contrasts such as $\bar{y}_{Norte} - \bar{y}_{Centro}$, $\bar{y}_{Sur} - \bar{y}_{Centro}$ and $\bar{y}_{Occidente} - \bar{y}_{Oriente}$ could be considered. Each of these expressions represents a specific linear contrast between regional means. Jointly, these contrasts can be organized using the following contrast matrix:

$$\begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

In this matrix, each row defines a different contrast and each column corresponds to one of the regions considered in the analysis. The positive and negative coefficients indicate the means that participate in each comparison, while the zeros represent the regions that do not participate in the corresponding contrast. Below is the implementation of the contrasts in R, the results of which are presented in Table 3.27. From these results, it can be concluded that the Southern and Central regions have the most similar average household incomes, given that the estimated difference between both regions is the smallest among the contrasts evaluated.

```
contrast_regions <- svycontrast(
  stat = region_mean_contrast,
  contrasts = list(
    north_south = c(1, 0, -1, 0, 0),
    south_center = c(0, 1, -1, 0, 0),
    west_east = c(0, 0, 0, 1, -1)
  )
)
```

Table 3.27: Contrasts in average income between regions

contrast	estimate	variance	standard_error
north_south	-98.4	6843	82.7
south_center	-25.0	7673	87.6
west_east	-24.7	7272	85.3

It is also possible to build contrasts between variables associated with different population groups, as occurs when comparing average income according to sex. Following the same procedure as the previous example, the first step is to estimate the average income for each group, in this case men and women, as presented in Table 3.28.

```
sex_mean <- survey_design %>%
  group_by(Sex) %>%
  summarise(mean = survey_mean(Income,
                               na.rm = TRUE,
```

```
                                vartype = c("se", "ci")))

sex_mean_contrast <- svyby(formula = ~Income,
                           by = ~Sex,
                           design = survey_design,
                           FUN = svymean,
                           na.rm = TRUE,
                           covmat = TRUE,
                           vartype = c("se", "ci"))
```

Table 3.28: Estimated average income by sex

Sex	mean	mean_se	mean_low	mean_upp
Female	558	25.8	506	609
Male	586	34.6	517	654

In this case, the contrast of interest is given by $\bar{y}_F - \bar{y}_M$, which allows quantifying the difference between the average income of women and that of men. Next, using the function `svycontrast()` of the `survey` package, the contrast estimation is obtained, the results of which are presented in Table 3.29. From these results, it is concluded that, on average, men receive 28.3 monetary units more than women, with an estimated standard error of 19.8 monetary units.

```
contrast_sex <- svycontrast(
  stat = sex_mean_contrast,
  contrasts = list(sex_difference = c(1, -1))
)
```

Table 3.29: Contrast of average income between women and men

contrast	estimate	variance	standard_error
sex_difference	-28.3	431	20.8

Since the contrasts correspond to linear functions of parameters, it is also possible to apply them to ratio estimators. An example of this is to compare the relationship between expenses and income according to sex. In this case, first the expense-income ratio is estimated for each group and subsequently the contrast between these ratios is constructed. Below are the computational codes used to perform this analysis, the results of which are shown in Table 3.30:

```
sex_ratio <- survey_design %>%
  group_by(Sex) %>%
  summarise(ratio = survey_ratio(Income,
                                 Expenditure,
```

```
                                  na.rm = TRUE,
                                  vartype = c("se", "ci")))

sex_ratio_contrast <- svyby(formula = ~Income,
                            by = ~Sex,
                            denominator = ~Expenditure,
                            design = survey_design,
                            FUN = svyratio,
                            na.rm = TRUE,
                            covmat = TRUE,
                            vartype = c("se", "ci"))
```

Table 3.30: Income/expenditure ratio disaggregated by sex

Sex	ratio	ratio_se	ratio_low	ratio_upp
Female	1.52	0.046	1.43	1.61
Male	1.56	0.070	1.43	1.70

The `svycontrast()` function is not limited only to the comparison of means or totals, but also allows the construction of contrasts between other parameters of interest, such as ratios and proportions estimated from complex sampling designs. The computational codes used to perform this analysis are presented below, the results of which are shown in Table 3.31. From these results, it can be concluded that the difference between the estimated ratios is 0.045 in favor of men.

```
contrast_sex_ratio <- svycontrast(
  stat = sex_ratio_contrast,
  contrasts = list(sex_difference = c(1, -1))
)
```

Table 3.31: Contrast of the income/expenditure ratio between sexes

contrast	estimate	standard_error	variance
sex_difference	-0.046	0.042	0.002

III Visualization of continuous variables

The graphical representation of continuous variables is an essential component in the analysis of household surveys. Well-designed graphs allow you to identify distributions, trends and relationships between variables, facilitating the interpretation of the findings and their communication to different audiences [Wilkinson, 2005]. A fundamental aspect in this context is

that the graphs must incorporate the sampling weights to reflect the distribution of the population of interest.

This section uses the `ggplot2` [Wickham et al., 2026b] and `patchwork` [Pedersen, 2025] packages for graph construction and composition, combined with `survey` [Lumley, 2024] and `srvyr` [Freedman Ellis and Schneider, 2024].

```
library(ggplot2)
library(patchwork)
data(BigCity, package = "TeachingSampling")
```

A Histograms

A histogram is a graphical representation of the distribution of a continuous numerical variable, constructed from rectangles whose height is proportional to the frequency of observations within certain class intervals. In the context of household surveys, weighted histograms allow us to approximate the distribution of the variable in the population, incorporating the expansion factors associated with the sampling design.

Below is the weighted histogram of the variable `Income`, constructed incorporating the sampling weights (`wk`) in order to represent the estimated distribution of income in the population. The results obtained are shown in Figure 3.2:

```
weighted_income_hist <- ggplot(
  data = survey_data,
  aes(x = Income, weight = wk)) +
  geom_histogram(aes(y = ..density..)) +
  ylab("") +
  ggtitle("Weighted histogram")
weighted_income_hist
```

In this code, the graph construction starts with the `survey_data` database. Within `aes()`, the argument `x = Income` indicates that the variable represented on the horizontal axis corresponds to income, while `weight = wk` incorporates the sampling weights to construct a weighted histogram that reflects the estimated distribution in the population. The function `geom_histogram()` generates the histogram, and using `aes(y = ..density..)`, the height of the bars is expressed in terms of density rather than absolute frequencies. `ylab("")` then removes the vertical axis label, and `ggtitle("Weighted histogram")` adds a descriptive title to the chart.

Histograms also allow you to visually compare the distribution of a variable between different subgroups of the population using the argument `fill`, which assigns a different color to each category. For example, it is possible to compare the income distribution between urban and rural areas, as shown in Figure 3.3:

```
zone_colors <- c(Urban = "#48C9B0", Rural = "#117864")

weighted_zone_hist <- ggplot(survey_data, aes(x = Income, weight = wk)) +
```

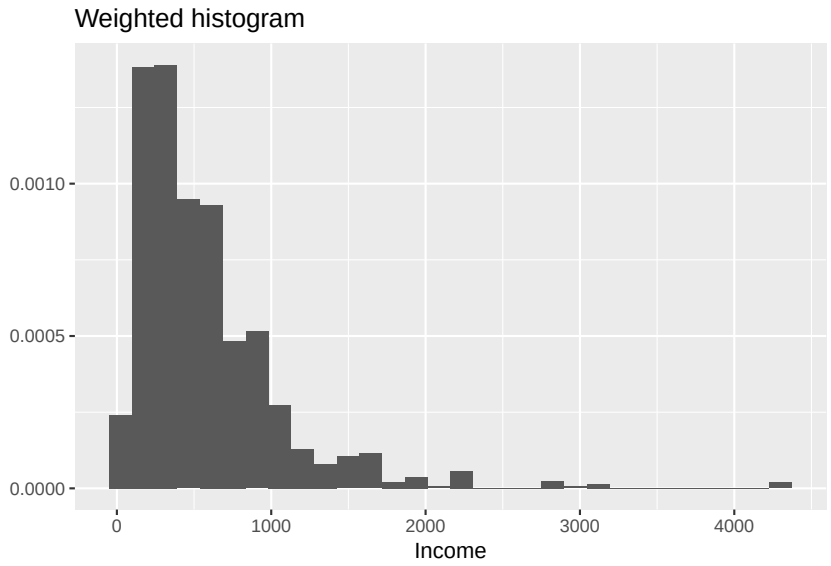

Figure 3.2: Weighted histogram of household income

```
  geom_histogram(
    aes(y = ..density.., fill = Zone),
    alpha = 0.5, position = "identity") +
  scale_fill_manual(values = zone_colors) +
  ylab("") +
  ggtitle("Weighted")

weighted_zone_hist
```

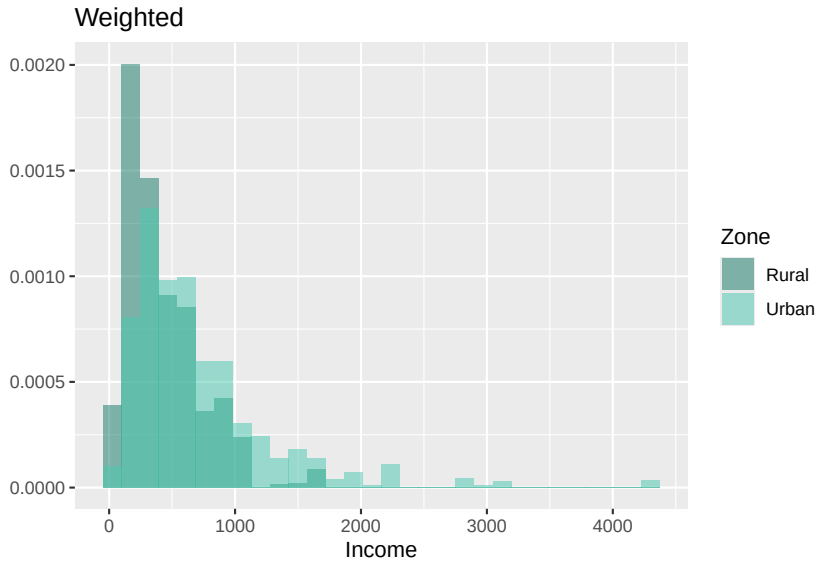

Figure 3.3: Weighted histogram of income by geographical area

In this code, a weighted histogram is constructed using the `survey_data` database, where `x =`

`Income` defines the variable represented on the horizontal axis and `weight = wk` incorporates the sample weights to reflect the population distribution of income. The `geom_histogram()` function generates the histogram and, through `aes(y =..density.., fill = Zone)`, represents the bars in terms of density and assigns different colors according to the `Zone` variable, allowing the income distribution to be visually compared between zones. The `alpha = 0.5` argument adds transparency to the bars to make it easier to overlay histograms, while `position = "identity"` allows the distributions to be drawn on top of each other rather than stacked. Subsequently, `scale_fill_manual(values = zone_colors)` manually defines the colors associated with each zone. Finally, `ylab("")` removes the vertical axis label, `ggtitle("Weighted")` adds a title to the chart.

Additionally, it is possible to superimpose smoothed density curves using the `geom_density()` function, which allows the shape of the distribution of the analyzed variable to be continuously displayed. This type of representation facilitates the identification of patterns such as asymmetries, concentration of observations or the presence of multiple modes. An example of this type of graph is presented in Figure 3.4:

```
weighted_income_hist + geom_density(fill = "blue", alpha = 0.3) |
  weighted_zone_hist + geom_density(aes(fill = Zone), alpha = 0.3) +
  theme(legend.position = "top")
```

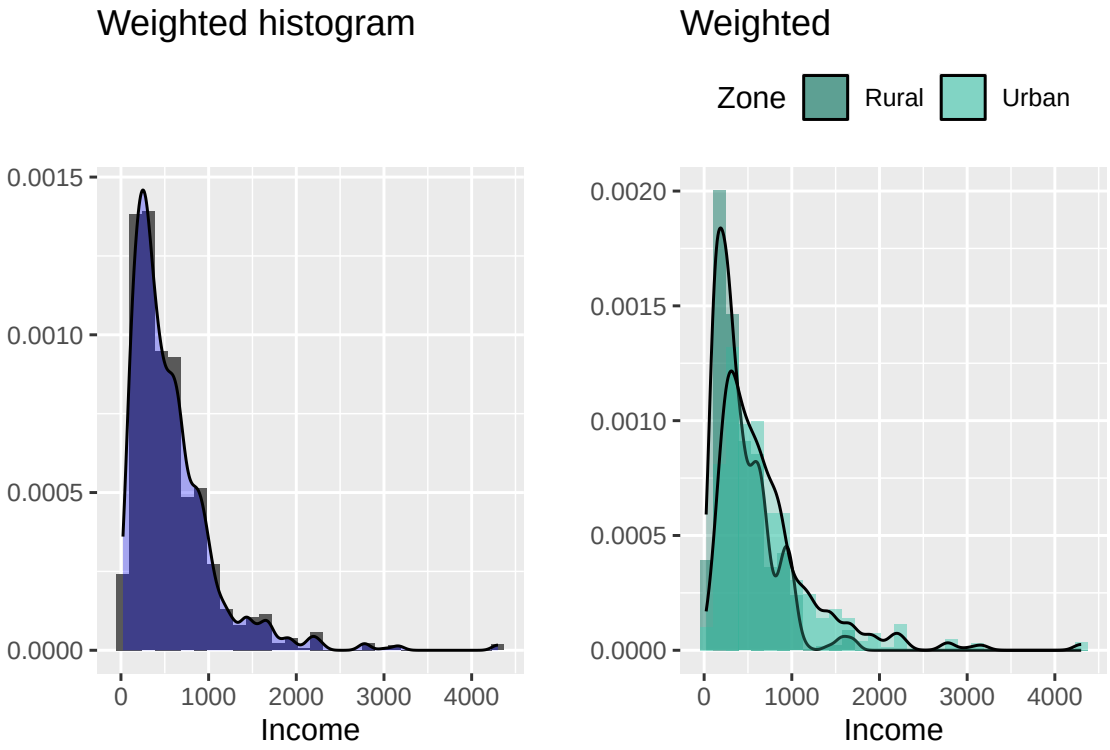

Figure 3.4: Histograms with superimposed density curves for income and expenditure

B Box plots (*boxplots*)

Boxplots allow you to identify outliers and evaluate the dispersion of the distribution. Introduced by John Tukey in 1977, they are a graphic representation that summarizes the distribution of a variable using statistics based on quartiles, allowing the median, dispersion of the data and the presence of outliers to be identified. In the context of household surveys, this type of graph is useful for comparing distributions between different population groups in a compact and visually intuitive way.

To construct *boxplots* weighted in `ggplot2` the function `geom_boxplot()` is used, as shown in Figure 3.5. In this case, the function `ggplot()` starts the construction of the graph using the `survey_data` database, where `x = Income` defines the numerical variable analyzed and `weight = wk` incorporates the sampling weights to represent the estimated distribution in the population. Subsequently, `geom_boxplot()` generates the box plots and, using `aes(fill = Zone)` or `aes(fill = Sex)`, assigns different colors to each category of the variables `Zone` and `Sex`, respectively, facilitating visual comparison between groups. The `coord_flip()` function inverts the axes to present the *boxplots* horizontally, thus improving their readability. For its part, `scale_fill_manual()` allows you to manually define the colors associated with each category using the `zone_colors` and `sex_colors` vectors.

```
weighted_zone_boxplot <- ggplot(survey_data, aes(x = Income, weight = wk)) +
  geom_boxplot(aes(fill = Zone)) +
  ggtitle("Weighted") +
  coord_flip() +
  scale_fill_manual(values = zone_colors)

sex_colors <- c(Male = "#5DADE2", Female = "#2874A6")

weighted_sex_boxplot <- ggplot(survey_data, aes(x = Income, weight = wk)) +
  geom_boxplot(aes(fill = Sex)) +
  ggtitle("Weighted") +
  coord_flip() +
  scale_fill_manual(values = sex_colors)

weighted_zone_boxplot | weighted_sex_boxplot
```

In addition to the tools available in `ggplot2`, the `survey` package offers specialized functions such as `svyhist()` and `svyboxplot()`, which directly incorporate the features of complex sample design into the construction of the graphs. These functions allow us to obtain graphical representations consistent with the survey design, facilitating the exploratory analysis of the numerical variables. An example of its application is presented in Figure 3.6.

```
par(mfrow = c(1, 2))
svyhist(~Income, survey_design,
  main = "Household income",
  col  = "grey80", xlab = "Income",
  probability = FALSE)
```

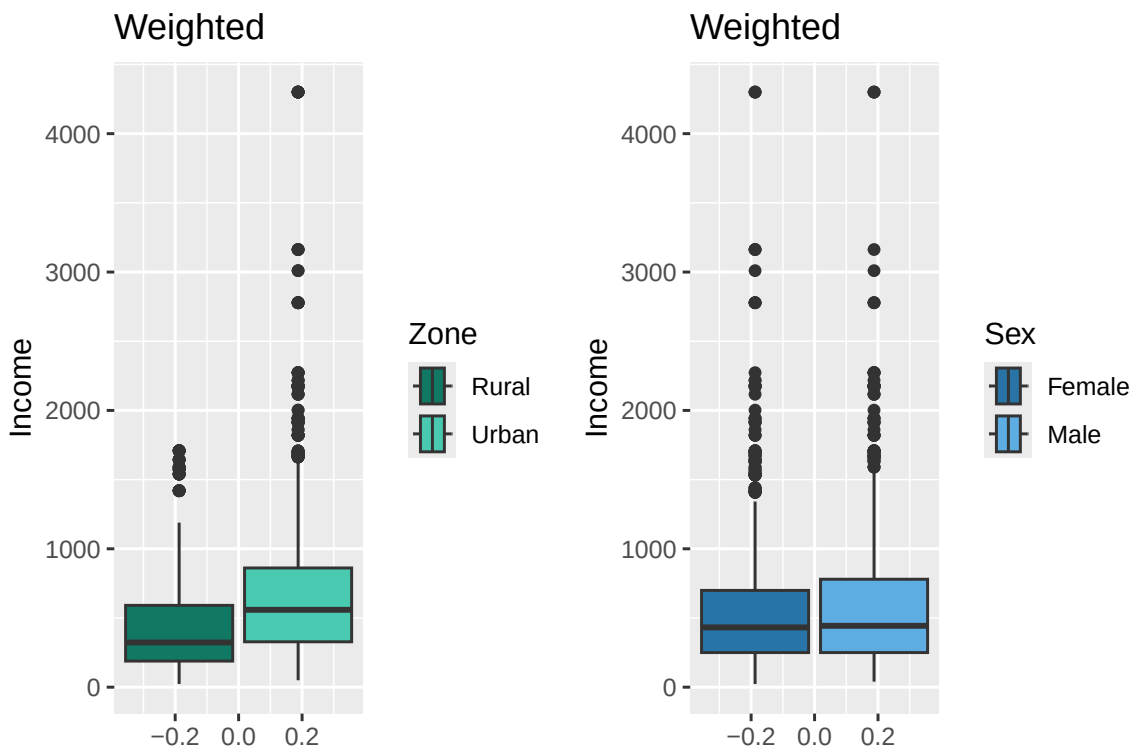

Figure 3.5: Weighted boxplots of income by area and sex

```
svyboxplot(Income ~ Zone, survey_design,
  col  = "grey80",
  ylab = "Income", xlab = "Zone")
```

C Scatter plots

Scatter plots allow you to visually explore the relationship between two continuous variables, facilitating the identification of patterns of association, trends, and possible outliers. In the context of surveys with complex sample designs, it is important to incorporate information on sampling weights so that the graphical representation adequately reflects the relative contribution of each observation in the population.

According to Lumley [2010], when working with large data sets, plotting all points simultaneously can result in graphs that are difficult to interpret due to overlapping observations. However, in moderately sized samples, a simple and effective strategy is to represent the weights by the size of the symbols in the scatterplot. In this way, the observations with the greatest population weight appear visually highlighted. An example of this type of representation is presented in Figure 3.7:

```
weighted_basic_scatter <- ggplot(
  data = survey_data,
  aes(y = Income, x = Expenditure)) +
```

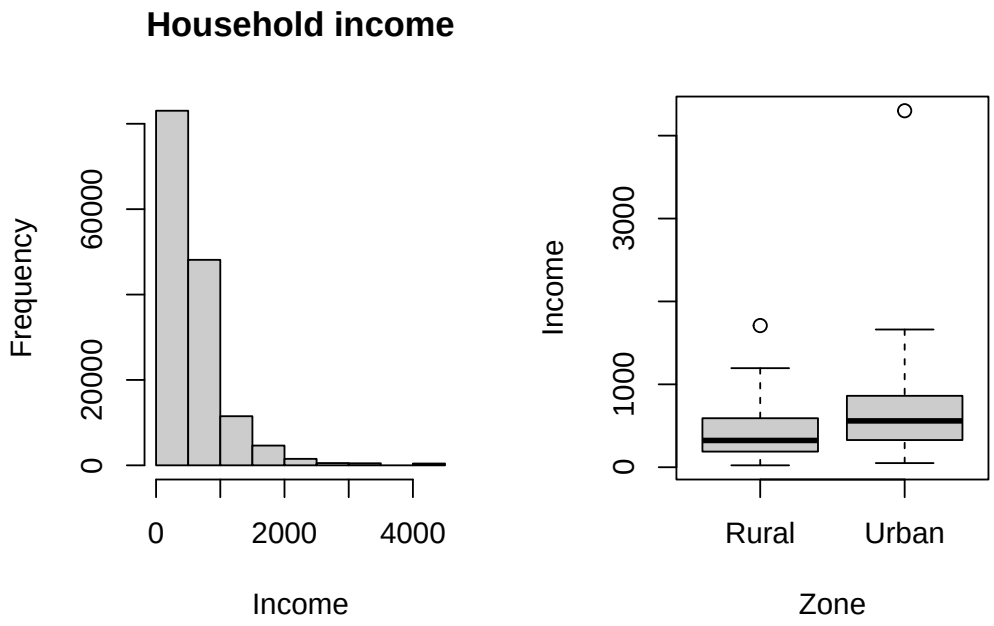

Figure 3.6: Histogram and boxplot of income adjusted to the complex sample design

```
  geom_point(aes(size = wk), alpha = 0.3)

weighted_basic_scatter
```

In this code, a scatterplot is constructed between the variables `x = Expenditure` and `y = Income`. The `geom_point()` function adds the points on the graph and, using `aes(size = wk)`, uses the sampling weights (`wk`) to control the size of each point, so that observations with greater population representation appear visually more prominent. The `alpha = 0.3` argument incorporates transparency to the points, reducing the overlapping problem and facilitating the visualization of areas with a high concentration of observations.

Additionally, it is possible to incorporate grouping variables in the scatter diagrams using the arguments `shape` and `color`, which allow observations to be visually differentiated according to specific categories of the population. This facilitates the identification of differentiated patterns between groups and improves the interpretation of the relationship between the analyzed variables. An example of this type of representation is presented in Figure 3.8.

```
weighted_zone_scatter <- ggplot(
  data = survey_data,
  aes(y = Income, x = Expenditure, shape = Zone)) +
  geom_point(aes(size = wk, color = Zone), alpha = 0.3) +
  labs(size = "Weight") +
  scale_color_manual(values = zone_colors)

weighted_zone_scatter
```

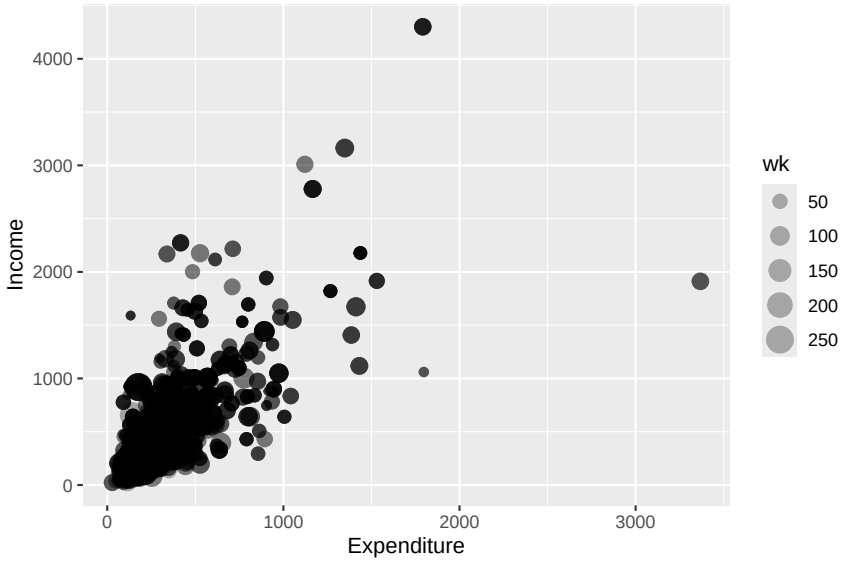

Figure 3.7: Scatter diagram between household income and expenditure

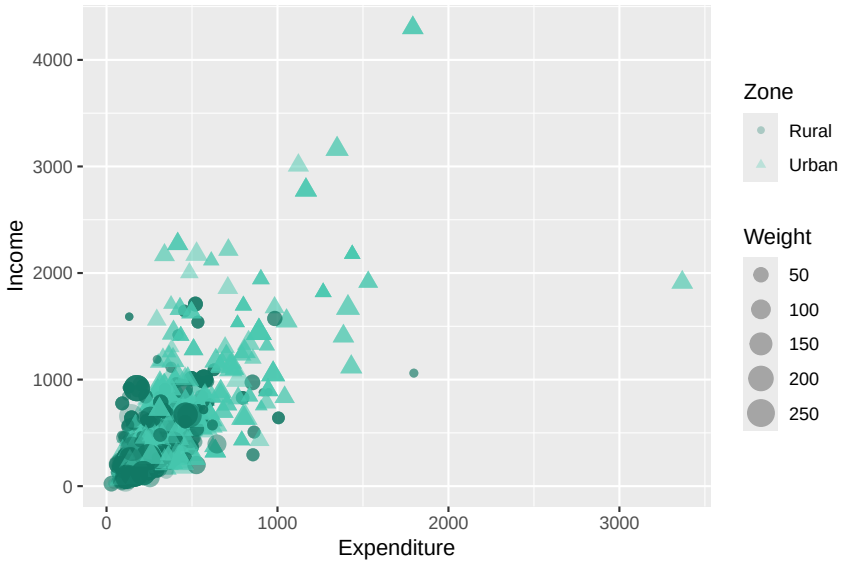

Figure 3.8: Scatter diagram between income and expenditure by geographic area

Chapter 4

Analysis of categorical variables

In household survey analysis, one of the most common outputs is the estimation of descriptive parameters associated with categorical variables, which make it possible to summarize the main characteristics of the population. These measures provide a clear and understandable representation of social phenomena based on information collected through probability samples.

The descriptive indicators most commonly used to analyze categorical variables include frequencies and proportions. Frequencies indicate how many people or households belong to each category, for example, the number of people living in poverty, while proportions express the relative weight of each category within the population as a whole.

Although descriptive analysis of categorical variables usually relies on these basic parameters, it is also possible to combine them with measures derived from numerical variables, such as quantiles or inequality indicators, which are discussed in detail in Chapter 4. However, in this chapter we focus exclusively on describing and estimating parameters associated with categorical variables using complex sample designs.

The analysis begins by defining the sampling design, following the guidelines explained in previous chapters and using the same database. This step is essential because it makes it possible to properly incorporate the characteristics of the survey's complex design, such as expansion factors, primary sampling units, and selection strata. Correctly specifying these elements ensures that the estimates obtained are representative of the target population and that the measures of precision appropriately reflect the variability introduced by the sampling scheme.

```
library(tidyverse)
library(survey)
library(srvyr)

survey_data <- readRDS("Data/encuesta.rds")
options(survey.lonely.psu = "adjust")

survey_design <- survey_data %>%
  as_survey_design(
    strata = Stratum,
    ids = PSU,
    weights = wk,
```

```
    nest = TRUE
  )
```

Next, several categorical variables derived from the original survey information are generated and used in the exercises developed throughout this chapter. Among them, a variable is constructed to identify whether or not the person is living in poverty. In addition, other categorical variables of analytical interest are also created to support disaggregations and comparisons across different population groups:

```
survey_design <- survey_design %>%
  mutate(
    poor = ifelse(Poverty != "NotPoor", 1, 0),
    unemployed = ifelse(Employment == "Unemployed", 1, 0),
    age_18 = case_when(
      Age < 18 ~ "< 18 years",
      TRUE ~ ">= 18 years"
    )
  )
```

The previous code creates new derived variables using the `mutate()` function from the `dplyr` package [Wickham et al., 2026c]. The variables `poor` and `unemployed` are constructed with the `ifelse()` function, which evaluates a logical condition and assigns a value depending on whether the condition is met. In this case, the variables take the value `1` when the person is living in poverty or is unemployed, respectively, and `0` otherwise. The variable `age_18`, in turn, is generated with the `case_when()` function, which makes it possible to define categories from one or more conditions more clearly and flexibly than `ifelse()`. This function is especially useful when categorical variables with multiple levels need to be constructed. In the example, people under 18 years of age are classified in the `"< 18 years"` category, while the rest are grouped in the `">= 18 years"` category.

In addition, it is necessary to generate population subgroups to produce disaggregated estimates. This chapter uses four example subpopulations:

```
urban_subset <- survey_design %>%
  filter(Zone == "Urban")

rural_subset <- survey_design %>%
  filter(Zone == "Rural")

female_subset <- survey_design %>%
  filter(Sex == "Female")

male_subset <- survey_design %>%
  filter(Sex == "Male")
```

I Estimation of population size

In household survey analysis, estimating the size of subpopulations is essential. This size is understood as the number of people or households that belong to specific categories and the proportion they represent within the population. These estimates, based on categorical variables, make it possible to characterize the demographic and socioeconomic profile of the population, which is key information for guiding resource allocation, designing public policies, and formulating social programs. For example, estimating how many people are below the poverty line, how many are unemployed, or how many have reached a given educational level is essential for identifying gaps and evaluating well-being conditions.

The size of a population or subpopulation is estimated from categorical variables, which segment the population into mutually exclusive groups (partitions). Population size refers to the total number of individuals or households in the survey database that belong to a given category. These categories may correspond, for example, to employment status or educational attainment. To obtain these estimates, respondents' answers are combined with the sampling weights, which indicate how many people or households each sample unit represents in the total population.

The population size estimator is defined as:

$$\hat{N} = \sum_h \sum_i \sum_k w_{hik}$$

Where w_{hik} is the weight or expansion factor of unit k in PSU i of stratum h. Similarly, estimating the size of a subpopulation follows the same principle, but is framed within a subset defined by a specific characteristic. To determine how many people belong to a particular category, that group is identified in the database and its sampling weights are summed. This makes it possible to quantify specific groups of interest and determine their size within the population:

$$\hat{N}_d = \sum_h \sum_i \sum_k w_{hik}\ I(y_{hik} = d)$$

Where $I(y_{hik} = d)$ is a binary variable that takes the value 1 if unit k in PSU i in stratum h belongs to category d of the variable of interest y, and 0 otherwise. Also note that if d was used in the calibration of the weights, the value of $\hat{N}_d$ will coincide with the external control applied.

In `R`, using the sampling design defined previously, it is possible to generate summaries of the survey information for each category of the `Zone` variable. To do this, the data are grouped by geographic area, which makes it possible to obtain separate results for each of these study domains. This procedure produces two main types of results. First, the number of observations actually collected in the sample is calculated without considering expansion factors, corresponding to the available sample size in each zone. Second, the population size associated with each domain is estimated by incorporating the sampling design information.

Together with the population estimates, measures of precision such as the standard error and confidence interval are also obtained. These make it possible to assess the level of uncertainty associated with the results produced from the sample. The results are presented in Table 4.1. In that table, `n` represents the number of sample observations in each zone, while `size` corresponds to the estimated population total of the subpopulation. The `unweighted()` function is used to calculate unweighted summaries directly from the observed sample data.

```
survey_design %>%
  group_by(Zone) %>%
  summarise(
    n = unweighted(n()),
    size = survey_total(vartype = c("se", "ci"))
  )
```

Table 4.1: Estimated population size by geographic zone

Zone	n	size	size_se	size_low	size_upp
Rural	1297	72102	3062	66039	78165
Urban	1308	78164	2847	72526	83802

Indeed, the sample size was 1,297 people in the rural zone and 1,308 in the urban zone. Based on these samples, the population was estimated at 72,102 people for the rural zone, with a standard error of 3,062, and 78,164 people for the urban zone, with a standard error of 2,847. These results reflect not only the estimated population size in each domain, but also the level of precision associated with the estimates obtained from the sampling design.

Using a 95% confidence level, the confidence intervals for the population estimates ranged from 66,038.5 to 78,165.4 people in the rural zone and from 72,526.2 to 83,801.7 people in the urban zone. Similarly, it is also possible to estimate the number of people by different poverty levels; these results are presented in Table 4.2.

```
survey_design %>%
  group_by(Poverty) %>%
  summarise(size = survey_total(vartype = c("se", "ci")))
```

Table 4.2: Estimated population size by poverty status

Poverty	size	size_se	size_low	size_upp
NotPoor	91398	4395	82696	100101
Extreme	21519	4949	11719	31319
Relative	37349	3695	30032	44666

Another relevant categorical variable in household surveys is employment status. The corresponding code is presented below, and its results are shown in Table 4.3. In this case, the information is grouped according to the categories of the `Employment` variable, making it possible to obtain separate estimates for each labor-force status of the population. Using the `group_by()` function, estimates can be disaggregated into more detailed levels, facilitating comparative analysis across occupational groups.

Before producing the estimates, records with missing values in the employment variable are excluded, ensuring that the results are calculated only with valid information. Then, for each

employment category, population size is estimated using the sampling design defined previously. In addition to the point estimate, measures of precision are also calculated, specifically the standard error and confidence intervals, making it possible to assess the degree of uncertainty associated with each estimate.

```
survey_design %>%
  filter(!is.na(Employment)) %>%
  group_by(Employment) %>%
  summarise(size = survey_total(vartype = c("se", "ci")))
```

Table 4.3: Estimated population size by employment status

Employment	size	size_se	size_low	size_upp
Unemployed	4635	761	3129	6141
Inactive	41465	2163	37183	45748
Employed	61877	2540	56847	66907

Based on the results obtained, an estimated 4,634.8 people are unemployed, with a 95% confidence interval between 3,128.6 and 6,140.9 people. Similarly, an estimated 41,465.2 people belong to the inactive population, with a confidence interval between 37,182.6 and 45,747.8. Finally, the employed population was estimated at 61,877.0 people, with a confidence interval from 36,784.2 to 47,793.5 people.

Finally, estimates can also be disaggregated simultaneously by more than one categorical variable. For example, it is possible to analyze employment status by poverty level; the results are presented in Table 4.4. Among other findings, these results show that approximately 44,600.3 employed people are not living in poverty, with a 95% confidence interval between 39,459.6 and 49,741.0 people. Similarly, an estimated 6,421.8 inactive people are living in extreme poverty, with a 95% confidence interval between 3,806.6 and 9,037.0 people.

```
survey_design %>%
  filter(!is.na(Employment)) %>%
  group_by(Employment, Poverty) %>%
  cascade(
    size = survey_total(vartype = c("se", "ci")),
    .fill = "Total"
  )
```

Table 4.4: Estimated population size by employment status and poverty status

Employment	Poverty	size	size_se	size_low	size_upp
Unemployed	NotPoor	1768	405	966	2571

Employment	Poverty	size	size_se	size_low	size_upp
Unemployed	Extreme	1169	348	480	1859
Unemployed	Relative	1697	458	791	2604
Unemployed	Total	4635	761	3129	6141
Inactive	NotPoor	24346	1736	20908	27784
Inactive	Extreme	6422	1321	3807	9037
Inactive	Relative	10697	1460	7806	13589
Inactive	Total	41465	2163	37183	45748
Employed	NotPoor	44600	2596	39460	49741
Employed	Extreme	5128	1122	2907	7349
Employed	Relative	12149	1347	9483	14816
Employed	Total	61877	2540	56847	66907
Total	Total	107977	3466	101115	114839

II Estimation of proportions

Proportions express the relative weight of specific groups within the population. For example, knowing the percentage of households below the poverty line is essential for assessing inequalities and characterizing well-being conditions. To obtain this type of indicator, the weighted mean of the dichotomous variable is calculated, ensuring that the estimate adequately represents the population distribution. According to Heeringa et al. [2017], when the original categories are converted into indicator variables, the estimated proportion is calculated as:

$$\hat{p}_d = \frac{\hat{N}_d}{\hat{N}} = \frac{\sum_h \sum_i \sum_k w_{hik}\, I(y_{hik} = d)}{\sum_h \sum_i \sum_k w_{hik}}$$

Although this estimator is conceptually simple, it is a nonlinear function of population totals. For this reason, deriving its statistical properties, particularly its variance and standard error, requires approximation methods such as Taylor linearization. In this context, the function $z_{hik} = I(y_{hik} = d) - \hat{p}_d$ is used. In practice, statistical software implements these procedures automatically and produces proportion estimates together with their corresponding standard errors and confidence intervals.

When proportions take values very close to 0 or 1, the precision of the estimates may be affected, so in many cases larger sample sizes are needed to obtain stable and reliable results. This situation is especially relevant in the analysis of rare or highly concentrated phenomena, where small variations in the sample can produce important changes in the estimates.

In addition, confidence intervals constructed using normal approximations may have lower limits below 0 or upper limits above 1, making them difficult to interpret because they are proportions. To avoid this problem, a widely used alternative is to apply the logit transformation before constructing the confidence intervals. This procedure ensures that, after returning to the original scale, the limits obtained remain within the valid range for proportions.

$$CI(\hat{p}_d; 1-\alpha) = \frac{exp\left[ln\left(\frac{\hat{p}_d}{1-\hat{p}_d}\right) \pm \frac{\widehat{me}}{\hat{p}_d(1-\hat{p}_d)}\right]}{1 + exp\left[ln\left(\frac{\hat{p}_d}{1-\hat{p}_d}\right) \pm \frac{\widehat{me}}{\hat{p}_d(1-\hat{p}_d)}\right]}$$

Where $\widehat{me}$ corresponds to the estimated margin of error and is obtained as the product of the critical value from Student's t distribution and the estimated standard error of the proportion $\hat{p}_d$. The term $t_{1-\alpha/2,df}$ represents the percentile associated with the selected confidence level, considering df degrees of freedom. In the context of complex surveys, these degrees of freedom are usually defined as the difference between the number of primary sampling units (PSUs) and the number of strata present in the subgroup or analysis domain.

Unlike other statistical software that presents results as percentages, `R` reports proportions on the [0,1] scale. The functions in these packages make it possible to calculate the estimates, their standard errors, and confidence intervals directly, facilitating the analysis of categorical variables in household surveys. The following example illustrates how to obtain the proportion of people by zone using the sampling design defined previously, presented in Table 4.5:

```
survey_design %>%
  group_by(Zone) %>%
  summarise(proportion = survey_mean(
    vartype = c("se", "ci"),
    proportion = TRUE
  ))
```

Table 4.5: Proportion of people by geographic zone

Zone	proportion	proportion_se	proportion_low	proportion_upp
Rural	0.48	0.014	0.452	0.508
Urban	0.52	0.014	0.492	0.548

In this example, the `survey_mean()` function is used to calculate the estimate of population proportions and, through the `proportion = TRUE` argument, it specifies that the analysis focuses on estimating the relative distribution of the population across the different categories of the variable analyzed. The results indicate that approximately 47.9% of people live in the rural zone, with a 95% confidence interval between 45.2% and 50.7%, while 52.0% corresponds to the population living in the urban zone, with a confidence interval between 49.2% and 54.7%.

The `survey` library also provides the `survey_prop()` function, designed specifically for estimating proportions in complex surveys. This function produces results equivalent to those obtained with `survey_mean()`, while also offering a more direct and intuitive syntax when the objective of the analysis is exclusively to calculate proportions. The results obtained with this alternative are presented in Table 4.6.

These estimates show that, in the urban zone, 53.6% of the population is female, with a 95% confidence interval between 51.0% and 56.2%, while 46.4% is male, with a confidence interval between 43.7% and 48.9%. These results make it possible to examine not only the percentage distribution of the population by sex, but also the level of precision associated with the estimates.

```
survey_design %>%
  group_by(Zone) %>%
  summarise(proportion = survey_prop(vartype = c("se", "ci")))
```

Table 4.6: Proportion of people by geographic zone (survey_prop function)

Zone	proportion	proportion_se	proportion_low	proportion_upp
Rural	0.48	0.014	0.452	0.508
Urban	0.52	0.014	0.492	0.548

If the focus is now on estimating proportions for specific subpopulations, for example, the proportion of men and women living in the urban zone, the analysis must be restricted to that study domain and the corresponding estimates then calculated for each sex category. The corresponding computational code is presented in Table 4.7.

```
urban_subset %>%
  group_by(Sex) %>%
  summarise(proportion = survey_prop(vartype = c("se", "ci")))
```

Table 4.7: Proportion of men and women in the urban zone

Sex	proportion	proportion_se	proportion_low	proportion_upp
Female	0.537	0.013	0.511	0.563
Male	0.463	0.013	0.437	0.489

Similarly, the same exercise can be carried out for the population living in the rural zone by estimating the proportion of men and women. This type of disaggregation makes it possible to compare the population composition by sex between urban and rural areas, providing a more detailed view of the demographic characteristics of the surveyed population. The corresponding results are presented in Table 4.8.

```
rural_subset %>%
  group_by(Sex) %>%
  summarise(
    n = unweighted(n()),
    proportion = survey_prop(vartype = c("se", "ci"))
  )
```

Table 4.8: Proportion of men and women in the rural zone

Sex	n	proportion	proportion_se	proportion_low	proportion_upp
Female	679	0.516	0.008	0.500	0.533
Male	618	0.484	0.008	0.467	0.500

Now, if the analysis is restricted only to the male population included in the database and the interest is to estimate the percentage distribution of men by zone of residence, it is possible to calculate the proportion of men living in urban and rural areas using the sampling design defined previously. The corresponding computational code is presented in Table 4.9.

```
male_subset %>%
  group_by(Zone) %>%
  summarise(proportion = survey_prop(vartype = c("se", "ci")))
```

Table 4.9: Distribution by zone in the male subpopulation

Zone	proportion	proportion_se	proportion_low	proportion_upp
Rural	0.491	0.018	0.455	0.526
Urban	0.509	0.018	0.474	0.545

If results with several levels of disaggregation are to be estimated, as in previous chapters, the use of the `group_by()` function makes it possible to generate different aggregation levels by combining two or more categorical variables. For example, it is possible to estimate the proportion of men by zone of residence and poverty status simultaneously. This type of analysis makes it possible to characterize the socioeconomic conditions of specific subpopulations in greater detail. The corresponding procedure is presented in Table 4.10.

```
male_subset %>%
  group_by(Zone, Poverty) %>%
  summarise(proportion = survey_prop(vartype = c("se", "ci")))
```

Table 4.10: Proportion of men by zone and poverty status

Zone	Poverty	proportion	proportion_se	proportion_low	proportion_upp
Rural	NotPoor	0.549	0.063	0.424	0.668
Rural	Extreme	0.198	0.067	0.096	0.364
Rural	Relative	0.254	0.037	0.187	0.334
Urban	NotPoor	0.660	0.037	0.584	0.728
Urban	Extreme	0.113	0.025	0.073	0.171

Zone	Poverty	proportion	proportion_se	proportion_low	proportion_upp
Urban	Relative	0.227	0.026	0.180	0.283

Based on the results obtained, 19.7% of men in the rural zone are living in extreme poverty, while in the urban zone this proportion is 11.3%. Although the point estimates suggest a higher incidence of extreme poverty in rural areas, comparison of the confidence intervals indicates that there is not enough evidence to conclude that the differences are statistically significant at the confidence level considered, because the two intervals overlap.

It is also possible to categorize quantitative variables to facilitate their joint analysis with other categorical variables. For example, age can be grouped into specific age ranges and then crossed with variables related to employment status. This type of procedure is useful for identifying differences in labor participation across age groups.

Below, the age variable is classified into two categories for the female population: women between 18 and 35 years of age, and women belonging to other age ranges. These categories are then analyzed together with the employability variable, making it possible to estimate the distribution of labor status by age group. The corresponding results are presented in Table 4.11.

```
female_subset %>%
  filter(!is.na(Employment)) %>%
  mutate(age_range = case_when(
    Age >= 18 & Age <= 35 ~ "18 - 35",
    TRUE ~ "Other"
  )) %>%
  group_by(age_range, Employment) %>%
  summarise(proportion = survey_prop(vartype = c("se", "ci")))
```

Table 4.11: Proportion of women by age range and employment status

age_range	Employment	proportion	proportion_se	proportion_low	proportion_upp
18 - 35	Unemployed	0.029	0.009	0.015	0.054
18 - 35	Inactive	0.517	0.038	0.442	0.591
18 - 35	Employed	0.455	0.036	0.385	0.526
Other	Unemployed	0.016	0.007	0.007	0.036
Other	Inactive	0.571	0.030	0.511	0.629
Other	Employed	0.413	0.029	0.356	0.472

III Contrasts and differences in proportions

According to Heeringa et al. [2017], the proportions obtained in the rows of a two-way table can be interpreted as estimates for different subpopulations, defined from the levels of a categorical

variable. In this context, it is relevant not only to estimate the proportions associated with each group, but also to evaluate the differences between them using linear contrasts.

For example, suppose the interest is in comparing the proportion of women living in poverty with the proportion of men in the same situation. Formally, this contrast can be expressed as $\hat{p}_F - \hat{p}_M$, where $\hat{p}_F$ represents the estimated proportion of women in poverty and $\hat{p}_M$ the estimated proportion of men in that condition. To construct this contrast, the proportions of men and women in poverty are first estimated using the same procedure described in previous chapters. The corresponding results are presented in Table 4.12:

```
sex_poverty_proportion <- survey_design %>%
  group_by(Sex) %>%
  summarise(proportion = survey_mean(poor,
                                     na.rm = TRUE,
                                     vartype = c("se", "ci")))

sex_poverty_contrast <- svyby(
  formula = ~poor,
  by = ~Sex,
  design = survey_design,
  FUN = svymean,
  na.rm = TRUE,
  covmat = TRUE,
  vartype = c("se", "ci")
)
```

Table 4.12: Proportion living in poverty by sex

Sex	proportion	proportion_se	proportion_low	proportion_upp
Female	0.389	0.032	0.327	0.452
Male	0.395	0.037	0.322	0.467

Once the proportions of men and women living in poverty have been estimated, it is possible to calculate the difference between the two proportions together with its corresponding standard error. In this case, the estimated difference is obtained by subtracting the proportion of men in poverty from the proportion of women in the same condition:

```
0.3892 - 0.3946
```

```
## [1] -0.0054
```

To correctly calculate the variability associated with this difference, it is not enough to consider only the variances of each estimate separately. It is also necessary to incorporate the covariance between the two proportions, since the estimates come from the same sample and therefore are not independent. The variance-covariance matrix can be obtained with the `vcov` function; the results are presented in Table 4.13:

```
vcov(sex_poverty_contrast) %>%
  data.frame()
```

Table 4.13: Variance-covariance matrix of the poverty proportion by sex

	Female	Male
Female	0.000998	0.000918
Male	0.000918	0.001342

Based on this matrix, the standard error of the difference in proportions is estimated using the general expression for the variance of a difference between estimators, which corresponds to the sum of the variances of each estimate minus twice the covariance between them. In this case, the calculation is carried out as follows:

```
sqrt(0.0009983 + 0.0013416 - 2 * 0.0009183)
```

```
## [1] 0.0224
```

However, the `survey` package makes it possible to carry out this procedure directly using the `svycontrast` function, which calculates both the linear contrast and its associated standard error. To obtain the difference between the proportions of women and men living in poverty, the following code is used:

```
svycontrast(
  stat = sex_poverty_contrast,
  contrasts = list(sex_difference = c(1, -1))
) %>%
  data.frame()
```

```
##                 contrast sex_difference
## sex_difference -0.00532         0.0224
```

From this, it is concluded that the difference between the proportions of women and men living in poverty is -0.005 (-0.5%), with a standard error of 0.022, indicating that the estimated proportion of women living in poverty is slightly lower than that observed among men.

Another exercise that can be developed from a household survey consists of estimating the proportion of unemployed people by region of residence. To do this, unemployment proportions are estimated for each of the regions considered in the analysis. The results obtained are presented in Table 4.14:

```
region_unemployment_proportion <- survey_design %>%
  filter(!is.na(unemployed)) %>%
  group_by(Region) %>%
  summarise(proportion = survey_mean(unemployed,
                                     na.rm = TRUE,
                                     vartype = c("se", "ci")))

region_unemployment_contrast <- svyby(
  formula = ~unemployed,
  by = ~Region,
  design = survey_design %>%
    filter(!is.na(unemployed)),
  FUN = svymean,
  na.rm = TRUE,
  covmat = TRUE,
  vartype = c("se", "ci")
)
```

Table 4.14: Proportion unemployed by region

Region	proportion	proportion_se	proportion_low	proportion_upp
Norte	0.049	0.020	0.009	0.088
Sur	0.066	0.024	0.019	0.113
Centro	0.039	0.012	0.014	0.063
Occidente	0.040	0.012	0.016	0.064
Oriente	0.030	0.013	0.005	0.054

Once the unemployment proportions by region have been estimated, the next step is to evaluate whether there are statistically significant differences between some of these proportions using linear contrasts. In particular, the interest is in comparing the differences between the estimated unemployment proportions of different regions. The contrasts considered are $\hat{p}_{North} - \hat{p}_{Center} = 0.01004$, $\hat{p}_{South} - \hat{p}_{Center} = 0.02691$, and $\hat{p}_{West} - \hat{p}_{East} = 0.01046$. In matrix form, these regional contrasts can be written as follows:

$$\begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

To calculate the standard errors associated with each of the previous contrasts, it is necessary to consider not only the variances of the regional estimates, but also the covariances between them. This information is summarized in the variance-covariance matrix of the estimated unemployment proportions by region, presented below:

```
vcov(region_unemployment_contrast) %>%
  data.frame()
```

Table 4.15: Variance-covariance matrix of the unemployment proportion by region

	Norte	Sur	Centro	Occidente	Oriente
Norte	0	0.000	0	0	0
Sur	0	0.001	0	0	0
Centro	0	0.000	0	0	0
Occidente	0	0.000	0	0	0
Oriente	0	0.000	0	0	0

Note that the covariances between the regional estimates are equal to zero because the samples selected in each region are independent of one another. Therefore, the standard error of each contrast is obtained by applying the variance expression for the difference between two estimators. In this case, the calculation reduces to the square root of the sum of the corresponding variances:

```
sqrt(0.0002981 + 0.0002884 - 2 * 0)
```

```
## [1] 0.0242
```

```
sqrt(0.0001968 + 0.0002884 - 2 * 0)
```

```
## [1] 0.022
```

```
sqrt(0.0001267 + 0.0004093 - 2 * 0)
```

```
## [1] 0.0232
```

Alternatively, the `survey` package allows these contrasts to be calculated directly using the `svycontrast` function, which obtains both the estimated differences between proportions and their respective standard errors. In this case, the contrasts defined above are estimated as follows:

```
svycontrast(
  stat = region_unemployment_contrast,
  contrasts = list(
    north_south = c(1, 0, -1, 0, 0),
    south_center = c(0, 1, -1, 0, 0),
    west_east = c(0, 0, 0, 1, -1)
  )
) %>%
  data.frame()
```

```
##              contrast     SE
## north_south    0.0100 0.0236
## south_center   0.0269 0.0268
## west_east      0.0105 0.0176
```

IV Cross-tabulations

In addition to analyzing each variable individually, it is especially relevant to study whether there is an association between two categorical variables. This type of analysis makes it possible to identify patterns and relationships that provide valuable information for decision-making and for understanding social phenomena. For example, in the field of public policy, education and employment can be related to design labor-market strategies; in the evaluation of social programs, differences in access to health services can be analyzed by income level; and in social research, demographic variables and living conditions can be studied together to understand population dynamics and trends.

Formally, let x and y be two categorical variables with R row categories and C column categories, respectively. In the context of sample surveys, the interest usually focuses on estimating the joint distribution of both variables, that is, the proportion of population units that simultaneously belong to each possible combination of categories. To do this, the entries of a cross-tabulation, also known as a contingency table, are estimated using weighted frequencies obtained from the expansion factors and the sampling design. The estimate of the population total associated with each cell (r, c) is defined as:

$$\hat{N}_{rc} = \sum_h \sum_i \sum_k w_{hik}\ I(x_{hik} = r,\ y_{hik} = c),$$

where w_{hik} represents the expansion factor associated with element k of PSU i belonging to stratum h, while $I(x_{hik} = r,\ y_{hik} = c)$ corresponds to an indicator function that takes the value one when the observed unit simultaneously belongs to category r of variable x and category c of variable y, and takes the value zero in any other case.

In addition to joint frequencies, it is also possible to estimate marginal sizes by rows and columns, defined respectively as $\hat{N}_{r+} = \sum_{c=1}^{C} \hat{N}_{rc}$ and $\hat{N}_{+c} = \sum_{r=1}^{R} \hat{N}_{rc}$. The grand total is expressed as $\hat{N}_{++} = \sum_{r=1}^{R} \sum_{c=1}^{C} \hat{N}_{rc}$. The general structure of a contingency table with R rows and C columns can be represented as follows:

Variable 2	Variable 1		⋯		Row marginal
	1	2	⋯	C	
1	$\hat{N}_{11}$	$\hat{N}_{12}$	⋯	$\hat{N}_{1C}$	$\hat{N}_{1+}$
2	$\hat{N}_{21}$	$\hat{N}_{22}$	⋯	$\hat{N}_{2C}$	$\hat{N}_{2+}$
⋮	⋮	⋮	⋱	⋮	⋮
R	$\hat{N}_{R1}$	$\hat{N}_{R2}$	⋯	$\hat{N}_{RC}$	$\hat{N}_{R+}$
Column marginal	$\hat{N}_{+1}$	$\hat{N}_{+2}$	⋯	$\hat{N}_{+C}$	$\hat{N}_{++}$

Traditionally, contingency tables are usually represented as two-dimensional arrays of dimension $R \times C$. However, they can also be extended to incorporate one or more additional variables, generating subsets or subtables that make it possible to study more complex relationships between categorical variables. Based on the estimated sizes, proportions can also be estimated for each cell of the contingency table as follows:

$$\hat{p}_{rc} = \frac{\hat{N}_{rc}}{\hat{N}_{++}}$$

Next, continuing with the example database, the percentage distribution of men and women by poverty status is estimated, also incorporating their respective standard errors and confidence intervals. The results are presented in Table 4.17. This type of analysis makes it possible to describe the composition of the poor population by sex and evaluate the precision of the estimates obtained from the sampling design.

```
poverty_design <- survey_design %>%
  mutate(poor = factor(
    poor,
    levels = 0:1,
    labels = c("Not poor", "Poor")
  ))
```

```
poverty_design %>%
  group_by(poor, Sex) %>%
  summarise(proportion = survey_prop(vartype = c("se", "ci")))
```

Table 4.17: Proportion of men and women living in poverty and not living in poverty

poor	Sex	proportion	proportion_se	proportion_low	proportion_upp
Not poor	Female	0.529	0.012	0.505	0.554
Not poor	Male	0.471	0.012	0.446	0.495
Poor	Female	0.524	0.016	0.492	0.555
Poor	Male	0.476	0.016	0.445	0.508

The results indicate that 52.3% of the population living in poverty is female, while 47.6% is male. For women, the 95% confidence interval is between 49.2% and 55.5%, while the corresponding interval for men is between 44.5% and 50.7%. These estimates make it possible to observe the relative distribution of poverty by sex, while also considering the uncertainty inherent in the sampling process.

Using the same `srvyr` approach, the contingency table can also be estimated with `group_by()` and `summarise()`. The following example estimates the distribution of poverty status by sex, also obtaining measures of precision such as standard errors and confidence intervals. The corresponding code and results are presented in Table 4.18.

```
sex_by_poverty_proportion <- poverty_design %>%
  group_by(poor, Sex) %>%
  summarise(proportion = survey_prop(vartype = c("se", "ci")))
```

Table 4.18: Proportion of men and women by poverty status

poor	Sex	proportion	proportion_se	proportion_low	proportion_upp
Not poor	Female	0.529	0.012	0.505	0.554
Not poor	Male	0.471	0.012	0.446	0.495
Poor	Female	0.524	0.016	0.492	0.555
Poor	Male	0.476	0.016	0.445	0.508

In addition, confidence intervals can be obtained using the `confint()` function, which automatically calculates the lower and upper limits associated with the estimates produced. The procedure for estimating the confidence intervals is presented in Table 4.19. Note that the intervals coincide with those generated previously using the `group_by` function.

```
sex_by_poverty_proportion %>%
  dplyr::select(poor, Sex, proportion_low, proportion_upp)
```

Table 4.19: Confidence intervals for the proportion by sex and poverty

poor	Sex	proportion_low	proportion_upp
Not poor	Female	0.505	0.554
Not poor	Male	0.446	0.495
Poor	Female	0.492	0.555
Poor	Male	0.445	0.508

Another analysis of interest related to two-way tables in household surveys consists of estimating the percentage of unemployed people by sex. The corresponding procedure and results are presented in Table 4.20.

```
sex_by_employment_proportion <- survey_design %>%
  filter(!is.na(Employment)) %>%
  group_by(Employment, Sex) %>%
  summarise(proportion = survey_prop(vartype = c("se", "ci")))
```

Table 4.20: Proportion of men and women by employment status

Employment	Sex	proportion	proportion_se	proportion_low	proportion_upp
Unemployed	Female	0.273	0.054	0.180	0.390
Unemployed	Male	0.727	0.054	0.610	0.820
Inactive	Female	0.770	0.023	0.721	0.813
Inactive	Male	0.230	0.023	0.187	0.279
Employed	Female	0.405	0.019	0.369	0.442
Employed	Male	0.595	0.019	0.558	0.631

From the previous output, it can be observed that 27.2% of women and 72.7% of men are unemployed, with standard errors for these estimates of 5.3% for both women and men. The corresponding confidence intervals are calculated below and presented in Table 4.21:

```
sex_by_employment_proportion %>%
  dplyr::select(Employment, Sex, proportion_low, proportion_upp)
```

Table 4.21: Confidence intervals for the proportion by sex and employment

Employment	Sex	proportion_low	proportion_upp
Unemployed	Female	0.180	0.390
Unemployed	Male	0.610	0.820
Inactive	Female	0.721	0.813
Inactive	Male	0.187	0.279
Employed	Female	0.369	0.442
Employed	Male	0.558	0.631

If the objective is now to estimate poverty by region, the numeric variable `poor` is used within an `srvyr` workflow, as shown in Table 4.22.

```
region_poverty_proportion <- survey_design %>%
  group_by(Region, poor) %>%
  summarise(proportion = survey_prop(vartype = c("se", "ci")))
```

Table 4.22: Proportion living in poverty by region

Region	poor	proportion	proportion_se	proportion_low	proportion_upp
Norte	0	0.641	0.055	0.526	0.742
Norte	1	0.359	0.055	0.258	0.474
Sur	0	0.656	0.043	0.566	0.737

Region	poor	proportion	proportion_se	proportion_low	proportion_upp
Sur	1	0.344	0.043	0.263	0.434
Centro	0	0.635	0.079	0.470	0.773
Centro	1	0.365	0.079	0.227	0.530
Occidente	0	0.599	0.047	0.504	0.687
Occidente	1	0.401	0.047	0.313	0.496
Oriente	0	0.548	0.088	0.374	0.711
Oriente	1	0.452	0.088	0.289	0.626

From the above, it can be concluded that in the North region, 35% of people are living in poverty, while in the South the figure is 34%. The highest poverty level is found in the East region, with 45% of people living in poverty. The standard errors of the estimates.

V The *odds* ratio

The *odds* ratio is a widely used measure for studying the association between two categorical variables, especially when the interest is in comparing the probability of an event occurring across different population groups. Its interpretation is based on comparing the relative odds of an event occurring between two analysis categories.

In this sense, this parameter makes it possible to evaluate how those odds change across different groups of interest and is a fundamental tool in social, economic, and health studies. In addition, this measure can also be used to quantify the relationship between the levels of a variable and a categorical factor by comparing the relative odds of the event occurring in each group [Heeringa et al., 2017].

For example, suppose the aim is to study the association between people's sex and poverty status. In particular, the interest is in evaluating whether the relative odds of belonging to the non-poor group versus the poor group differ between women and men. To do this, the following *odds* ratio can be defined:

$$\frac{P(\text{Sex} = \text{Female} \mid \text{pobreza} = 0) \big/ P(\text{Sex} = \text{Female} \mid \text{pobreza} = 1)}{P(\text{Sex} = \text{Male} \mid \text{pobreza} = 1) \big/ P(\text{Sex} = \text{Male} \mid \text{pobreza} = 0)}$$

The numerator expression represents the relative odds of observing women not living in poverty relative to observing women living in poverty. Analogously, the denominator represents the relative odds of observing men living in poverty relative to men not living in poverty. Therefore, the *odds* ratio compares both relative odds and makes it possible to quantify whether the association between sex and poverty favors one group more than the other.

A value equal to one would indicate the absence of association between the variables, that is, that the relationship between sex and poverty status is similar for men and women. Values greater than one would indicate greater relative odds for women compared with men, while values below one would suggest greater relative odds for men. In the context of household surveys, this type of measure is especially useful for studying sociodemographic inequalities and gaps in well-being conditions across different population groups.

The procedure for carrying this out in R is first to estimate the proportions of the cross-tabulation between the sex and poverty variables, presented in Table 4.23:

```
sex_poverty_interaction_proportion <- survey_design %>%
  group_by(Sex, poor) %>%
  summarise(proportion = survey_prop(vartype = c("se", "ci")))

sex_poverty_odds_contrast <- svymean(x = ~interaction(Sex, poor),
                                     design = survey_design,
                                     se = TRUE, na.rm = TRUE, ci = TRUE,
                                     keep.vars = TRUE)
```

Table 4.23: Joint proportions of sex and poverty status

Sex	poor	proportion	proportion_se	proportion_low	proportion_upp
Female	0	0.611	0.032	0.547	0.671
Female	1	0.389	0.032	0.329	0.453
Male	0	0.605	0.037	0.531	0.675
Male	1	0.395	0.037	0.325	0.469

Then, the contrast is performed by dividing each of the elements of the expression shown above:

```
odds_ratio <- quote(
 (`interaction(Sex, poor)Female.0` /
  `interaction(Sex, poor)Female.1`) /
 (`interaction(Sex, poor)Male.0` /
  `interaction(Sex, poor)Male.1`)
)

svycontrast(
  stat = sex_poverty_odds_contrast,
  contrasts = odds_ratio
)
```

```
##          nlcon  SE
## contrast  1.02 0.1
```

The result is that the estimated *odds* that a woman is not living in poverty, compared with a man, is equal to 1.02. This means that, without considering other survey variables, women have odds of not being in poverty that are approximately 2% higher than those observed among men. In other words, the relative probability of not living in poverty is slightly higher for women than for men.

VI χ^2 test of independence

As introduced in previous chapters, a hypothesis test is a statistical procedure used to evaluate whether the evidence observed in a sample is compatible with a statement about the population. In the context of contingency tables, tests of independence make it possible to analyze whether there is an association between two categorical variables.

In this case, the null hypothesis (H_0) states that both variables are independent; that is, the distribution of one variable does not depend on the categories of the other. Under this assumption, the differences observed in the sample are attributed only to sampling variability and not to a real relationship between the variables in the population. Mathematically, this hypothesis can be expressed as:

$$H_0 : p^0_{rc} = p_{r+} \times p_{+c}, \quad \text{para todo } r = 1, \dots, R \text{ y } c = 1, \dots, C$$

Where p^0_{rc} represents the expected proportion in cell (r, c) under the assumption of independence, while P_{r+} and P_{+c} correspond to the row and column marginal proportions, respectively. Under this formulation, the independence hypothesis implies that the joint probability of each cell can be expressed as the product of its marginal probabilities.

Consequently, the test of independence consists of comparing the observed or estimated proportions $\hat{p}_{rc}$ with the expected proportions p^0_{rc} under the null hypothesis. When the differences between them are small, the empirical evidence is consistent with the assumption of independence. Conversely, sufficiently large discrepancies suggest the existence of an association between the variables and lead to rejection of H_0.

As noted previously, in household surveys, features of the complex sampling design, such as stratification, cluster selection, and the use of expansion factors, affect the variability of the estimators compared with what would be obtained under simple random sampling. As a result, Pearson's classical χ^2 test of independence is not appropriate for analyzing contingency tables from this type of design, because it may underestimate or overestimate the real variability of the estimates.

To correct this problem, Fay [1979] and Fellegi [1980] proposed the first adjustments to Pearson's chi-square statistic based on the generalized design effect ($GDEFF$). Later, Rao and Scott [1984] and Thomas and Rao [1987] expanded and formalized the theoretical framework for these corrections, giving rise to what is now known as the Rao-Scott test, considered the reference procedure for analyzing categorical data obtained from complex surveys.

The central idea of this approach is to adapt the classical test of independence by incorporating the generalized design effect, thereby obtaining a statistic that is robust to the complexities of the sampling design:

$$\chi^2_{RS} = \frac{n_{++}}{GDEFF} \sum_r \sum_c \frac{(\hat{p}_{rc} - p^0_{rc})^2}{p^0_{rc}}$$

where n_{++} represents the total sample size, $\hat{p}_{rc}$ corresponds to the estimated proportion in cell (r, c) of the contingency table, and p^0_{rc} denotes the expected proportion under the null hypothesis of independence, calculated from the product of the row and column marginal proportions.

The term $GDEFF$ [Heeringa et al., 2017] represents the generalized design effect and measures how much the variability of the estimates increases or decreases as a consequence of the complex survey design relative to the variability that would be observed under simple random sampling. In this way, the Rao–Scott adjustment allows the inferences derived from the test of independence to adequately reflect the structure of the sampling design.

Under the null hypothesis H_0, the statistic χ^2_{RS} approximately follows a χ^2 distribution with $(R-1)(C-1)$ degrees of freedom, where R and C correspond to the number of row and column categories, respectively. In situations with small samples or few degrees of freedom, adjustments based on the F distribution are often used because they improve the precision of statistical inference. Rao-Scott tests are currently the standard for analyzing association between categorical variables in complex surveys [Heeringa et al., 2017].

The `survey` package in `R` implements this type of test through the `svychisq()` function, which automatically incorporates the Rao-Scott adjustments to account for the characteristics of the complex sampling design. This function makes it possible to evaluate the existence of association between two categorical variables using weighted contingency tables and properly correcting the variability of the estimators. As an example, to evaluate whether poverty status is independent of sex, the following code is run:

```
svychisq(formula = ~Sex + poor, design = survey_design, statistic = "F")
```

```
##
##  Pearson's X^2: Rao & Scott adjustment
##
## data:  NextMethod()
## F = 0.06, ndf = 1, ddf = 119, p-value = 0.8
```

In this case, the `formula` argument specifies the two categorical variables whose independence is to be evaluated; `design` corresponds to the sampling design defined previously; and `statistic = "F"` indicates that the test will be carried out using the adjusted version based on the F distribution. If the p-value is greater than 5%, the null hypothesis H_0 is not rejected, leading to the conclusion that there is not enough statistical evidence to state that poverty and sex are associated. Conversely, if the p-value is less than 5%, H_0 is rejected, suggesting the existence of dependence or association between the two categorical variables. Other relationships can be evaluated in the same way, such as unemployment and sex or poverty and region.

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

```
##
##  Pearson's X^2: Rao & Scott adjustment
##
## data:  NextMethod()
## F = 62, ndf = 2, ddf = 201, p-value <0.0000000000000002
```

In this first case, for the test of independence between the variables `Sex` and `Employment`, the statistic obtained was $F = 62.251$, with a p-value less than 2.2×10^{-16}, providing statistically

significant evidence to reject the null hypothesis of independence between the two variables. Consequently, the results suggest that there is a significant association between sex and employment status in the study population.

```
svychisq(formula = ~Region + poor, design = survey_design, statistic = "F")
```

```
##
##  Pearson's X^2: Rao & Scott adjustment
##
## data:  NextMethod()
## F = 0.5, ndf = 3, ddf = 358, p-value = 0.7
```

On the other hand, this test of independence between the variables `Region` and `poor` produces $F = 0.48794$, with a p-value of 0.6914. Because this p-value is considerably greater than conventional significance levels, there is not enough evidence to reject the null hypothesis of independence. Therefore, the results suggest that, under the sampling design considered, no statistically significant association is observed between region and poverty status.

VII Visualization of categorical variables

The visualization of categorical variables is fundamental in household survey analysis because it makes it possible to clearly communicate the distribution of population groups and the comparisons between them. As in the analysis of continuous variables, graphs must incorporate sampling weights so that the visual representation corresponds to valid population-level estimates. When statistical precision is relevant, it is advisable to accompany bars with confidence intervals, which communicate both the central value and the uncertainty associated with the estimation process.

This section uses the `ggplot2` package [Wickham et al., 2026b].

```
library(ggplot2)
```

A Bar charts

Constructing bar charts in `R` requires, as a prior step, obtaining the estimates to be represented visually. In the context of complex surveys, these estimates can be calculated using the `survey_total()` or `survey_prop()` functions from the `srvyr` package, depending on whether the interest is in population totals or proportions.

This procedure is illustrated below using the estimated population size by zone of residence. The corresponding graphical results are presented in Figure 4.1:

```
zone_size <- survey_design %>%
  group_by(Zone) %>%
  summarise(size = survey_total(vartype = c("se", "ci")))
```

```
zone_colors <- c(Urban = "#48C9B0", Rural = "#117864")

ggplot(
  data = zone_size,
  aes(
    x = Zone, y = size,
    ymax = size_upp, ymin = size_low,
    fill = Zone
  )
) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_errorbar(position = position_dodge(width = 0.9), width = 0.3) +
  scale_fill_manual(values = zone_colors) +
  theme_bw() +
  theme(legend.position = "top")
```

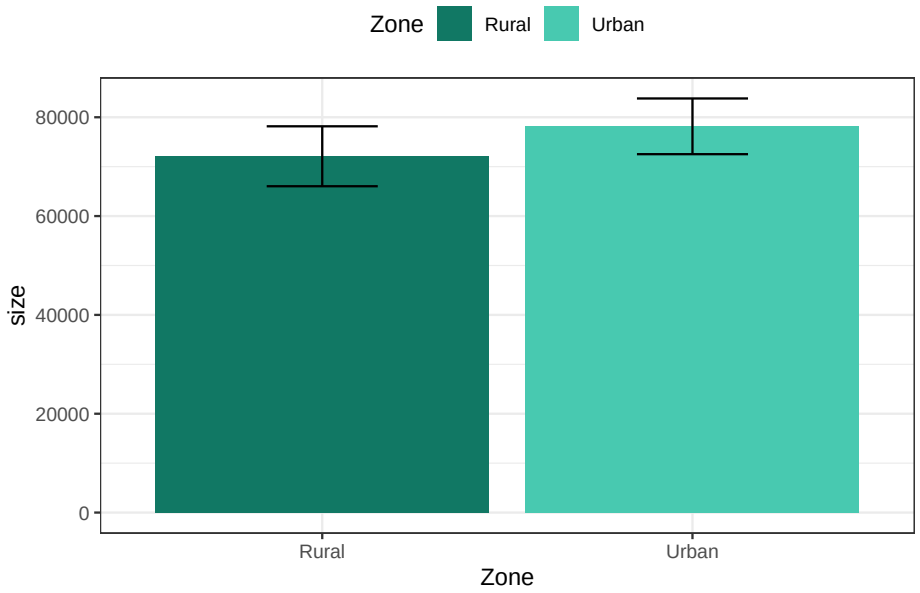

Figure 4.1: Bar chart of population size by geographic zone

The previous procedure can also be applied to variables with more than two categories. In these cases, the bar chart makes it easy to compare the estimates associated with each category of the variable of interest. For example, the estimated distribution of the population by poverty status is presented below, with the graphical results shown in Figure 4.2.

```
poverty_size <- survey_design %>%
  group_by(Poverty) %>%
  summarise(size = survey_total(vartype = c("se", "ci")))

ggplot(
```

```
  data = poverty_size,
  aes(
    x = Poverty, y = size,
    ymax = size_upp, ymin = size_low,
    fill = Poverty
  )
) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_errorbar(position = position_dodge(width = 0.9), width = 0.3) +
  theme_bw() +
  theme(legend.position = "top")
```

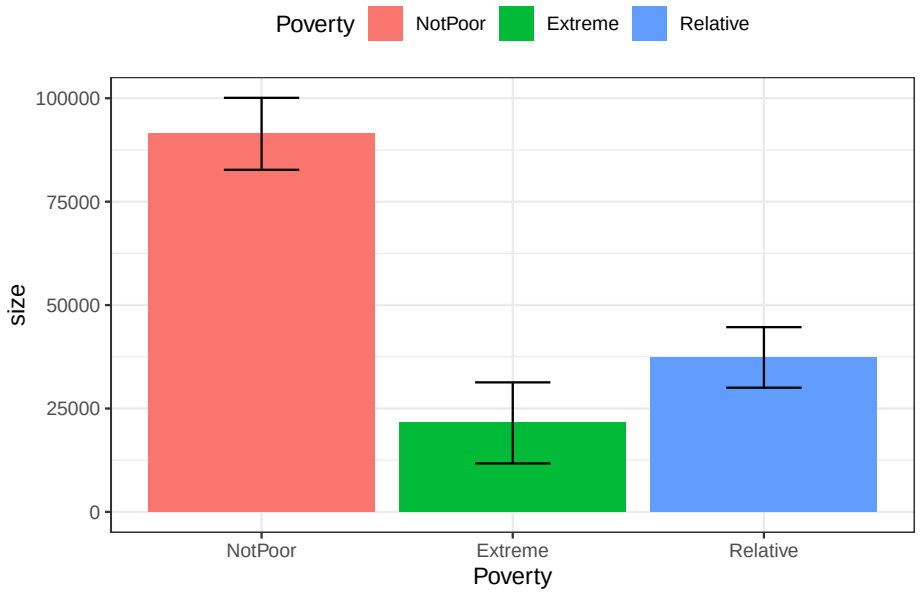

Figure 4.2: Bar chart of population size by poverty status

One advantage of bar charts is that they can easily be extended to comparisons between two categorical variables. For example, population size by poverty level crossed with employment status can be analyzed, as presented in Figure 4.3:

```
employment_poverty_size <- survey_design %>%
  group_by(unemployed, Poverty) %>%
  summarise(size = survey_total(vartype = c("se", "ci"))) %>%
  as.data.frame() %>%
  mutate(
    unemployed = case_when(
      unemployed == 0 ~ "Occupied",
      unemployed == 1 ~ "Unemployed",
      is.na(unemployed) ~ "Not in the Labour Force"
    )
  )
```

```
ggplot(
  data = employment_poverty_size,
  aes(
    x = Poverty, y = size,
    ymax = size_upp, ymin = size_low,
    fill = as.factor(unemployed)
  )
) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_errorbar(position = position_dodge(width = 0.9), width = 0.3) +
  theme_bw() +
  theme(legend.position = "top")
```

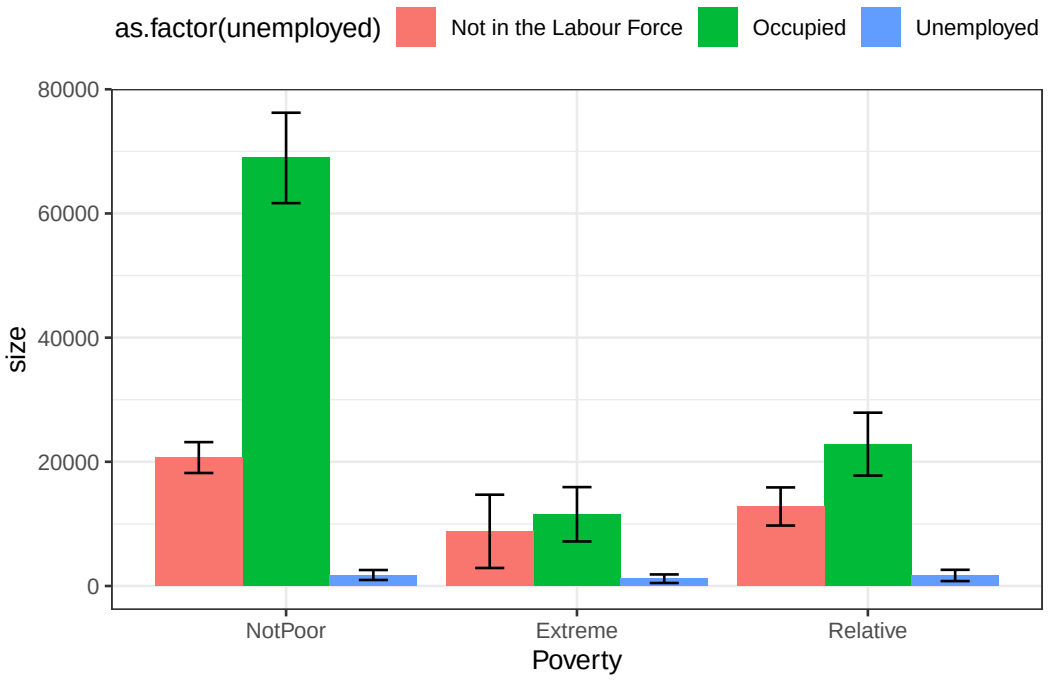

Figure 4.3: Bar chart of population size by employment status and poverty

This type of graphical analysis is especially useful for identifying intersections of vulnerability by showing two or more population characteristics simultaneously. It is also possible to present proportions instead of counts. For example, the proportion of men living in poverty by zone is presented in Figure 4.4:

```
male_zone_poverty_proportion <- male_subset %>%
  group_by(Zone, Poverty) %>%
  summarise(proportion = survey_prop(vartype = c("se", "ci")))

ggplot(
  data = male_zone_poverty_proportion,
```

```
  aes(
    x = Poverty, y = proportion,
    ymax = proportion_upp, ymin = proportion_low,
    fill = Zone
  )
) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_errorbar(position = position_dodge(width = 0.9), width = 0.3) +
  scale_fill_manual(values = zone_colors) +
  theme_bw() +
  theme(legend.position = "top")
```

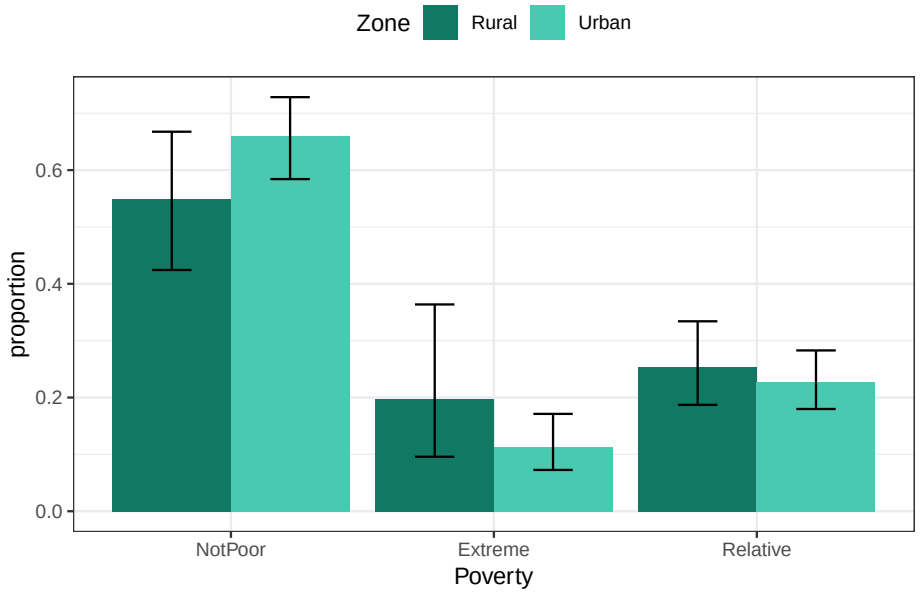

Figure 4.4: Proportion of men living in poverty by geographic zone

Similarly, the proportion of men living in poverty disaggregated by region can be graphed, as presented in Figure 4.5:

```
male_region_poverty_proportion <- male_subset %>%
  group_by(Region, poor) %>%
  summarise(proportion = survey_prop(vartype = c("se", "ci"))) %>%
  data.frame()

ggplot(
  data = male_region_poverty_proportion,
  aes(
    x = Region, y = proportion,
    ymax = proportion_upp, ymin = proportion_low,
    fill = as.factor(poor)
  )
```

```
) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_errorbar(position = position_dodge(width = 0.9), width = 0.3) +
  theme_bw() +
  theme(legend.position = "top")
```

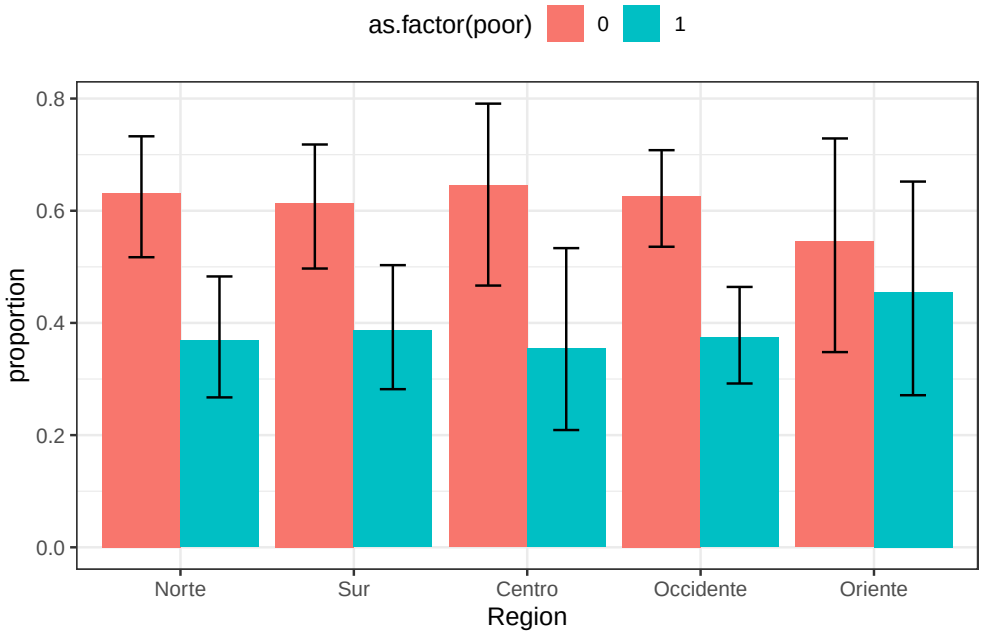

Figure 4.5: Proportion of men living in poverty by region

In summary, bar charts together with their corresponding standard errors provide a clear description of poverty, employment, or other categorical-variable patterns in the population and, when applied correctly, are a powerful tool for communicating findings from household surveys while maintaining statistical rigor in the representation of results.

B Maps

Thematic maps are an especially useful visualization tool in household survey analysis because they make it possible to represent the territorial distribution of social and demographic indicators. In this context, the proportion estimates calculated in previous sections, such as the percentage of poverty or unemployment by region, can be projected directly onto geographic units, thereby facilitating the identification of spatial patterns and regional inequalities.

To build this type of map in `R`, geospatial information in *shapefile* format is required, containing the polygons associated with each geographic unit of analysis. This chapter uses the file `bigcity_shape/BigCity.shp`, whose regions correspond to those defined in the `BigCity` database. The `sf` [Pebesma et al., 2026] and `tmap` [Tennekes, 2026] packages make it possible to load, manipulate, and visualize this geographic information in an integrated way with the results obtained from the survey.

The following code loads the packages needed to work with geospatial information and cartographic visualization in R. First, the `sf` library makes it possible to read and manipulate modern spatial objects, while `tmap` provides tools for creating thematic maps. The instruction `tmap_mode("plot")` sets the maps to static visualization mode, which is appropriate for documents and reports. Finally, the `read_sf()` function imports the geographic file `BigCity.shp`, storing it in the `shapeBigCity` object, which contains the polygons corresponding to the analysis regions.

```
library(sf)
library(tmap)
tmap_mode("plot")
bigcity_shape <- read_sf("Data/shapeBigCity/BigCity.shp")
```

With the shapefile loaded, a map of the regions can be generated using `tm_shape()` and `tm_polygons()`, as shown in Figure 4.6:

```
tm_shape(bigcity_shape) +
  tm_polygons(col = "Region")
```

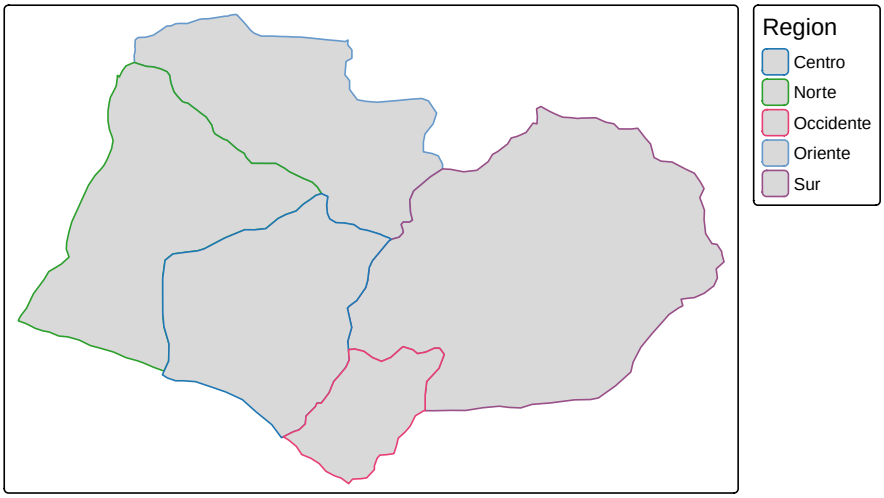

Figure 4.6: Map of BigCity geographic regions

For example, if the aim is to geographically represent the proportion of men living in poverty by region, estimated previously in Figure 4.5, the estimates obtained must be integrated with the spatial information from the shapefile. To do this, the `male_region_poverty_proportion` object, which contains the estimated proportions for each region, is linked to the geographic file using the `left_join()` function. Then, only the observations corresponding to the poverty category (`poor == 1`) are filtered, so that the map exclusively represents the spatial distribution of the male population living in poverty.

```
shape_region_map <- tm_shape(
  bigcity_shape %>%
    left_join(
      male_region_poverty_proportion %>%
        filter(poor == 1),
      by = "Region"
    )
)
```

Once the spatial object with the estimates of interest has been built, the `breaks` argument helps define the cut points that determine the intervals of the color scale used in the map. The thematic map is then generated, making it possible to visualize the regional distribution of the proportion of men living in poverty, as presented in Figure 4.7.

```
poverty_breaks <- c(0, 0.2, 0.4, 0.6, 0.8, 1)
shape_region_map +
  tm_polygons(
    col = "proportion",
    breaks = poverty_breaks,
    title = "Poverty proportion",
    palette = "YlOrRd"
  )
```

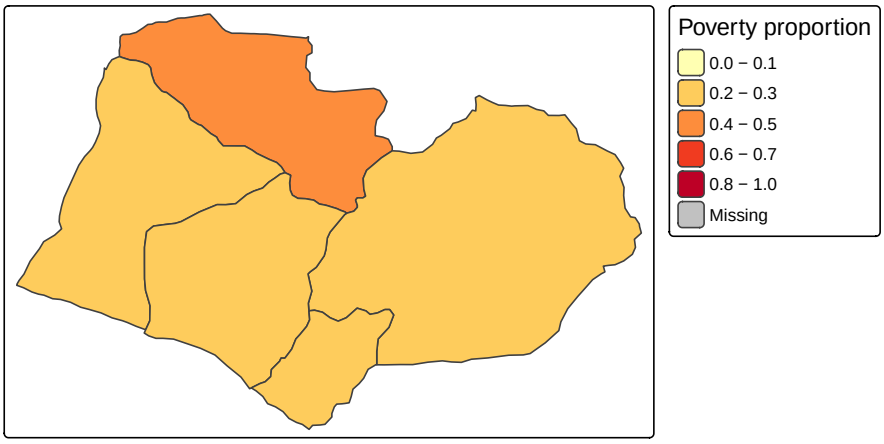

Figure 4.7: Proportion of men living in poverty by region

Another use of maps in surveys is the visualization of estimate precision. For example, using the coefficient of variation of mean income by region makes it possible to identify areas where the estimates are less reliable, as shown in Figure 4.8.

The following code calculates and geographically represents the coefficient of variation of mean income by region. First, the function estimates average income for each region. The `vartype = "cv"` argument requests calculation of the coefficient of variation for each estimate. The classification intervals of the color scale are then defined using the `cv_breaks` object, and the estimates are integrated with the geographic information from the shapefile using `left_join()`. Finally, using the `tm_shape()` and `tm_polygons()` functions from the `tmap` package, the map is constructed to spatially represent the coefficients of variation using a black-and-white color scale and adjusting the map aspect ratio with `tm_layout(asp = 0)`.

```
region_mean <- survey_design %>%
  group_by(Region) %>%
  summarise(mean = survey_mean(Income,
                               na.rm = TRUE,
                               vartype = c("cv")))

cv_breaks <- c(0, 0.1, 0.2, 1)
shape_cv_map <- tm_shape(
  bigcity_shape %>%
    left_join(region_mean, by = "Region")
)

shape_cv_map +
  tm_polygons(
    "mean_cv",
    breaks = cv_breaks,
    title = "Mean income CV",
    palette = c("#FFFFFF", "#000000")
  ) +
  tm_layout(asp = 0)
```

Polygons with darker shades correspond to regions where the income estimate has greater relative variability, which may indicate insufficient sample sizes to obtain precise estimates in those areas.

When two variables are to be represented simultaneously, for example, the poverty proportion and its coefficient of variation, it is possible to build a bivariate map using `ggplot2` together with the `biscale` [Prener et al., 2025] and `cowplot` [Wilke, 2025] packages. This type of visualization makes it possible to jointly identify regions with high poverty and high estimation uncertainty, as presented in Figure 4.9. The following code estimates the proportion of women living in poverty in rural zones for each region, also incorporating calculation of the coefficient of variation of the estimates, thereby generating a database ready for constructing a bivariate map.

```
region_sex_poverty_proportion <- survey_design %>%
  group_by(Region, Zone, Sex, poor) %>%
  summarise(proportion = survey_mean(vartype = "cv")) %>%
  filter(poor == 1, Zone == "Rural", Sex == "Female")
```

The following code constructs a bivariate map that makes it possible to simultaneously visualize the rural female poverty proportion and the coefficient of variation associated with that estimate

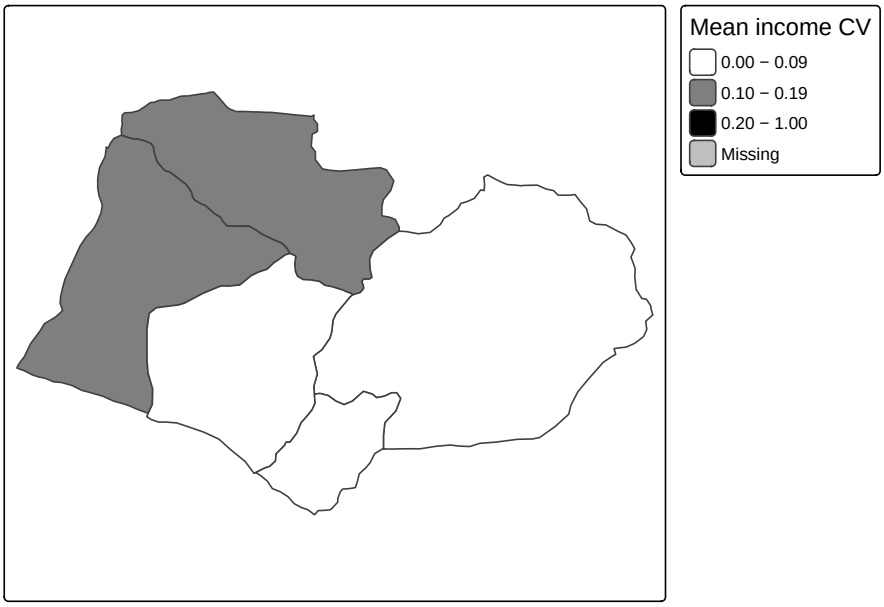

Figure 4.8: Coefficient of variation of mean income by region

by region. First, the estimates contained in `region_sex_poverty_proportion` are integrated with the geographic information from the shapefile using `left_join()`. Then, using the `bi_class()` function from the `biscale` package, the variables `proportion` and `proportion_cv` are jointly classified into a 3×3 grid of categories (`dim = 3`), using Fisher's classification method (`style = "fisher"`). Next, with `ggplot2` and `geom_sf()`, the thematic map is generated using bivariate colors defined by `bi_scale_fill()`, while `bi_theme()` applies a minimalist style and the default legend is removed.

Next, the `bi_legend()` function creates a specialized legend that makes it possible to interpret both dimensions of the map simultaneously: the coefficient of variation and rural female poverty. Finally, using the `ggdraw()` and `draw_plot()` functions from the `cowplot` package, the map and legend are combined into a single visualization.

```
library(biscale)
library(cowplot)

bivariate_shape <- bigcity_shape %>%
  left_join(region_sex_poverty_proportion, by = "Region")

k <- 3
bivariate_data <- bi_class(
  bivariate_shape,
  y = proportion,
  x = proportion_cv,
  dim = k,
  style = "fisher"
)
```

```
bivariate_map <- ggplot() +
  geom_sf(
    data = bivariate_data,
    aes(fill = bi_class, geometry = geometry),
    colour = "white",
    size = 0.1
  ) +
  bi_scale_fill(pal = "GrPink", dim = k) +
  bi_theme() +
  theme(legend.position = "none")

bivariate_legend <- bi_legend(
  pal = "GrPink",
  dim = k,
  xlab = "Coefficient of variation",
  ylab = "Rural female poverty",
  size = 8
)

ggdraw() +
  draw_plot(bivariate_map, 0, 0, 1, scale = 0.7) +
  draw_plot(bivariate_legend, 0.75, 0.4, 0.2, 0.2)
```

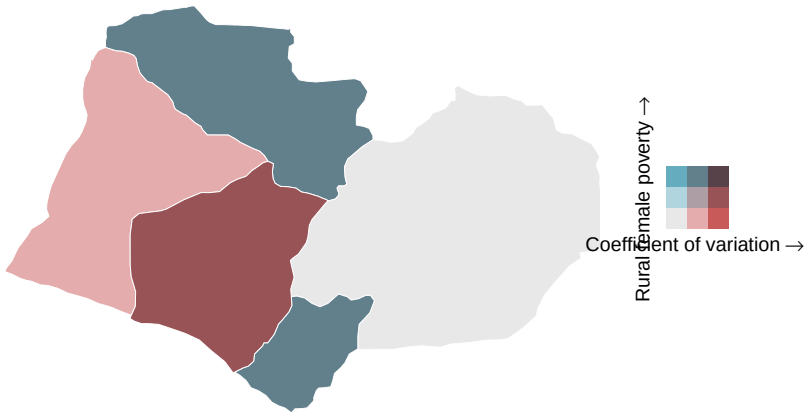

Figure 4.9: Bivariate map: rural female poverty proportion and its coefficient of variation by region

In this map, each cell of the legend combines a shade of the poverty proportion (vertical axis) with a shade of the coefficient of variation (horizontal axis), making it possible to simultaneously identify

regions with high poverty and high estimation uncertainty. This type of representation is especially valuable for guiding decisions about the need to increase sample size in specific geographic areas.

Chapter 5

Regression models

Regression models are one of the most widely used tools for analyzing survey data, since they make it possible to study the relationship between a variable of interest and a set of explanatory variables. Through these models, it is possible to assess how certain population characteristics vary according to demographic, social, or economic factors observed in the sample. However, the validity of the results obtained depends on appropriate model specification and on correctly accounting for the characteristics of the survey sampling design.

In general terms, regression models seek to describe and quantify the association between a response (dependent) variable and one or more explanatory (independent) variables, providing elements for statistical interpretation and for drawing inferences about the study population. Nevertheless, the validity of the results obtained depends, to a large extent, on appropriate model specification and on correctly accounting for the characteristics of the survey sampling design [Heeringa et al., 2017].

For example, household income can be analyzed as a response variable as a function of characteristics such as educational attainment and the employment status of household members, considered as explanatory variables. Using information from household surveys, this type of model makes it possible to identify patterns of association, quantify the effect of different socioeconomic factors on income, and generate empirical evidence useful for the design, monitoring, and evaluation of public policies.

However, because household surveys are based on complex sampling designs, classical regression methods, developed under simple random sampling assumptions, may be inappropriate. Ignoring the characteristics of the sampling design can generate biased estimates of regression coefficients, as well as underestimation of their standard errors and variances, compromising the validity of statistical inferences.

Consequently, the analysis of survey data requires close attention to the sampling design. Incorporating survey weights and the adjustments corresponding to stratification and clustering makes it possible to obtain valid and precise inferences. In addition, simplified alternatives have been proposed in some cases, such as the use of normalized weights or approximate weighting approaches, which seek to balance methodological complexity with the practical feasibility of the analysis.

The study of regression under complex sampling designs has a well-documented history. Kish and Frankel [1974] were among the first to discuss the impact of these designs on inferences derived from regression models. Subsequently, Fuller [1975] developed a variance estimator based

on linearization techniques for multiple linear regression models with unequal weighting, and introduced specific methods for stratified and two-stage designs.

Later, Shah et al. [1977] addressed the problem of violations of classical assumptions when working with survey data, proposing robust inference alternatives for the parameters. In parallel, Binder [1983] focused on the sampling distributions of regression estimators in finite populations, establishing procedures for estimating variances under complex schemes.

In the following years, Skinner et al. [1989] expanded these contributions through variance estimators for regression coefficients that incorporated stratification and clustering, explicitly recommending the use of linearization methods or alternative techniques for variance estimation. Later, Fuller [2002] provided a compendium of estimation methods applicable to regression models in complex surveys, while Pfeffermann [2011] discussed more recent developments, such as *q-weighted* weighting methods, presenting empirical evidence of their usefulness.

I Model formulation

Regression models under complex sampling designs make it possible to move beyond descriptive statistics and approach causal or predictive explanations, provided that the particularities of the sampling design are recognized and adjusted for. Their correct application opens the possibility of examining how sociodemographic and economic characteristics are associated with different outcomes of interest, contributing key evidence for public policy formulation.

As a starting point, it is useful to present the general structure of regression models. The most basic formulation corresponds to the simple linear regression model, which describes the relationship between a response variable and a single explanatory variable through the following expression:

$$y = \beta_0 + \beta_1 x + \varepsilon$$

Here, y represents the response (dependent) variable, x the explanatory (independent) variable, β_0 the model intercept, β_1 the coefficient associated with the independent variable, and ε the error term, which captures the variability not explained by the model and can be interpreted as the difference between the observed value and the value estimated by the model, denoted by $\hat{y}_k$.

In many empirical applications, and especially in the analysis of household surveys, the phenomenon of interest depends simultaneously on multiple factors. In these cases, multiple linear regression models are used, incorporating several explanatory variables:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon,$$

where each coefficient β_j quantifies the association between the response variable and the corresponding covariate x_j, holding the other variables included in the model constant. More compactly, the model can be expressed using matrix notation as:

$$y_k = \mathbf{x}_k \beta + \varepsilon_k, \quad k = 1, \ldots, n,$$

where $\mathbf{x}_k = [1, x_{1i}, \dots, x_{pi}]$ represents the vector of covariates associated with unit k, while $\beta = [\beta_0, \beta_1, \dots, \beta_p]'$ corresponds to the vector of unknown model parameters. In this context, the expected value of the response variable conditional on the set of covariates can be expressed as:

$$E(y \mid \mathbf{x}) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

Linear regression models are based on a series of theoretical assumptions that guarantee the validity of the estimates and the inferences derived from the model. First, it is assumed that the expected value of the residuals conditional on the covariates is equal to zero, that is, $E(\varepsilon_k \mid \mathbf{x}_k) = 0$, which implies the absence of systematic bias in the estimation of the response variable. Likewise, homogeneity of the error variance is assumed, so that residual variability remains constant for all values of the covariates, that is, $Var(\varepsilon_k \mid \mathbf{x}_k) = \sigma^2$. In addition, the errors are considered to follow a normal distribution with zero mean and constant variance, expressed as $\varepsilon_k \mid \mathbf{x}_k \sim N(0, \sigma^2)$, and the residuals associated with different observations are assumed to be independent of one another, such that $Cov(\varepsilon_k, \varepsilon_j \mid \mathbf{x}_k, \mathbf{x}_j) = 0$.

The validity of linear regression models depends on the fulfillment of several classical assumptions widely discussed in the specialized literature. Taken together, these assumptions allow the estimators obtained through the model to have desirable statistical properties, such as unbiasedness, efficiency, and consistency.

II Use of sampling weights

When working with data from surveys based on complex sampling designs, the classical assumptions of regression models are rarely strictly satisfied, since observations do not come from simple and independent random samples, but from selection schemes that incorporate elements such as stratification, clustering, and unequal probabilities of selection. For example, the presence of clustering may violate the assumption of independence of errors, since individuals belonging to the same household, segment, or geographic area tend to share similar characteristics. Similarly, unequal probabilities of selection and weighting adjustments can generate heterogeneity in the variance of observations, violating the assumption of homoscedasticity.

In this context, a fundamental question arises: how should sampling weights be incorporated into regression models? The answer is not straightforward, since survey weights reflect not only probabilities of selection, but also various adjustments associated with nonresponse, calibration, and coverage corrections. Although their use makes it possible to produce estimates that are representative of the population, the direct incorporation of weights into models can also reduce statistical efficiency and increase estimator variability.

In general terms, the literature distinguishes two main approaches to this problem [Heeringa et al., 2017]. The first corresponds to the design-based approach, whose objective is to obtain valid inferences for the target population while respecting the characteristics of the sample selection process. From this perspective, sampling weights are essential for correcting unequal probabilities of inclusion and producing unbiased estimates of the regression coefficients. However, this approach does not protect against possible model specification errors; that is, even when the estimates are valid from a design perspective, the model may not adequately represent the relationships existing in the population.

The second corresponds to the model-oriented approach, according to which weights are not necessarily required as long as the model is correctly specified and the sampling mechanism is non-informative. Under this assumption, the relationships between the variables observed in the sample coincide with those in the population, so the sampling design does not introduce relevant biases in parameter estimation. From this perspective, incorporating weights could even be counterproductive, by unnecessarily increasing the variance of the estimators and, consequently, the standard errors.

The discussion about whether to use weights in regression models has been extensively developed in the specialized literature, particularly in works such as Skinner et al. [1989] and Pfeffermann [2011]. In practice, a frequent methodological recommendation is to estimate models both with and without weights and then compare the results obtained. If including the weights produces important changes in the estimated coefficients or modifies the substantive conclusions of the analysis, this suggests that the sampling design is informative or that the model has specification problems, making the use of weights advisable [United Nations Statistics Division, 2026]. Conversely, if the coefficients remain relatively stable and the weights only increase the standard errors, it may be considered that the model adequately captures the structure of the data and that the use of weights is not strictly necessary.

In applied terms, this decision usually depends on the analytical purpose of the study. When the objective is to carry out descriptive inference and produce estimates that are representative of the population, the use of weights is indispensable. By contrast, in contexts of analytical inference oriented toward the study of associations, causal relationships, or hypothesis tests, both weighted and unweighted models may be used, especially when the model incorporates variables related to the sampling design, such as strata or clusters. Nevertheless, the use of unweighted models must be carefully justified, since it implies assuming more restrictive conditions about the sample selection mechanism and the correct specification of the statistical model.

In summary, when weighting is chosen, weights correct possible biases from over- or underrepresentation of certain groups and help obtain more accurate variance estimates. Within the design-based approach, this allows the results to approximate unbiased values comparable to those that would be obtained in a complete census, even when the model is not optimally formulated. However, when weights are highly dispersed, they can increase the variance of the estimated parameters and make the estimates unstable, which is why in explanatory or analytical contexts unweighted models may, at times, yield more consistent and efficient results.

In any case, if the model is misspecified, ignoring sampling weights can lead to biased, uninformative estimates, or even estimates lacking inferential validity. For this reason, the appropriate selection of the variables included in the model is a fundamental aspect of the analysis. To address these issues, various adjustment strategies have been proposed to achieve a balance between both objectives. The most commonly used procedures include the following:

1. Senate-type weights: this procedure adjusts the weights so that their sum matches the sample size rather than the population size. The objective is to maintain the relative representativeness of the units while reducing the dispersion of the original weights, which is particularly advantageous in surveys where there is high variability among expansion factors. Thus, the new weights are $w_k^{Senate} = w_k \times \frac{n}{\sum w_k}$.

2. Normalized weights: in this approach, the original weights are rescaled so that their sum is equal to one, which avoids an unnecessary increase in variance in the models. This technique

is especially useful when working with different data subsets (for example, models estimated in subpopulations) or when seeking to minimize variance inflation. Thus, the new weights are $w_k^{Normalized} = \frac{w_k}{\sum w_k}$.

It is important to note that, in both methods, the adjusted weights are obtained through direct multiplicative transformations of the original sampling weights. Therefore, they should not be used to calculate population sizes or totals. Likewise, these procedures do not alter estimates of ratios, such as means or proportions, since in those cases the weights cancel out in the quotient. In practice, the use of these adjustments can be considered a pragmatic solution in contexts where specialized survey software is not available. However, when tools such as the R packages `survey` [Lumley, 2024] and `srvyr` [Freedman Ellis and Schneider, 2024], described in previous chapters, are available, rescaling is no longer necessary, since these packages apply the appropriate treatment to preserve both representativeness and the inferential properties of the model.

III Estimation of model parameters

For fitting regression models with household surveys, advanced inferential methodologies have been developed that integrate sources of uncertainty within a single analytical framework, seeking to reflect both the structure of the design and the assumptions and limitations of the model. Among the most relevant approaches are pseudo-likelihood [Molina and Skinner, 1992] and combined inference [Binder, 2011].

The pseudo-likelihood method extends traditional maximum likelihood techniques to adapt them to the particularities of complex sampling designs. In this approach, the sampling distribution defined by the design plays a central role, while the model distribution moves into the background. Although in contexts of well-specified models pseudo-likelihood-based estimators tend to be unbiased or consistent, their main virtue lies in avoiding the biases that would arise from ignoring the sampling design. In practical terms, this method translates the traditional model into one that respects the way in which the data were obtained, ensuring more robust inferences.

By contrast, combined inference proposes a unified framework in which sampling variability and model uncertainty are integrated simultaneously. By considering both sources, this approach offers a more complete view of total variability and makes it possible to obtain more precise and reliable estimates. Its main contribution lies in avoiding the biases that can arise when the analysis is carried out exclusively from either the design or the model perspective. Thus, combined inference is especially useful in applications where population representativeness and the statistical robustness of the fitted models must be balanced.

When estimating the parameters of a linear regression model from a sample with a complex design, the standard approach changes. Consequently, traditional estimators, such as those obtained by maximum likelihood or least squares, may be biased and produce unreliable standard errors. In response to this situation, Wolter and Wolter [2007] proposes the use of robust nonparametric methods, such as Taylor linearization or bootstrap, to estimate variances and obtain valid inferences.

In the case of a simple linear regression model, the estimation of β_1 under a complex sampling scheme is carried out using a weighted estimator, which can be expressed as follows:

$$\hat{\beta}_1 = \frac{\sum_h \sum_i \sum_k w_{hik}\,(y_{hik} - \hat{\bar{y}})(x_{hik} - \hat{\bar{x}})}{\sum_h \sum_i \sum_k w_{hik}\,(x_{hik} - \hat{\bar{x}})^2} = \frac{\hat{t}_{xy}}{\hat{t}_{xx}}$$

Here, $\hat{\bar{y}} = \hat{t}_y/\hat{N}$ and $\hat{\bar{x}} = \hat{t}_x/\hat{N}$ correspond to the estimated means of the study and auxiliary variables, respectively, while w_{hik} represents the sampling weights associated with each observation unit. This formulation explicitly incorporates these weights, making it possible to account for unequal probabilities of selection and the specific characteristics of the sampling design. Likewise, the estimated variance of $\hat{\beta}_1$ can be approximated as:

$$\widehat{Var}\left(\hat{\beta}_1\right) \approx \frac{\widehat{Var}(\hat{t}_{xy}) + \hat{\beta}_1^2 \widehat{Var}(\hat{t}_{xx}) - 2\hat{\beta}_1 \widehat{Cov}(\hat{t}_{xy}, \hat{t}_{xx})}{(\hat{t}_{xx})^2}$$

If $\hat{\beta}_1$ corresponds to the slope estimator in a weighted simple linear regression model, then the intercept estimator $\hat{\beta}_0$ is given by

$$\hat{\beta}_0 = \hat{\bar{y}} - \hat{\beta}_1 \hat{\bar{x}}$$

Note that the variance of $\hat{\beta}_0$ depends simultaneously on the estimators $\hat{\bar{y}}$, $\hat{\bar{x}}$, and $\hat{\beta}_1$. For this reason, its approximation can be obtained using the properties of the variance of linear combinations of estimators, together with Taylor linearization techniques. From this procedure, the variance-covariance matrix of the model coefficients is derived, from which both the standard errors and the covariances associated with the parameter estimates are obtained.

In the case of multiple regression, if $\hat{\beta}$ represents the estimator of β, then the variance of each coefficient is estimated by considering its interdependence with the other parameters, which is reflected in the construction of a variance-covariance matrix that captures both individual variability and the covariances among all estimators. According to Kish and Frankel [1974], this calculation requires using weighted totals of squares and cross-products for all combinations between the dependent variable y and the set of predictors $\mathbf{x} = (1, x_1, \ldots, x_p)$. In general terms:

$$\widehat{Var}(\hat{\beta}) = \hat{\Sigma} = \begin{bmatrix} \widehat{Var}(\hat{\beta}_0) & \widehat{Cov}(\hat{\beta}_0, \hat{\beta}_1) & \cdots & \widehat{Cov}(\hat{\beta}_0, \hat{\beta}_p) \\ \widehat{Cov}(\hat{\beta}_0, \hat{\beta}_1) & \widehat{Var}(\hat{\beta}_1) & \cdots & \widehat{Cov}(\hat{\beta}_1, \hat{\beta}_p) \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{Cov}(\hat{\beta}_0, \hat{\beta}_p) & \widehat{Cov}(\hat{\beta}_1, \hat{\beta}_p) & \cdots & \widehat{Var}(\hat{\beta}_p) \end{bmatrix}$$

This chapter shows how to implement these models in `R` using the `survey` package [Lumley, 2024], together with `srvyr` [Freedman Ellis and Schneider, 2024] and `tidyverse` [Wickham et al., 2026a] to organize the workflow. It covers specification of the sampling design, estimation of linear and logistic models, and the calculation of standard errors and design-adjusted hypothesis tests, integrating theoretical foundations with reproducible practical examples. To illustrate the concepts developed up to this point, the same database used throughout the book will be used. The process begins by loading the packages, the data, and defining the sampling design:

```
library(survey)
library(srvyr)
library(tidyverse)

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

survey_design <- survey_data %>%
  as_survey_design(
    strata = Stratum,
    ids = PSU,
    weights = wk,
    nest = TRUE
  )
```

The examples developed throughout this chapter will use the same subgroups considered in the previous chapters.

```
urban_subset <- survey_design %>%
  filter(Zone == "Urban")

rural_subset <- survey_design %>%
  filter(Zone == "Rural")

female_subset <- survey_design %>%
  filter(Sex == "Female")

male_subset <- survey_design %>%
  filter(Sex == "Male")
```

In order to fit a regression model between the income and expenditure variables, it is useful first to explore the relationship between them. For this purpose, a scatter plot is constructed with ggplot2, which makes it possible to visually assess the shape and strength of the association. The result is presented in Figure 5.1.

```
unweighted_plot <- ggplot(
  data = survey_data,
  aes(x = Expenditure, y = Income)
) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)

unweighted_plot
```

The sample data preserve an approximately linear relationship between income and expenditure, although greater dispersion is observed among the values corresponding to households with higher

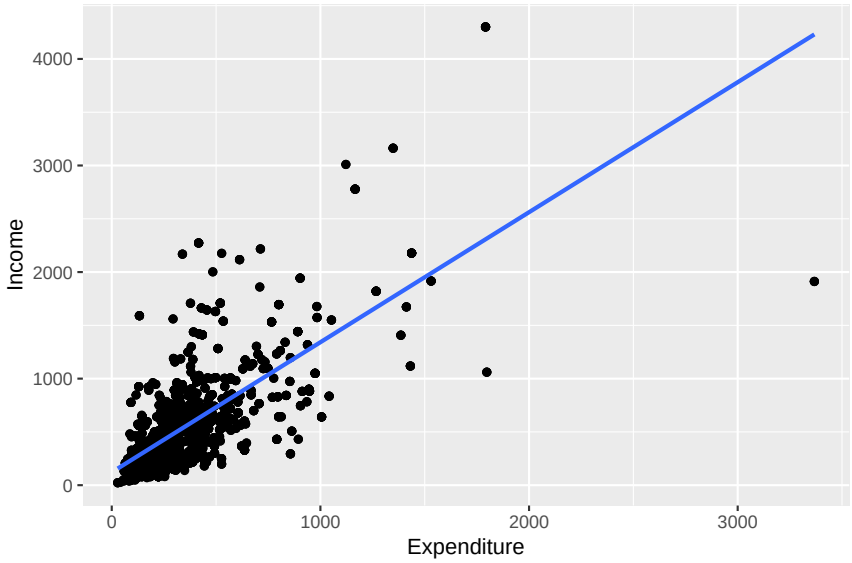

Figure 5.1: Scatter plot between expenditure and income in the sample (unweighted)

expenditure levels. Once the graphical exploration of the data has been completed, the linear regression models are fitted.

To fit models incorporating the expansion factors, the `svyglm` function from the `survey` package is used; this function makes it possible to estimate linear models while explicitly incorporating the characteristics of the sampling design. In this case, the expression `Income ~ Expenditure` specifies that the variable `Income` is modeled as a function of the variable `Expenditure`; that is, a simple linear regression is fitted in which expenditure acts as the explanatory variable for income. The argument `design = survey_design` indicates that the model must use the previously defined sampling design object, while `family = stats::gaussian()` specifies that a linear model with a normal distribution is fitted, equivalent to a classical linear regression.

```
fit_svy <- svyglm(
  Income ~ Expenditure + Zone + Sex,
  design = survey_design,
  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 + Zone + Sex, design = survey_design,
```

```
##     family = stats::gaussian())
##
## Coefficients:
## (Intercept)  Expenditure    ZoneUrban      SexMale
##       73.58         1.22        66.65        20.64
##
## Degrees of Freedom: 2604 Total (i.e. Null);  116 Residual
## Null Deviance:       635000000
## Residual Deviance: 309000000     AIC: 38300
```

The output above corresponds to the fit of a linear regression model in which the variable `Income` is explained by the variables `Expenditure`, `Zone`, and `Sex`. The results indicate the existence of a positive relationship between income and expenditure, such that, holding the other variables constant, expected income increases by approximately 1.22 units for each additional unit of expenditure.

Likewise, systematic differences associated with the sociodemographic characteristics included in the model are evident: residing in an urban area is associated with an average increase of 66.65 units in income relative to the reference category, while being male is associated with an increase of 20.64 units compared with the base sex category. The estimated intercept, equal to 73.58, represents the expected income level for the model reference category (rural area and female sex) when expenditure is equal to zero.

IV Model diagnostics

In the analysis of household surveys, the evaluation of a statistical model is an important aspect that guarantees the validity of the inferences. The literature notes that an appropriately specified model depends not only on the selection of covariates, but also on the fulfillment of the assumptions that support the consistency and reliability of the results [Tellez Piñerez and Lemus Polanía, 2015].

For linear regression models applied to complex surveys, it is necessary to examine different aspects related to goodness of fit and the behavior of the errors. As indicated above, these include the explanatory capacity of the model, the normality and homogeneity of the residual variance, the independence of errors, and the possible presence of influential observations or outliers that may affect the estimates.

These checks are particularly important in the context of complex sampling designs, where characteristics such as stratification, clustering, and the use of expansion factors can intensify problems associated with heteroscedasticity, dependence among observations, or sensitivity to extreme values. For this reason, model diagnostics should not be limited to the classical assumptions of regression, but should also incorporate the particularities derived from the survey design. The systematic application of these procedures makes it possible to assess the robustness of the model, strengthen confidence in the inferences, and ensure that the results obtained are representative and useful for both applied social research and public policy analysis.

A Coefficient of determination

One of the most widely used measures for evaluating the fit of a regression model is the coefficient of determination, denoted by R^2 and also known as the multiple correlation coefficient. This indicator quantifies the proportion of the variability of the dependent variable that is explained by the model. Its values range from zero to one: the closer it is to one, the greater the explanatory capacity of the model; by contrast, values close to zero indicate a low level of explanation of the observed variability. The magnitude of R^2 should not be evaluated in absolute terms, but rather in light of the phenomenon being studied and the type of information available.

The coefficient is calculated from the total and error sums of squares, as $R^2 = 1 - \frac{SSE}{SST}$, where SST represents the total sum of squares and SSE the sum of squared errors. In surveys with complex sampling designs, this measure must be adjusted to incorporate the design structure and the sampling weights. The weighted estimator of the coefficient of determination is defined as:

$$\hat{R}^2 = 1 - \frac{\widehat{SSE}}{\widehat{SST}}$$

where $\widehat{SSE}$ is the weighted sum of squared errors, calculated as $\widehat{SSE} = \sum_h \sum_i \sum_k w_{hik} (y_{hik} - x_{hik}\hat{\beta})^2$; while $\widehat{SST}$ represents the weighted total sum of squares, defined by $\widehat{SST} = \sum_h \sum_i \sum_k w_{hik} (y_{hik} - \hat{\bar{y}})^2$.

Since R^2 tends to increase as more variables are incorporated into the model, it is recommended to complement its interpretation with the adjusted coefficient of determination (R^2_{adj}), which introduces a penalty based on the number of covariates included and the sample size [Pfeffermann, 2011]. This coefficient is defined as $\hat{R}^2_{adj} = 1 - \frac{(n-1)}{(n-p)}(1 - \hat{R}^2)$, where n corresponds to the number of observations, p represents the number of estimated parameters, and $\hat{R}^2$ denotes the weighted coefficient of determination defined above.

In this way, the adjustment makes it possible to evaluate the explanatory capacity of the model while controlling for the effect derived from incorporating new variables. This adjustment facilitates a more equitable comparison between models with different numbers of predictors and is particularly useful in survey analysis, where design complexity and the use of weights can have a notable influence on the magnitude of R^2.

In `R`, the coefficient of determination R^2 can be estimated from the previously fitted models. To do this, a null model is first fitted (including only the intercept), which makes it possible to calculate the weighted total sum of squares ($\widehat{SST}$).

```
null_model <- svyglm(Income ~ 1, design = survey_design)

s1 <- summary(fit_svy)
s0 <- summary(null_model)

wsst <- s0$dispersion[1]
wsse <- s1$dispersion[1]
```

The estimate of the coefficient of determination and its adjusted counterpart is obtained from the following calculation:

```r
n <- nrow(survey_design)
p <- 4
r2 <- 1 - wsse / wsst
r2_adj <- 1 - ((n - 1) / (n - p)) * (1 - r2)

r2
```

```
## [1] 0.5138
```

```r
r2_adj
```

```
## [1] 0.5133
```

B Residuals

In model diagnostics, residual analysis is one of the most relevant tools. If the model is correctly specified, the residuals act as an approximation of the unobserved errors and make it possible to assess indirectly whether the model assumptions are reasonably satisfied in the data. Their systematic review makes it possible to identify possible deviations in the specification, helping to determine whether the fit is adequate or whether, on the contrary, it is necessary to reconsider the functional form of the model or the estimation method used.

In surveys with complex sampling designs, Pearson residuals are a common tool for evaluating discrepancies between observed values and expected values under the fitted model. Following Heeringa et al. [2017], they are defined as:

$$r_k = \frac{y_k - \mu_k}{\sqrt{\widehat{Var}(\mu_k)/w_k}},$$

where $\hat{\mu}_k = \mathbf{x}_k^\top \hat{\beta}$ represents the expected value of y_k under the fitted model, w_k corresponds to the sampling weight associated with unit k, and $\widehat{Var}(\hat{\mu}_k)$ denotes the estimated variance under the model family.

Pearson residuals are also used to evaluate key aspects of model fit, particularly the normality and homogeneity of the error variance. As a complement, the plot of residuals against fitted values is an especially informative diagnostic tool, since visual inspection makes it possible to detect possible systematic patterns suggesting violations of the assumptions of independence or heteroscedasticity.

The assumption of normality of the errors should also be checked explicitly. A standard tool for this purpose is the quantile-quantile plot (QQ plot), which compares the quantiles of the observed residuals with the theoretical quantiles of a normal distribution with the same mean and variance. An approximate alignment of the points along the 45° line indicates that the normality assumption is reasonable within the context of the model.

These diagnostics are applied below to the previously fitted models. For this purpose, the `svydiags` package [Valliant, 2024] is used; it was developed as an extension of the `survey` package for

diagnosing models estimated under complex sampling designs. This package makes it possible to obtain standardized residuals directly, facilitating the evaluation of model assumptions in this type of context:

```
library(svydiags)
stdresids <- as.numeric(svystdres(fit_svy)$stdresids)
survey_design$variables <- survey_design$variables %>%
  mutate(stdresids = stdresids)
```

The normality analysis can be complemented with the histogram of standardized residuals, presented in Figure 5.2. The histogram of standardized residuals reveals a significant lack of fit with respect to the theoretical normal distribution curve (in red), which corroborates a violation of the normality assumption. The distribution of the residuals (shown in blue) has a much sharper and narrower peak than the normal curve, as well as marked positive skewness with an extended right tail. This discrepancy indicates that the model fails to capture the real variability of the data, suggesting the need to apply a transformation of the dependent variable or to use a model with a more flexible and robust distribution family.

```
ggplot(
  data = survey_design$variables,
  aes(x = stdresids)
) +
  geom_histogram(
    aes(y = ..density..),
    colour = "black",
    fill = "blue",
    alpha = 0.3
  ) +
  geom_density(size = 1, colour = "blue") +
  geom_function(fun = dnorm, colour = "red", size = 1) +
  labs(y = "")
```

Homoscedasticity of the errors (constant variance) is one of the central assumptions in regression models. Violating this assumption affects the efficiency of the model estimators. Since residuals represent the discrepancies between the observed values and the values fitted by the model, analyzing them as a function of predicted values or covariates is a fundamental diagnostic tool. Under an appropriately specified model, the residuals are expected to be randomly distributed around zero, without structured patterns; by contrast, the appearance of systematic shapes (such as funnel structures or curvatures) suggests the presence of heteroscedasticity (nonconstant variance) or possible functional relationships not captured by the model specification.

To evaluate this assumption, it is recommended to analyze the residuals graphically as a function of the estimated values $\hat{\mu}_k$ or of specific covariates in the model. The appearance of systematic patterns in these plots constitutes evidence of heteroscedasticity or possible functional specification problems. In particular, to evaluate homoscedasticity with respect to the covariates included in the model, plots of residuals against each covariate are constructed and combined with `patchwork` [Pedersen, 2025].

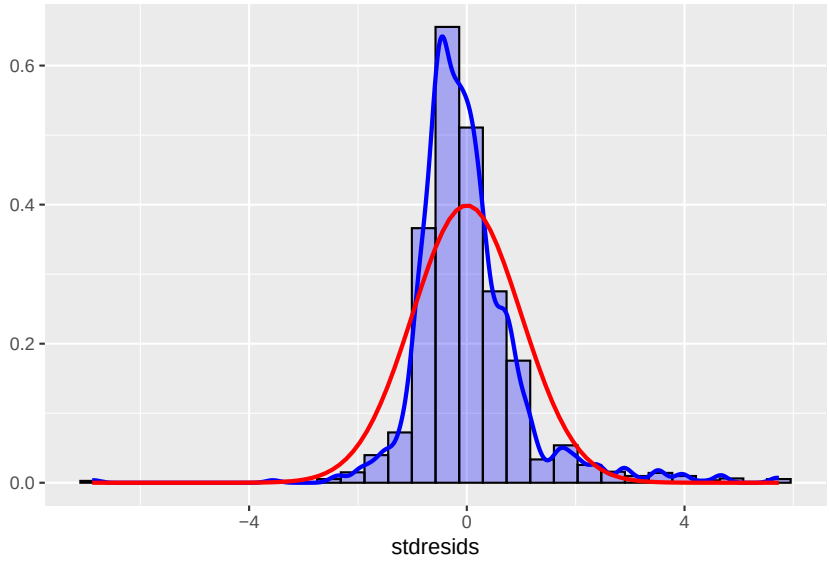

Figure 5.2: Histogram of standardized residuals with estimated density curve and theoretical normal distribution

```
library(patchwork)

g1 <- ggplot(
  data = survey_design$variables,
  aes(x = Expenditure, y = stdresids)
) +
  geom_point() +
  geom_hline(yintercept = 0)

g2 <- ggplot(
  data = survey_design$variables,
  aes(x = Zone, y = stdresids)
) +
  geom_point() +
  geom_hline(yintercept = 0)

g3 <- ggplot(
  data = survey_design$variables,
  aes(x = Sex, y = stdresids)
) +
  geom_point() +
  geom_hline(yintercept = 0)

(g1 | g2 | g3)
```

Figure 5.3 presents the scatter plots of standardized residuals against the model covariates. Taken together, these results suggest possible limitations in the functional specification. In particular, for

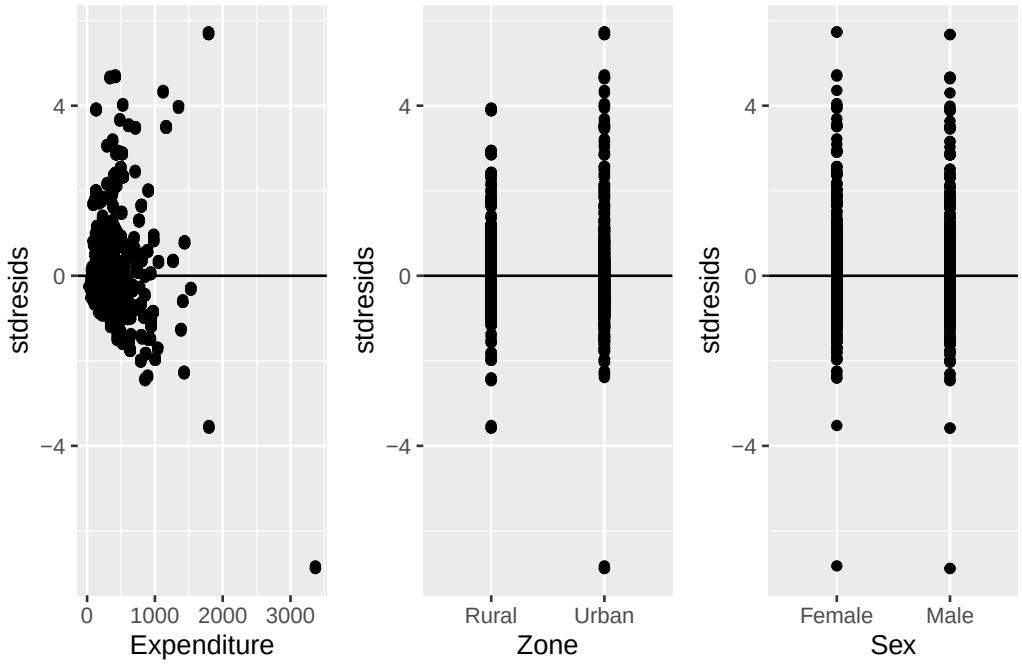

Figure 5.3: Plots of standardized residuals against each model covariate

`Expenditure`, a slight curvature is observed in the trend of the residuals, which could indicate the presence of a nonlinear relationship not fully captured by the model. In turn, the plots associated with `Zone` and `Sex` show unequal dispersion of residuals across categories, suggesting differences in conditional variability or in the mean fit between groups.

C Influential observations

In diagnostic analysis, identifying influential observations is a technique of particular importance. These are sample units whose impact on the model fit is disproportionate relative to the rest of the sample. It is worth emphasizing that an influential observation does not necessarily correspond to an outlier: while an outlier may be far from the general pattern of the data without substantially affecting the fit, an influential observation can significantly alter the parameter estimates, even if it does not appear unusual.

Cook's distance measures the impact that removing observation k would have on the overall model fit, by simultaneously combining the magnitude of the residual, the estimated variance of the model, and its leverage. In the context of surveys with complex sampling designs, Cook's distance is denoted as c_k and its expression explicitly incorporates the sampling weights [Heeringa et al., 2017, page 213]. To evaluate the magnitude of this statistic, an approximation based on comparison with an F distribution is used, through the expression

$$\frac{(df - p + 1)\, c_k}{df} \sim F_{(p,\, df-p)},$$

where p denotes the model parameters and df represents the degrees of freedom associated with the sampling design. In practice, an observation may be considered influential if its Cook's distance

exceeds approximate reference values between 2 and 3, as proposed in the literature [Heeringa et al., 2017].

In R, the presence of influential observations is evaluated using Cook's distance, estimated with the `svyCooksD` function from the `svydiags` package. The results obtained are presented in Figure 5.4, where it is observed that none of the Cook's distances exceeds the reference threshold of 3; therefore, under this criterion, no observations with significant influence on the model fit are identified.

```
svyCooksD(fit_svy, doplot = TRUE) %>%
  head(10)
```

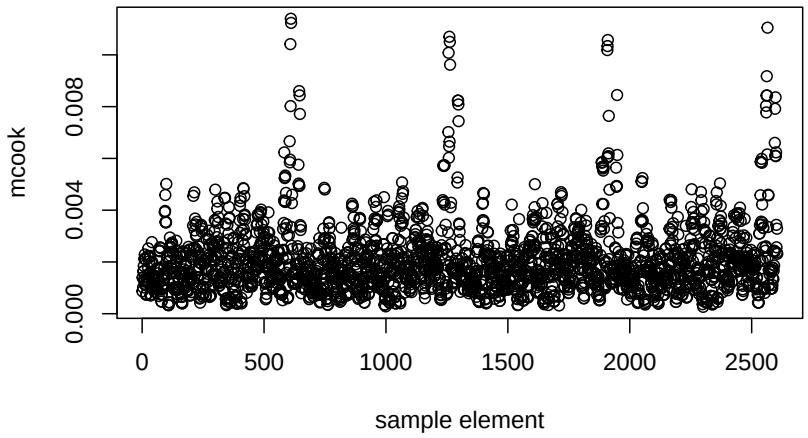

Figure 5.4: Cook's distance for each sample observation

```
##           1           2           3           4           5           6           7           8
## 0.0008667 0.0013961 0.0013979 0.0010179 0.0016359 0.0016423 0.0018452 0.0013093
##           9          10
## 0.0021197 0.0012425
```

Another statistic that measures the degree of influence of an observation is DFBETA, denoted as $D_f\text{Beta}_{(k)j}$, which measures the change in the regression coefficient $\hat{\beta}_j$ when observation k is removed from the estimation process. Consequently, this measure evaluates the individual influence of each observation on the model parameters, identifying units whose presence substantially alters the estimates obtained. High values of $D_f\text{Beta}_{(k)j}$ indicate that the observation exerts considerable influence on the corresponding coefficient, which can affect the stability and interpretation of the model.

According to Li and Valliant [2015], as a practical diagnostic criterion, an observation is usually considered influential for the coefficient $\hat{\beta}_j$ when

$$\left|D_f \text{Beta}_{(k)j}\right| \geq \frac{z}{\sqrt{n_I \times DEFF}},$$

where z commonly takes values of 2 or 3, n_I represents the number of PSUs selected in the first-stage sample, and $DEFF$ corresponds to the design effect. The influence of the observations on the coefficients of the example model is evaluated below using the `svydfbetas` function. The results corresponding to the first ten observations are presented in Table 5.1.

```
d_dfbetas <- data.frame(t(svydfbetas(fit_svy)$Dfbetas))
colnames(d_dfbetas) <- paste0("Beta_", 1:4)
d_dfbetas %>%
  slice(1:10L)
```

Table 5.1: DfBetas values for the first ten sample observations

Beta_1	Beta_2	Beta_3	Beta_4
0.000	0	0.001	-0.003
-0.001	0	0.001	0.002
-0.001	0	0.001	0.002
0.000	0	0.001	-0.003
-0.001	0	0.001	0.002
0.003	0	-0.003	-0.008
0.003	0	-0.003	-0.008
0.000	0	-0.003	0.008
0.003	0	-0.003	-0.008
-0.001	0	0.001	-0.003

Once the values of the DFBETA measure have been obtained for each observation and model coefficient, the corresponding influence threshold is calculated from the results generated by the `svydfbetas` function. Based on this criterion, a dichotomous variable is constructed to classify observations according to their level of influence on the model estimates. The results of this classification are presented in Table 5.2.

```
d_dfbetas$id <- 1:nrow(d_dfbetas)
d_dfbetas <- reshape2::melt(d_dfbetas, id.vars = "id")
cutoff <- svydfbetas(fit_svy)$cutoff
d_dfbetas <- d_dfbetas %>%
  mutate(criterion = ifelse(abs(value) > cutoff, "Yes", "No"))

label_text <- d_dfbetas %>%
  filter(criterion == "Yes") %>%
  arrange(desc(abs(value))) %>%
  slice(1:10L)

label_text
```

Table 5.2: The ten most influential observations according to the DfBetas criterion

id	variable	value	criterion
889	Beta_1	0.284	Yes
891	Beta_1	0.284	Yes
890	Beta_2	-0.280	Yes
889	Beta_2	-0.275	Yes
891	Beta_2	-0.275	Yes
890	Beta_1	0.250	Yes
890	Beta_3	0.161	Yes
890	Beta_4	0.158	Yes
889	Beta_3	0.156	Yes
891	Beta_3	0.156	Yes

The results show that several observations exceed the threshold established by the $D_f\text{Beta}_{(k)j}$ criterion, indicating that these units exert substantial influence on the estimation of some model coefficients and suggesting the presence of units with high impact on the stability of the fit. Figure 5.5 presents the graphical representation of the D_fBeta values together with the reference threshold used to identify influential observations. In the figure, the points highlighted in red correspond to the observations that exceed this threshold and therefore require more detailed review within the model diagnostic process.

```
ggplot(d_dfbetas, aes(y = abs(value), x = id)) +
  geom_point(aes(col = criterion)) +
  geom_text(
    data = label_text,
    angle = 45,
    vjust = -1,
    aes(label = id)
  ) +
  geom_hline(aes(yintercept = cutoff)) +
  facet_wrap(. ~ variable, nrow = 2) +
  scale_color_manual(
    values = c("Yes" = "red", "No" = "black")
  )
```

Finally, the DFFITS statistic, denoted by $D_f\text{Fits}_{(k)}$, measures the influence that observation k exerts on the fitted values of the model. In particular, this statistic evaluates how much the model predictions change when an observation is removed from the estimation process. Consequently, high values of $D_f\text{Fits}_{(k)}$ indicate observations capable of substantially altering the overall model fit and, therefore, potentially influential in the inference process. According to Li and Valliant [2015], an observation is considered influential if:

$$|D_f\text{Fits}_{(k)}| \geq z\sqrt{\frac{p}{n \times DEFF}}$$

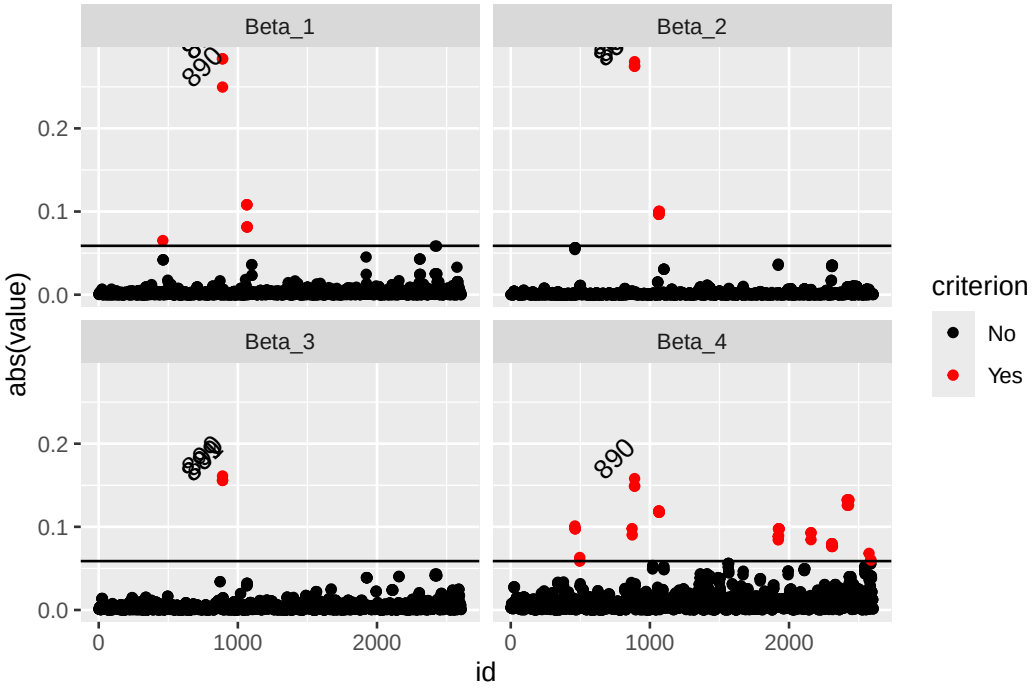

Figure 5.5: DfBetas values by parameter: influential observations indicated in red

where z commonly takes values of 2 or 3, while n represents the total sample size of elements included in the survey. In the context of complex surveys, the use of thresholds adjusted by the design effect is especially relevant, since it incorporates the loss of precision derived from stratification, clustering, and the use of unequal weights.

In R, this statistic can be calculated using the `svydffits` function from the `svydiags` package, which obtains influence measures while explicitly considering the structure of the sampling design. The corresponding graphical results are presented in Figure 5.6, where the existence of some influential observations is confirmed (indicated in red).

```
d_dffits <- data.frame(
  dffits = svydffits(fit_svy)$Dffits,
  id = 1:length(svydffits(fit_svy)$Dffits)
)
cutoff <- svydffits(fit_svy)$cutoff
d_dffits <- d_dffits %>%
  mutate(cutoff_criterion = ifelse(abs(dffits) > cutoff, "Yes", "No"))

ggplot(d_dffits, aes(y = abs(dffits), x = id)) +
  geom_point(aes(col = cutoff_criterion)) +
  geom_hline(yintercept = cutoff) +
  scale_color_manual(
    values = c("Yes" = "red", "No" = "black")
  )
```

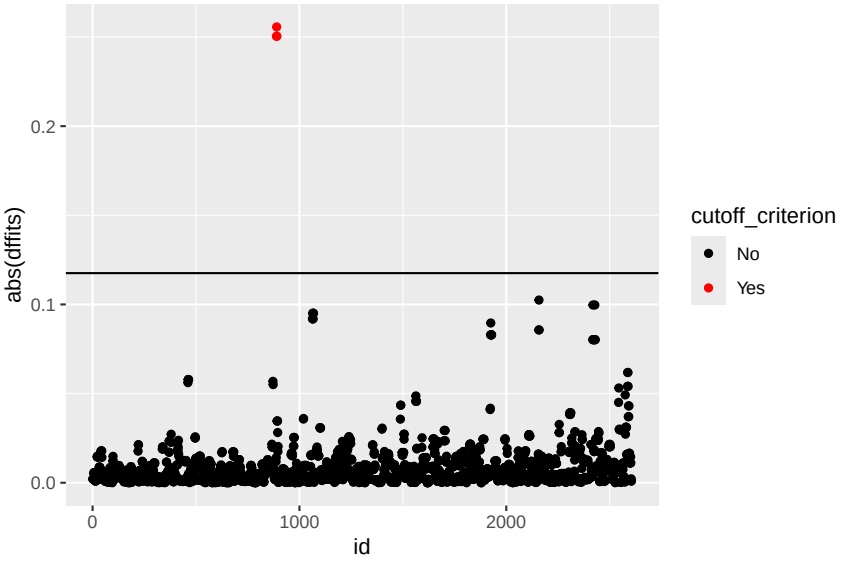

Figure 5.6: DfFits values: influential observations on the overall fit indicated in red

V Inference about model parameters

Once the model fit has been evaluated and the fulfillment of the assumptions associated with the errors has been checked, the next step is to determine the statistical significance of the estimated parameters. This analysis makes it possible to establish whether the covariates included in the model provide relevant information for explaining the variability of the response variable.

In models fitted with data from complex surveys, hypothesis tests for regression coefficients are constructed using variance estimators that are consistent with the sampling design. To evaluate the effect of a covariate associated with the parameter β_j, the following contrast is usually considered

$$H_0 : \beta_j = 0 \qquad \text{frente a} \qquad H_1 : \beta_j \neq 0.$$

The corresponding test statistic is defined as

$$t_j = \frac{\hat{\beta}_j - \beta_j}{\sqrt{\widehat{Var}(\hat{\beta}_j)}}$$

which, under the null hypothesis, approximately follows a Student's t distribution with df degrees of freedom associated with the sampling design (usually defined as the number of primary sampling units minus the number of strata). The inferential decision is based on comparing the absolute value of the statistic t_j with the critical value of the reference distribution. When the observed value of the statistic is sufficiently large in magnitude, the null hypothesis is rejected and it is concluded that the corresponding covariate has a statistically significant association with the response variable.

The same distributional properties make it possible to construct confidence intervals for the regression coefficients. Thus, a confidence interval at the $(1-\alpha) \times 100\%$ level for the parameter β_j is given by

$$\hat{\beta}_j \pm t_{1-\frac{\alpha}{2},df}\sqrt{\widehat{Var}(\hat{\beta}_j)},$$

where $t_{1-\frac{\alpha}{2},df}$ represents the quantile of order $1-\alpha/2$ of the Student's t distribution with df degrees of freedom.

To implement these procedures in `R`, the functions `summary.svyglm` and `confint.svyglm` are used; these make it possible to obtain, respectively, significance tests based on the t statistic and confidence intervals for the coefficients estimated under the complex sampling design. The corresponding results are presented in Table 5.3.

```
summary_table <- summary(fit_svy)$coefficients %>%
  as.data.frame() %>%
  tibble::rownames_to_column("Parameter")
```

Table 5.3: t tests and 95% confidence intervals for the model parameters

Parameter	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	73.580	59.470	1.237	0.218
Expenditure	1.222	0.197	6.212	0.000
ZoneUrban	66.652	39.666	1.680	0.096
SexMale	20.644	15.611	1.322	0.189

The model results indicate that the variable `Expenditure` has a statistically significant association with the variable of interest (`Income`). This effect is highly statistically significant ($p < 0.001$), providing strong evidence of a positive linear relationship between the two variables. In turn, for the categorical variables `ZoneUrban` and `SexMale`, the level of statistical significance is marginal ($p = 0.096$ and $p = 0.189$, respectively), so the evidence is not conclusive at the conventional 5% level.

VI Estimation and prediction

According to Heeringa et al. [2017], regression models serve two fundamental objectives. The first is to explain the variability of the response variable using the available covariates, a purpose that has been developed throughout this chapter. The second corresponds to predicting the expected value of the variable of interest for new observations, both within and outside the sample domain. The following expression represents the expected value of the response variable y_k conditional on the observed covariate vector $\mathbf{x}_k$. In other words, the model uses the available information from the explanatory variables to estimate the expected average value of the variable of interest for unit i.

$$\hat{\mu}_k = \hat{E}(y_k \mid \mathbf{x}_k) = \mathbf{x}_k\hat{\beta}$$

Likewise, the variance associated with the estimation of the expected value, defined as $Var(\hat{\mu}_k) = Var(\mathbf{x}_k\hat{\beta})$, can be estimated using the following expression:

$$\widehat{Var}(\hat{\mu}_k) = \mathbf{x}_k' \, \hat{\Sigma} \, \mathbf{x}_k$$

Here, $\hat{\Sigma} = \widehat{Var}(\hat{\beta})$ corresponds to the estimated variance-covariance matrix of the regression model parameters. This expression quantifies the uncertainty associated with the prediction obtained for unit k, simultaneously considering the variability of each estimated coefficient and the correlation between them. Consequently, units whose covariate values are associated with regions of greater uncertainty in coefficient estimation will have higher variances. Therefore, the precision of the estimates depends directly on the precision with which the regression model coefficients are estimated. Continuing with the example model, recall that the estimated model coefficients are presented in Table 5.4:

The model coefficients are converted to a tidy table using `broom::tidy()` [Robinson et al., 2026], which makes it easier to present estimates, standard errors, and tests in a rectangular structure.

Table 5.4: Estimated coefficients of the regression model with complex sampling design

term	estimate	std.error	statistic	p.value
(Intercept)	73.580	59.470	1.237	0.218
Expenditure	1.222	0.197	6.212	0.000
ZoneUrban	66.652	39.666	1.680	0.096
SexMale	20.644	15.611	1.322	0.189

Based on the estimated model coefficients, the estimate of the expected value of the response variable for unit k can be expressed as:

$$\hat{E}(y_k \mid \mathbf{x}_k) = 73.58 + 1.22 \times x_{1i} + 66.65 \times x_{2i} + 20.64 \times x_{3i}$$

The following code calculates the regression model predictions for the observations included in the sample, using the design matrix and the vector of estimated coefficients:

```
x_obs <- model.matrix(fit_svy) %>%
  data.frame() %>%
  as.matrix()
hat_beta <- fit_svy$coe
hat_mu <- as.numeric(x_obs %*% hat_beta)
hat_mu %>%
  head(10)
```

```
##  [1] 517.6 496.9 496.9 517.6 496.9 553.1 553.1 573.7 553.1 184.8
```

First, the instruction `model.matrix(fit_svy)` constructs the design matrix associated with the fitted model, automatically incorporating the explanatory variables. Next, the object `hat_beta` stores the vector of estimated regression model coefficients. The estimates are then obtained through the matrix product `x_obs %*% hat_beta`. Finally, the function `head(10)` makes it possible to view the first ten estimates obtained.

To calculate the variance of the estimates, the variance-covariance matrix of the model parameters is required. The elements of the main diagonal correspond to the variances of the estimated parameters, while the off-diagonal elements represent the covariances between pairs of coefficients. The estimated variance-covariance matrix is presented in Table 5.5:

```
vcov(fit_svy) %>%
  as.data.frame() %>%
  tibble::rownames_to_column("Parameter")
```

Table 5.5: Variance-covariance matrix of the model coefficients

Parameter	(Intercept)	Expenditure	ZoneUrban	SexMale
(Intercept)	3536.68	-10.551	123.689	326.13
Expenditure	-10.55	0.039	-3.295	-0.57
ZoneUrban	123.69	-3.295	1573.353	-121.45
SexMale	326.13	-0.570	-121.446	243.72

In computational terms, this involves combining the covariate vector for each observation with the matrix $\hat{\Sigma}$ to obtain the precision of the individual predictions. The following code calculates the variance associated with the predictions obtained from the regression model:

```
cov_beta <- vcov(fit_svy) %>%
  as.matrix()
mu_variance <- as.numeric(x_obs %*% cov_beta %*% t(x_obs))
mu_variance %>%
  head(10)
```

```
##  [1] 1374.9 1002.6 1002.6 1374.9 1002.6 1107.8 1107.8 1480.1 1107.8  750.8
```

The confidence interval, which provides a range of plausible values for the expected value of the response variable conditional on the observed covariates, is obtained by:

$$\mathbf{x}_k\hat{\beta} \pm t_{(1-\frac{\alpha}{2},df)}\sqrt{\mathbf{x}_k' \, \hat{\Sigma} \, \mathbf{x}_k}$$

In R, the function `predict(fit_svy, type = "link")` generates the linear predictions of the fitted model. The argument `type = "link"` indicates that the estimates are obtained on the scale of the linear predictor, that is, using directly the linear combination of the covariates and the estimated coefficients. The function `confint()` then calculates the confidence intervals associated with these predictions.

```
hat_mu <- data.frame(predict(fit_svy, type = "link"))
mu_ci <- data.frame(confint(predict(fit_svy, type = "link")))
colnames(mu_ci) <- c("lower_limit", "upper_limit")
```

The estimates obtained from the model, together with their respective confidence intervals, can be visualized graphically in Figure 5.7.

```
pred <- cbind(hat_mu, mu_ci)
pred$Expenditure <- survey_data$Expenditure
pd <- position_dodge(width = 0.2)
ggplot(
  pred %>%
    slice(1:100L),
  aes(x = Expenditure, y = link)
) +
  geom_errorbar(
    aes(ymin = lower_limit, ymax = upper_limit),
    width = .1,
    linetype = 1
  ) +
  geom_point(size = 2, position = pd) +
  theme_bw()
```

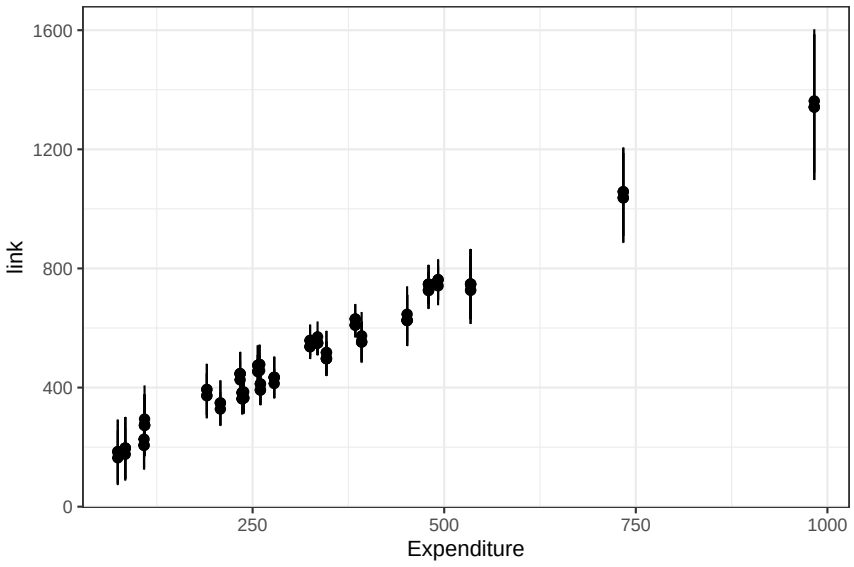

Figure 5.7: Point predictions and confidence intervals for the first 100 sample observations

When the interest is in making predictions for covariate values that fall outside the range observed in the sample, the uncertainty associated with the estimate increases. In these cases, it is not enough to consider only the variability derived from estimating the model parameters; the variability inherent to the random term of the regression must also be incorporated. For this

reason, the prediction variance includes an additional component associated with the residual error of the model:

$$\widehat{Var}\left[\hat{E}\left(y_k \mid \mathbf{x}_k\right)\right] = \mathbf{x}_k' \, \hat{\Sigma} \, \mathbf{x}_k + \hat{\sigma}^2_{yx}$$

In this expression, the term $\hat{\sigma}^2_{yx}$ corresponds to the estimated residual variance, that is, the part of the variability of the response variable that is not explained by the covariates included in the model. Incorporating this second component is fundamental when predicting for unobserved units, since it explicitly recognizes the existence of individual variability around the estimated mean.

Consequently, prediction intervals are usually wider than confidence intervals for the mean, reflecting greater uncertainty associated with the prediction of individual values. Thus, the prediction interval for a new observation can be expressed as:

$$\mathbf{x}_k\hat{\beta} \pm t_{(1-\frac{\alpha}{2},df)}\sqrt{\mathbf{x}_k^t\hat{\Sigma}_{\hat{\beta}}\mathbf{x}_k + \hat{\sigma}^2_{yx}}$$

Suppose, for example, that the goal is to obtain the model prediction for a new observation corresponding to a male individual (`Male`), residing in an urban area (`Urban`), with an expenditure level (`Expenditure`) equal to 1600. This profile can be defined in `R` using the following data set:

```
new_data <- data.frame(
  Expenditure = 1600,
  Sex = "Male",
  Zone = "Urban"
)

predict(fit_svy, newdata = new_data, type = "link")
```

```
##   link  SE
## 1 2117 243
```

```
confint(predict(fit_svy, newdata = new_data))
```

```
##   2.5 % 97.5 %
## 1  1641   2593
```

First, the object `new_data` specifies the values of the covariates included in the model. The function `predict()` then calculates the point prediction associated with that observation. The argument `newdata = new_data` indicates that the prediction should be made using the new information supplied, while `type = "link"` requests the result on the scale of the linear predictor. Finally, the function `confint()` makes it possible to obtain the confidence interval associated with the prediction made. This interval quantifies the uncertainty of the estimate and provides a range of plausible values for the expected value of the response variable corresponding to the specified profile.

Chapter 6

Generalized linear models

When the variable of interest is not continuous or the assumptions of the classical linear model are not satisfied, generalized linear models provide an appropriate alternative for modeling the relationships of interest. This approach was introduced by Nelder and Wedderburn [1972], who showed that several widely used statistical methods, such as logistic regression, log-linear models for count data, and certain advanced regression models for continuous variables, could be integrated within a single unified theoretical framework. This formulation made it possible to extend regression tools beyond the settings in which the response variable follows a normal distribution.

The need for this generalization arises because classical linear models assume, among other things, normality and homogeneity of variance in the response variable, conditions that are rarely met when categorical variables, proportions, or counts are analyzed. In these situations, generalized linear models offer a flexible framework for appropriately modeling the relationship between variables by selecting a probability distribution suitable for the response and a link function that connects the mean of that distribution with the linear predictor.

To illustrate the models presented in this chapter, the `survey_data` database and the `BigCity` population are used, following the same structure used in previous chapters. After loading `tidyverse` [Wickham et al., 2026a], `survey` [Lumley, 2024], `srvyr` [Freedman Ellis and Schneider, 2024], `broom` [Robinson et al., 2026], and `jtools` [Long, 2026], the following code block creates two new categorical variables (`poverty` and `unemployment`) from the original variables `Poverty` and `Employment`, respectively.

```
options(digits = 4)
library(tidyverse)
library(survey)
library(srvyr)
library(broom)
library(jtools)

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

survey_design <- survey_data %>%
```

```
  as_survey_design(
    strata = Stratum,
    ids = PSU,
    weights = wk,
    nest = TRUE
  )

survey_design <- survey_design %>%
  mutate(
    poverty = factor(
      ifelse(Poverty != "NotPoor", 1, 0),
      levels = c(0, 1),
      labels = c("=Poor", "=Not poor")
    ),
    unemployment = factor(ifelse(Employment == "Unemployed", 1, 0),
      levels = c(0, 1),
      labels = c("=Unemployed", "=Not unemployed")
    )
  )
```

I Logistic regression model

When the variable of interest is dichotomous, taking values of zero or one, the classical linear model is not appropriate because it can generate predictions outside the interval $(0, 1)$ and because the variance of the response depends on its mean. To address these limitations, the logistic regression model is used; it is one of the most widely used generalized linear models for analyzing binary variables. In the context of household surveys, logistic regression can be used to analyze the association between a condition of interest, such as poverty, and sociodemographic characteristics of individuals or households.

Consider a dichotomous variable $Y_k \in \{0, 1\}$, whose expectation is $\theta_k = Pr(Y_k = 1)$. The logistic regression model relates this probability to a set of explanatory variables through the logit link function, defined as

$$\log\left(\frac{\theta_k}{1-\theta_k}\right) = \beta_0 + \beta_1 x_{k1} + \cdots + \beta_p x_{kp} = \mathbf{x}_k\beta$$

Where β is the vector of regression coefficients associated with the covariates. This logit transformation maps the restricted probability interval $(0, 1)$ onto the entire real line, allowing the relationship between the response and the auxiliary variables to be modeled through a linear predictor. Equivalently, the probability of success can be expressed by the following relationship:

$$\theta_k = Pr(Y_k = 1 \mid \mathbf{x}_k) = \frac{\exp(\mathbf{x}_k\beta)}{1 + \exp(\mathbf{x}_k\beta)}$$

Equivalently, the probability that the unit does not have the characteristic of interest is $Pr(Y_k = 0 \mid \mathbf{x}_k) = 1 - \theta_k$. Under this parameterization, each coefficient in β represents the effect of a covariate on the logarithm of the odds that the event of interest occurs, holding the other variables in the model constant. The parameters can be estimated using the pseudo-maximum likelihood approach. This method incorporates the expansion factors into the conventional likelihood function in order to obtain estimators that are consistent with respect to the target population. Following Heeringa et al. [2017], the pseudo-likelihood function for the logistic model is defined as

$$PL(\beta) = \prod_k \left[\theta_k^{y_k} \{1 - \theta_k\}^{1-y_k}\right]^{w_k}$$

Where the subscript k represents an observed unit in the sample. Likewise, θ_k denotes the probability that unit k in PSU i in stratum h has the characteristic of interest, y_k corresponds to the observed value of the dichotomous variable for that unit, and w_k represents the expansion factor associated with the observation.

The contribution of each observation to the estimation process is determined by its respective expansion factor, allowing the function to appropriately reflect the characteristics of the complex survey design. Because maximizing the pseudo-likelihood is equivalent to maximizing its logarithm, the problem is therefore reduced to optimizing the following function:

$$\ell(\beta) = \sum_k w_k \left[y_k \log(\theta_k) + (1 - y_k) \log(1 - \theta_k)\right]$$

This expression is the objective function used by numerical optimization algorithms to obtain the estimator of the model parameters. It is important to note that the preceding function is not a likelihood in the strict sense under the complex survey design. For this reason, the term *pseudo-likelihood* is used, and the estimators obtained by maximizing it are known as pseudo-maximum likelihood estimators (*Pseudo Maximum Likelihood Estimators*, PMLE).

According to Molina and Skinner [1992], under regularity conditions, these estimators are consistent and asymptotically normal, forming the basis of the inference procedures implemented in most statistical packages for the analysis of complex survey data.

Once the pseudo-maximum likelihood estimators have been obtained, their variances are estimated using Taylor linearization techniques [Binder, 1983], which explicitly incorporate the characteristics of the complex sampling design. The variance-covariance matrix of the estimated parameters, denoted by $\text{Var}(\widehat{\beta}) = \Sigma$, is obtained from a linear approximation of the estimating equations around the true value of the parameters.

This matrix describes the precision of the estimators, since its diagonal elements correspond to the variances of each coefficient, while the off-diagonal elements represent the covariances between pairs of coefficients. Its estimation is the basis for calculating standard errors, constructing confidence intervals, and carrying out hypothesis tests on the model parameters.

Because the data come from a complex sample design, it is necessary to appropriately incorporate the weights, stratification, and clustering in order to obtain consistent estimators of the parameters and their standard errors. In R, this task can be carried out using the `svyglm()` function from the `survey` package, which extends generalized linear models to the context of complex surveys.

Continuing with the example survey, and in order to analyze the factors associated with poverty status, a logistic regression model is fitted using expenditure, employment status, and sex as explanatory variables. Because the data come from a survey with a complex sample design, estimation is carried out using the `svyglm()` function from the `survey` package, specifying `family = binomial` to indicate that the response variable is dichotomous and that the logit link function will be used. The results of the fit are presented in Table 6.1.

```
logit_model <- svyglm(
  formula = poverty ~ Expenditure + Employment + Sex,
  family = binomial,
  design = survey_design
)
```

In the model specification, the argument `formula = poverty ~ Expenditure + Employment + Sex` defines the relationship between the response variable `poverty` and the covariates included in the linear predictor. For categorical variables, the function automatically generates the corresponding indicator variables and estimates their effects relative to a reference category. The argument `design = survey_design`, in turn, incorporates the sample design information. The resulting object, `logit_model`, contains the estimated parameters, their precision measures, and the statistics needed to make inferences about the relationship between poverty status and the explanatory variables considered in the model.

The results in Table 6.1 show the estimates of the regression coefficients, together with their standard errors, confidence intervals, and p-values. The model indicates that a higher level of expenditure is associated with a lower probability of being in poverty. Likewise, compared with the reference category (unemployed), both inactive and employed persons have a lower propensity to be poor, with this effect being more pronounced among the latter. In contrast, sex has a very small influence on the probability of poverty once the effect of the other covariates included in the model is controlled for.

```
##                          2.5 %    97.5 %
## (Intercept)           1.448618  2.926867
## Expenditure          -0.007044 -0.003752
## EmploymentInactive   -1.618341 -0.108547
## EmploymentEmployed   -2.076705 -0.737132
## SexMale              -0.309026  0.314060
```

Table 6.1: 95% confidence intervals for the parameters of the logistic model

Parameter	term	estimate	std.error	statistic	p.value
1	(Intercept)	2.188	0.373	5.863	0.000
2	Expenditure	-0.005	0.001	-6.497	0.000
3	EmploymentInactive	-0.863	0.381	-2.266	0.025
4	EmploymentEmployed	-1.407	0.338	-4.161	0.000

Parameter	term	estimate	std.error	statistic	p.value
5	SexMale	0.003	0.157	0.016	0.987

Figure 6.1 presents the distribution of the estimators of the regression coefficients. It can be seen that, for all covariates except sex, the confidence intervals do not include the value zero, providing evidence of a statistically significant association. In contrast, the confidence interval associated with the sex variable contains the value zero, so there is not enough evidence to conclude that this variable has a significant effect once the other covariates included in the model are controlled for.

To visualize the distribution of the coefficients, `plot_summs()` from the `jtools` package is used, together with `ggstance` [Henry et al., 2024] for the horizontal geometry of the intervals.

```
library(ggstance)
plot_summs(logit_model,
  scale = TRUE,
  plot.distributions = TRUE
)
```

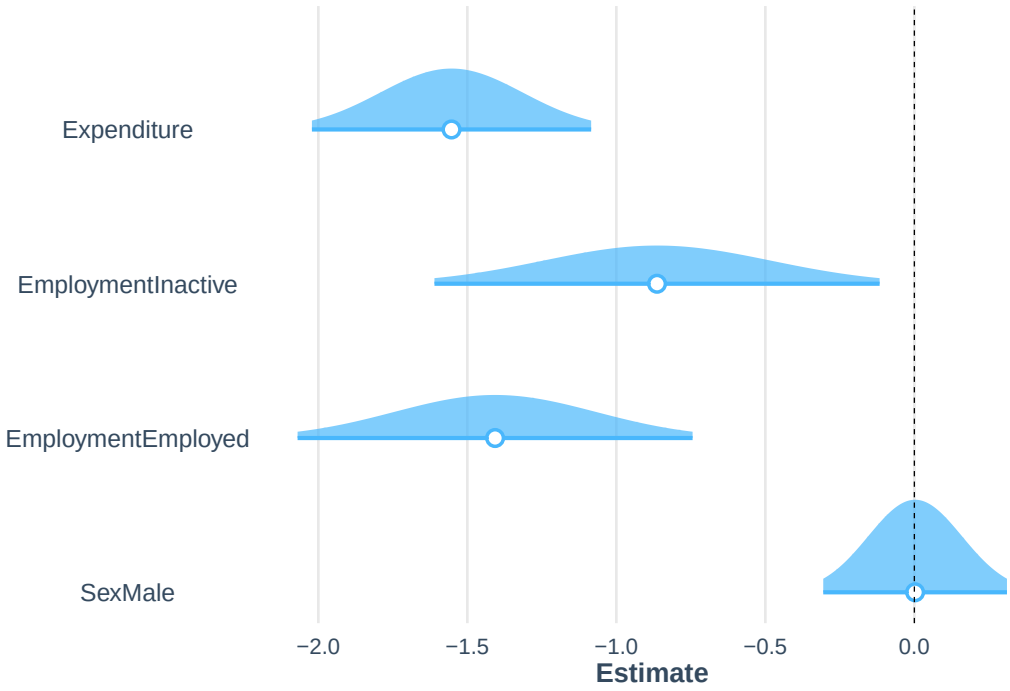

Figure 6.1: Distribution of the parameters of the baseline logistic model

The model can be extended by incorporating interaction terms to assess whether the effect of one explanatory variable depends on the values taken by another. In this case, an interaction between sex and employment status is included through the term `Sex:Employment`, with the aim of analyzing whether the relationship between employment status and the probability of poverty is the same for men and women. Without this term, the model assumes that the effect of employment status on poverty is identical for both sexes. By incorporating the interaction, however, this effect

is allowed to vary between men and women, capturing possible differences in the way labor-market status is associated with poverty in each group. The results of the fitted model are presented in Table 6.2.

```
interaction_logit_model <- svyglm(
  formula = poverty ~ Expenditure + Employment + Sex + Sex:Employment,
  family = binomial,
  design = survey_design
)
```

Table 6.2: Coefficients of the logistic model with interactions

term	estimate	std.error	statistic	p.value
(Intercept)	1.770	0.610	2.900	0.004
Expenditure	-0.005	0.001	-6.473	0.000
EmploymentInactive	-0.395	0.594	-0.665	0.507
EmploymentEmployed	-1.059	0.570	-1.860	0.066
SexMale	0.587	0.748	0.785	0.434
EmploymentInactive:SexMale	-0.846	0.860	-0.984	0.327
EmploymentEmployed:SexMale	-0.481	0.760	-0.633	0.528

When the interaction between sex and employment status is incorporated, expenditure continues to show a negative and statistically significant association with the probability of poverty, remaining the covariate with the strongest explanatory evidence in the model. In contrast, the main effects of employment status lose statistical importance once the interaction terms are included, suggesting greater uncertainty in the estimation of their effects. Neither the main effect of sex nor the coefficients associated with the interactions between sex and employment status are statistically significant at the 5% level. Consequently, the inclusion of the interaction terms does not appear to provide additional relevant information compared with the model without interaction.

Figure 6.2 presents a comparison of the coefficient estimates and their corresponding confidence intervals for the models with and without interaction. It can be seen that incorporating the interaction terms increases the uncertainty associated with several of the parameters, as reflected in wider confidence intervals. Likewise, the interaction coefficients have estimates close to zero and wide confidence intervals that include that value, which is consistent with the lack of statistical evidence for concluding that the effect of employment status on the probability of poverty differs between men and women.

```
plot_summs(interaction_logit_model, logit_model,
  scale = TRUE,
  plot.distributions = TRUE
)
```

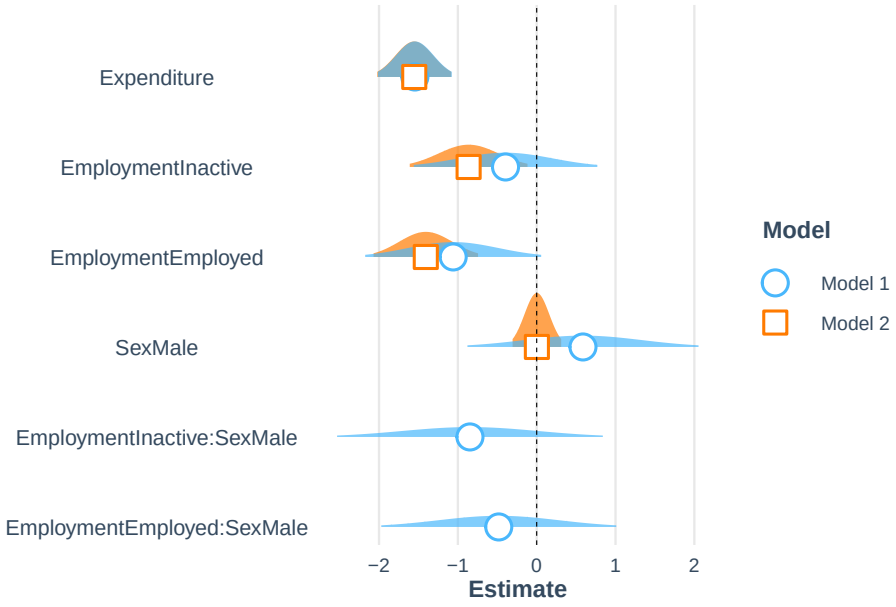

Figure 6.2: Comparison of the parameters of the logistic model with and without interactions

II Multinomial regression model

In household surveys, it is common to find response variables with more than two categories, such as employment status, whose possible states include employed, unemployed, and inactive. When these categories are mutually exclusive and do not have a natural order, an appropriate tool for modeling their relationship with a set of covariates is multinomial logistic regression, which is an extension of the binary logistic regression model.

This model makes it possible to estimate the probability of belonging to each category of the response variable as a function of continuous or categorical explanatory variables. Its application requires the response categories to be exhaustive and mutually exclusive and requires that there be no severe multicollinearity among the covariates. In addition, for continuous covariates, a linear relationship is assumed between them and the logarithm of the odds ratios (*log-odds*) of each category relative to a reference category.

The multinomial logistic model specifies the probability $\theta_{k(j)}$ that unit k belongs to category j, with $j = 1, \ldots, J$, given the covariate vector $\mathbf{x}_k$, through the following expression:

$$Pr(Y_k = j \mid \mathbf{x}_k) = \theta_{k(j)} = \frac{\exp(\mathbf{x}_k \beta_j)}{\sum_{l=1}^{J} \exp(\mathbf{x}_k \beta_l)}$$

Where β_j is the vector of regression coefficients for the j-th category. This formulation directly expresses the probabilities of belonging to each category; in practice, it is more convenient to rewrite the model using a reference category. Alternative parameterizations facilitate the interpretation of the parameters, since each set of coefficients describes the effect of the covariates on the logarithm of the odds ratios (*log-odds*) of belonging to a given category compared with the reference category.

The model parameters are estimated using the pseudo-maximum likelihood approach. Following Heeringa et al. [2017], the multinomial pseudo-likelihood function is defined as

$$PL(\beta \mid \mathbf{X}) = \prod_{k} \left\{ \prod_{j=1}^{J} \theta_{k(j)}^{y_{k(j)}} \right\}^{w_k}$$

Where k represents the observation units belonging to the complex sample. Likewise, $y_{k(j)}$ is an indicator variable that takes the value one when that unit belongs to category j and zero otherwise, and w_k corresponds to the expansion factor associated with the observed unit. For mathematical and computational convenience, the parameters are estimated from the pseudo-log-likelihood, whose maximization leads to the same estimators as the original pseudo-likelihood.

$$\ell(\beta) = \sum_{k} w_k \sum_{j=1}^{J} y_{k(j)} \log(\theta_{k(j)})$$

As mentioned above, the estimators obtained through this procedure are consistent and asymptotically normal under regular conditions [Molina and Skinner, 1992]. Statistical inference for the parameters of interest is based on variance estimates obtained through Taylor linearization techniques that explicitly incorporate the characteristics of the complex sample design, making it possible to calculate standard errors, confidence intervals, and hypothesis tests appropriate for survey data [Binder, 1983].

For fitting the model using `R`, as an initial reference, the proportions of persons by employment status are estimated, restricting the units of interest to those older than 15 years, as presented in Table 6.3.

```
survey_design %>%
  filter(Age >= 15) %>%
  group_by(Employment) %>%
  summarise(proportion = survey_mean(vartype = c("se", "ci")))
```

Table 6.3: Estimated proportion of persons by employment status (older than 15 years)

Employment	proportion	proportion_se	proportion_low	proportion_upp
Unemployed	0.043	0.007	0.029	0.057
Inactive	0.384	0.015	0.354	0.414
Employed	0.573	0.014	0.545	0.601

To model a multinomial response variable, the `svy_vglm()` function from the `svyVGAM` package [Lumley, 2026] is used; it allows multinomial regression models to be fitted while explicitly incorporating the characteristics of the complex survey design. In this example, a model is specified in which activity status is explained as a function of age, sex, and area of residence. Unlike conventional multinomial models, `svy_vglm()` calculates the variances of the estimators while accounting for the complexity of the sample design, including stratification, clustering, and expansion weights. In this way, both the point estimates and their standard errors and associated inferences are consistent with the survey design.

```r
library(svyVGAM)
survey_design_15 <- survey_design %>%
  filter(Age >= 15)
multinomial_model <- svy_vglm(
  formula = Employment ~ Age + Sex + Zone,
  design = survey_design_15,
  crit = "coef",
  family = multinomial(refLevel = "Unemployed")
)
```

The argument `family = multinomial(refLevel = "Unemployed")` defines a multinomial logistic model using the `Unemployed` category as the reference. Under this specification, the logarithms of the odds ratios (*log-odds*) between each of the `Employment` categories and the reference category are estimated simultaneously. The parameters are obtained through pseudo-maximum likelihood, incorporating the survey expansion factors into the objective function.

Because `broom::tidy()` cannot be applied directly to objects of class `svyVGAM`, the following helper function is defined to structure the model results in a standardized tabular format[1]:

```r
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)
}
```

Table 6.4 presents the estimated coefficients of the multinomial logistic model, using the *Unemployed* category as the reference. The results show that age has a positive and statistically significant effect on the relative probabilities of belonging to categories 1 and 2 compared with unemployment. Likewise, the variable `SexMale` has negative and significant coefficients, indicating that men have lower relative probabilities than women of being in those categories. By contrast, the area-of-residence variable does not show statistically significant effects at the conventional 5% level.

[1] Adapted from https://tech.popdata.org/pma-data-hub/posts/2021-08-15-covid-analysis/

```
tidy.svyVGAM(multinomial_model, exponentiate = FALSE, conf.int = FALSE)
```

Table 6.4: Estimated coefficients of the multinomial regression model

y.level	term	estimate	std.error	statistic	p.value
1	(Intercept)	2.556	0.550	4.648	0.000
1	Age	0.022	0.010	2.117	0.034
1	SexMale	-2.185	0.317	-6.901	0.000
1	ZoneUrban	-0.254	0.404	-0.627	0.531
2	(Intercept)	2.306	0.461	4.998	0.000
2	Age	0.018	0.009	2.056	0.040
2	SexMale	-0.578	0.277	-2.091	0.037
2	ZoneUrban	0.039	0.361	0.109	0.913

Figure 6.3 shows the confidence intervals of the estimated coefficients for each equation of the multinomial model. The visualization is built with `dotwhisker::dwplot()` [Solt and Hu, 2025], which makes it possible to compare coefficients and intervals on a common scale, and `gtools::stars.pval()` [Warnes et al., 2023] is used to encode statistical significance. Covariates whose intervals do not include the value zero show evidence of a statistically significant association with the response variable, while those whose intervals contain zero do not show significant effects at the confidence level considered.

```
multinomial_results %>%
  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 = 0, colour = "grey60", linetype = 2)
  ) +
  guides(color = guide_legend(reverse = TRUE)) +
  theme_bw() +
  theme(legend.position = "top")
```

III Gamma regression model

Gamma regression is an extension of generalized linear models suitable for continuous, strictly positive response variables whose variability increases with the mean. The Gamma model is an appropriate alternative for analyzing strictly positive continuous variables whose variability increases as their expected value grows. This behavior is common in economic and social variables, such as household income or consumption expenditure.

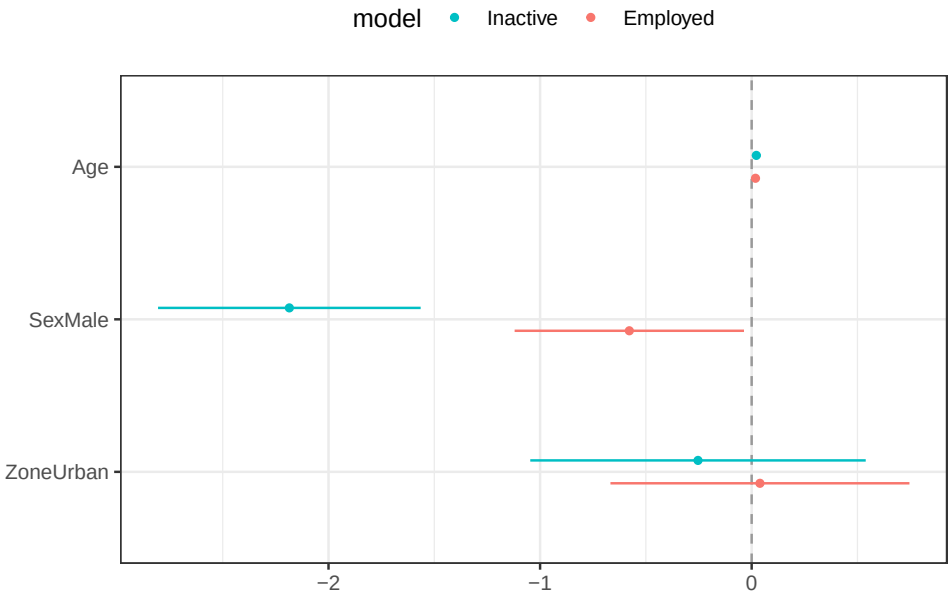

Figure 6.3: Confidence intervals for the coefficients of the multinomial model

If Y_k, the variable of interest observed for unit k, follows a Gamma distribution with mean μ_k and dispersion parameter ϕ, the model relates the expected value of the response variable to a set of covariates through a link function. The most commonly used link function is the logarithmic link, which ensures that predictions are always positive. In this case, the model is expressed as

$$\log(\mu_k) = \mathbf{x}_k\beta$$

Where $\mathbf{x}_k$ corresponds to the vector of explanatory variables associated with the observed unit and β is the vector of unknown model parameters. Equivalently, we have:

$$\mu_k = E(Y_k \mid \mathbf{x}_k) = \exp(\mathbf{x}_k\beta),$$

With the logarithmic link, each parameter can be interpreted as the effect of a covariate on the logarithm of the expected mean, so that $\exp(\beta_j)$ represents the multiplicative change associated with a one-unit increase in the corresponding covariate, holding the other variables in the model constant. Under the Gamma distribution, the conditional density function of Y_k can be written as

$$f(y_k \mid \mu_k, \phi) = \frac{1}{\Gamma(1/\phi)(\phi\mu_k)^{1/\phi}} y_k^{\frac{1}{\phi}-1} \exp\left(-\frac{y_k}{\phi\mu_k}\right), \qquad y_k > 0,$$

where $\Gamma(\cdot)$ denotes the Gamma function and ϕ represents the dispersion parameter. Under the inference approach based on pseudo-maximum likelihood, the objective function optimized to obtain the estimators of the model parameters is given by

$$\ell(\beta, \phi) = \sum_k w_k \log\left[f(y_{hik} \mid \mu_k, \phi)\right]$$

Where k represents the observation units in the sample and w_k corresponds to the expansion factor associated with each of them. This function defines the optimization criterion used to estimate the model parameters under the pseudo-maximum likelihood approach. In R, it is first necessary to define the sampling design.

```
gamma_design <- survey_data %>%
  as_survey_design(
    strata = Stratum,
    ids = PSU,
    weights = wk,
    nest = TRUE
  )
```

The following code fits a Gamma regression model for the household income variable (`Income`), incorporating the complex survey design defined in the `gamma_design` object. The model uses household expenditure (`Expenditure`), age (`Age`), sex (`Sex`), and area of residence (`Zone`) as explanatory variables. A logarithmic link function is also specified, ensuring positive predictions and allowing the effects of the explanatory variables to be interpreted in multiplicative terms on expected income. The estimated coefficients are presented in Table 6.5:

```
gamma_model <- svyglm(
  formula = Income ~ Expenditure + Age + Sex + Zone,
  design = gamma_design,
  family = Gamma(link = "log")
)
```

Table 6.5: Coefficients of the Gamma model for household income

term	estimate	std.error	statistic	p.value
(Intercept)	5.409	0.098	55.184	0.000
Expenditure	0.002	0.000	16.891	0.000
Age	0.002	0.001	2.430	0.017
SexMale	0.048	0.038	1.260	0.210
ZoneUrban	0.151	0.089	1.701	0.092

Because the model uses a logarithmic link, the coefficients can be interpreted as multiplicative effects on expected income. The coefficient associated with household expenditure is positive and statistically significant ($p < 0.001$), indicating that, holding the other variables in the model constant, a one-unit increase in expenditure is associated with an approximate 0.2% increase in expected income, since $\exp(0.002) \approx 1.002$. Age also has a positive and significant effect ($p = 0.017$). On average, each additional year of age is associated with an increase of around 0.2% in expected income, holding the other covariates constant. By contrast, the sex and area-of-residence variables do not show sufficient statistical evidence to conclude that they influence income.

Chapter 7

Multilevel Models

Multilevel models, also known as hierarchical models, are a statistical technique designed for the analysis of data with a hierarchical structure, such as those from household surveys, where the observation units are not independent of each other. Individuals belong to households, households are located in specific geographic areas (PSUs) and these, in turn, are part of broader territorial units (strata or domains). As a consequence, observations that share the same context tend to present characteristics that are more similar to each other than those belonging to different contexts. This dependence structure violates one of the fundamental assumptions of conventional regression models and can lead to inefficient estimates and incorrect inferences.

Unlike conventional regression models, multilevel models recognize that observations can be grouped at different levels and that each of them can contribute to explaining the observed variability. Consequently, they allow the effects associated with the characteristics of the units of analysis and those derived from the environment or context in which they are located to be simultaneously represented. This formulation is especially useful when seeking to study phenomena determined both by individual factors and by characteristics of households, communities, or geographic areas.

Another important feature of multilevel models is that they make it possible to quantify the proportion of variability associated with each level of grouping present in the data. While fixed effects summarize the average relationship between the response variable and the covariates included in the model, random effects represent systematic differences between groups that are not explained by these covariates. Thanks to this structure, it is possible to explicitly model the dependence between observations belonging to the same group and obtain more appropriate inferences when the data have a hierarchical organization.

The theoretical and methodological development of multilevel models has been widely documented in the sampling literature. Among the most influential references are the works of Goldstein [2011], Gelman and Hill [2006], and Rabe-Hesketh and Skrondal [2012], which present the conceptual foundations, estimation strategies, and various applications of these models in social and demographic contexts. In addition, Browne and Draper [2006] compare different estimation approaches for hierarchical models, evaluating the differences between methods based on maximum likelihood and Bayesian approaches. In the field of social epidemiology, Merlo et al. [2006] emphasize the usefulness of multilevel models for studying contextual phenomena, showing how factors associated with the environment can contribute to explaining inequalities in health and other outcomes of population interest.

I Motivation

To begin this chapter, `tidyverse` [Wickham et al., 2026a] is loaded for data manipulation and graph production, and `lme4` [Bates et al., 2026] for multilevel model estimation. The database that will be used throughout the examples is imported. The mathematical titles of some figures are formatted with `latex2exp` [Meschiari, 2026], called explicitly in the code.

```
library(tidyverse)
library(lme4)
```

```
survey_data <- readRDS("Data/encuesta.rds") %>%
  mutate(poverty = ifelse(Poverty != "NotPoor", 1, 0))
```

For illustrative purposes, the examples in this section are developed from a subsample of the survey. The following code block builds the database used in the chapter examples. To do so, it identifies the three strata with the smallest number of households and retains only the variables required for the analysis: income, expenditure, stratum, sex, region, area of residence, and poverty status.

```
plot_data <- survey_data %>%
  select(HHID, Stratum) %>%
  distinct() %>%
  group_by(Stratum) %>%
  tally() %>%
  arrange(n) %>%
  select(-n) %>%
  slice(1:3L) %>%
  inner_join(survey_data, by = "Stratum") %>%
  select(Income, Expenditure, Stratum, Sex, Region, Zone, poverty)
```

As a starting point, a conventional linear regression model is fitted that relates household income to household expenditure, temporarily ignoring the hierarchical structure of the data. Under this specification, it is assumed that all observations are independent and that the relationship between income and expenditure is homogeneous for all households, regardless of the stratum to which they belong. Figure 7.1 presents the estimated regression line along with the observed data.

```
ggplot(data = plot_data,
       aes(y = Income, x = Expenditure)) +
  geom_jitter() +
  geom_smooth(formula = y ~ x, method = "lm", se = FALSE) +
  ggtitle(latex2exp::TeX(
    "$Income_{i} \\sim \\hat{\\beta}_{0} + \\hat{\\beta}_{1}Expenditure_{i} + \\epsilon_{i
  )) +
  theme(
    legend.position = "none",
    plot.title = element_text(hjust = 0.5)
  )
```

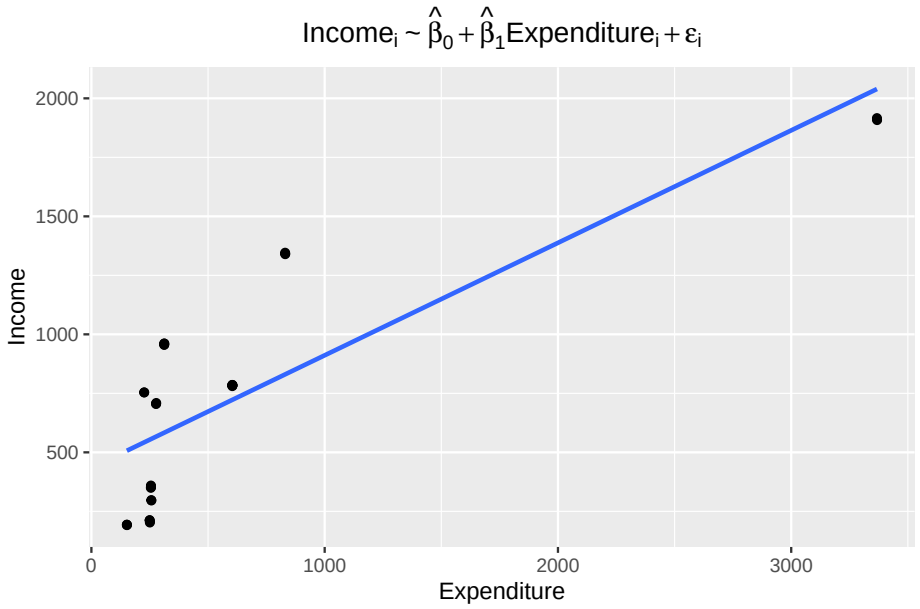

Figure 7.1: Simple linear regression model for income as a function of expenditure (without considering strata)

While this model is useful as an initial reference, its assumptions are often unrealistic in the context of household surveys. In particular, independence between observations may be compromised when households belong to strata that share similar socioeconomic, demographic, or geographic characteristics. Likewise, it is possible that average income levels differ systematically between strata, even when the relationship between income and expenditure is similar. As Finch et al. [2019] point out, ignoring this hierarchical structure can lead to incorrect estimates of standard errors and, consequently, unreliable statistical inferences.

For illustrative purposes only, the hierarchical structure of the data is gradually incorporated into the modeling process. To do so, a model is initially fitted that allows each stratum to have its own intercept, while maintaining a common slope for all observations. This specification is useful for visualizing how average income levels may differ between strata, even when the relationship between income and expenditure is considered constant.

```
beta_1 <- coef(lm(Income ~ Expenditure, data = plot_data))[2]
model_coef <- plot_data %>%
  group_by(Stratum) %>%
  summarise(beta_0 = coef(lm(Income ~ Expenditure))[1]) %>%
  mutate(beta_1 = beta_1)
```

In this way, the model introduces a first source of heterogeneity between groups and makes it possible to appreciate the limitations of the simple regression model presented previously, which assumed a single regression line for the entire population. Figure 7.2 presents the regression lines with intercepts differentiated by stratum:

```
ggplot(data = plot_data,
       aes(y = Income, x = Expenditure, colour = Stratum)) +
  geom_jitter() +
  geom_abline(data = model_coef,
              mapping = aes(slope = beta_1, intercept = beta_0, colour = Stratum)) +
  ggtitle(latex2exp::TeX(
    "$Income_{kj} \\sim \\hat{\\beta}_{0j} + \\hat{\\beta}_{1}Expenditure_{kj} + \\epsilon
  )) +
  theme(
    legend.position = "none",
    plot.title = element_text(hjust = 0.5)
  )
```

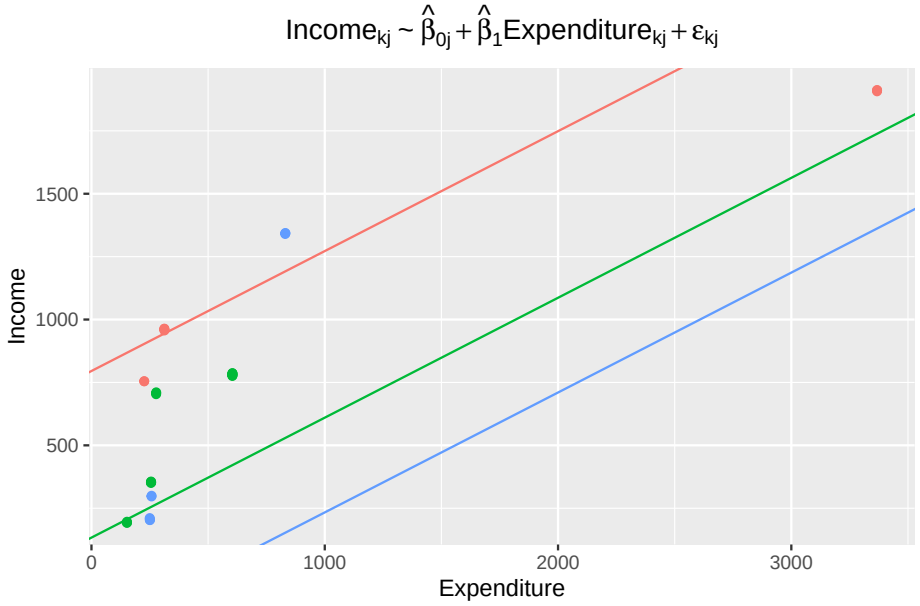

Figure 7.2: Model with stratum-specific intercept and common slope

As a next step, an alternative specification is considered in which all strata share the same average level of income, represented by a common intercept, but the relationship between income and expenditure is allowed to vary between strata through slopes specific to each stratum.

```
beta_0 <- coef(lm(Income ~ Expenditure, data = plot_data))[1]
model_coef <- plot_data %>%
  group_by(Stratum) %>%
  summarise(beta_1 = coef(lm(Income ~ Expenditure))[2]) %>%
  mutate(beta_0 = beta_0)
```

Under this specification, the differences between strata are manifested in the intensity of the association between income and expenditure, rather than in their average levels. Figure 7.3 makes it possible to visualize these differences in the estimated regression trajectories for each stratum.

```
ggplot(data = plot_data,
       aes(y = Income, x = Expenditure, colour = Stratum)) +
  geom_jitter() +
  geom_abline(data = model_coef,
              mapping = aes(slope = beta_1, intercept = beta_0, colour = Stratum)) +
  ggtitle(latex2exp::TeX(
    "$Income_{kj} \\sim \\hat{\\beta}_{0} + \\hat{\\beta}_{1j}Expenditure_{kj} + \\epsilon
  )) +
  theme(
    legend.position = "none",
    plot.title = element_text(hjust = 0.5)
  )
```

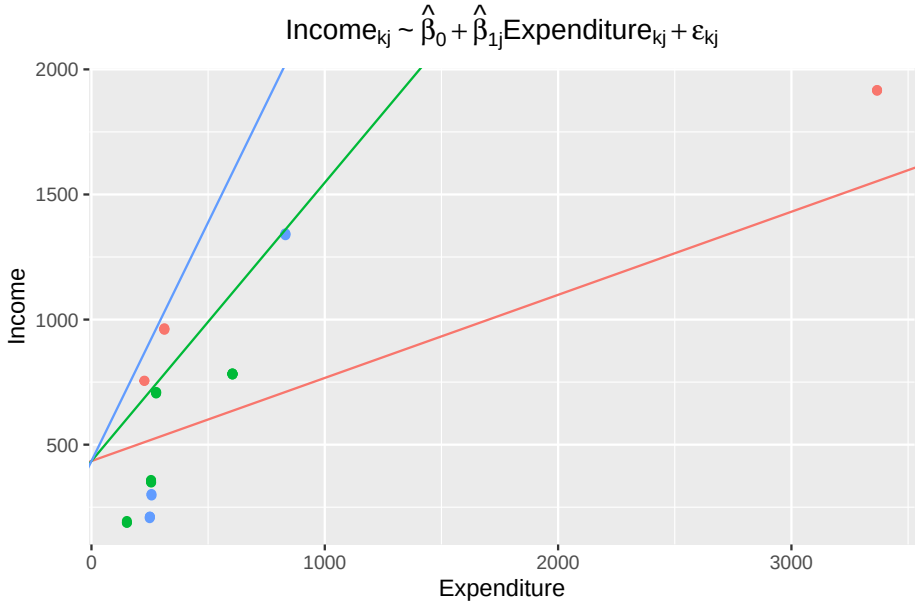

Figure 7.3: Model with stratum-specific slope and common intercept

Finally, the most flexible specification of those presented so far is considered, allowing both the average level of income and the relationship between income and expenditure to vary between strata. Under this approach, each stratum has its own regression line, with intercepts and slopes estimated independently.

```
model_coef <- plot_data %>%
  group_by(Stratum) %>%
  summarise(
    beta_0 = coef(lm(Income ~ Expenditure))[1],
    beta_1 = coef(lm(Income ~ Expenditure))[2]
  )
```

This formulation makes it possible to capture simultaneously differences in income levels and in the strength of their association with expenditure. Figure 7.4 shows the fitted lines for each stratum under this specification.

```
ggplot(data = plot_data,
       aes(y = Income, x = Expenditure, colour = Stratum)) +
  geom_jitter() +
  geom_abline(data = model_coef,
              mapping = aes(slope = beta_1, intercept = beta_0, colour = Stratum)) +
  ggtitle(latex2exp::TeX(
    "$Income_{kj} \\sim \\hat{\\beta}_{0j} + \\hat{\\beta}_{1j}Expenditure_{kj} + \\epsilo
  )) +
  theme(
    legend.position = "none",
    plot.title = element_text(hjust = 0.5)
  )
```

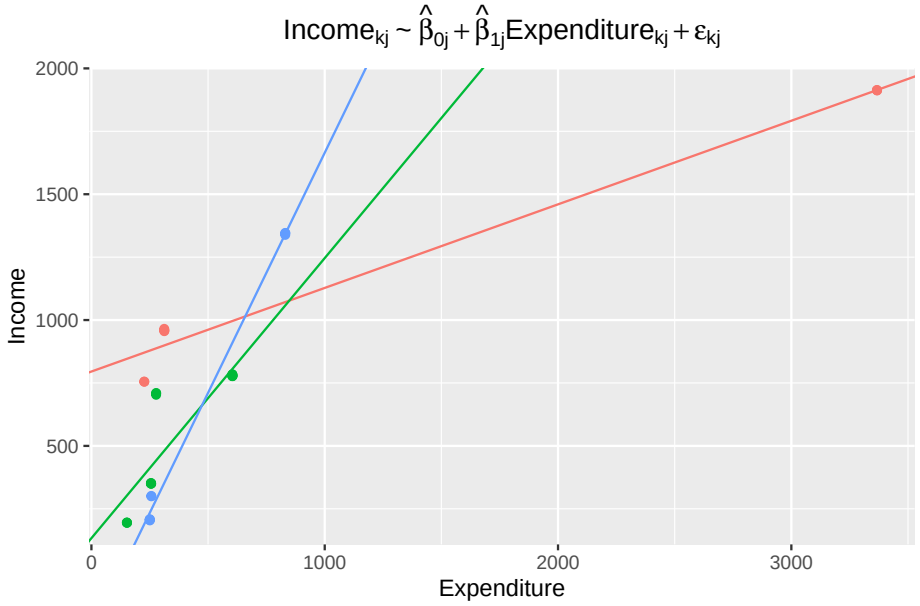

Figure 7.4: Model with stratum-specific intercept and slope (independent fit)

Although this last specification offers a more flexible representation of the data and allows the particular features of each stratum to be captured, it presents an important limitation: the parameters are estimated completely independently for each group. As a consequence, it does not take advantage of information shared across strata that may exhibit similar patterns, which can generate unstable estimates, especially in those groups with a small number of observations.

Multilevel models overcome this difficulty through a mechanism known as shrinkage, in which specific estimates for each group are obtained by combining the group's own information with information from the population as a whole. In this way, a balance is achieved between the representation of the differences between strata and the statistical efficiency of the estimates.

II *q-weighted* Weights

Regarding the incorporation of sampling weights in multilevel models, although there is a general consensus on the need to use them to guarantee inferences that are representative of the target population [Cai, 2013], there is no single methodological strategy for their implementation. In this context, Pfeffermann et al. [1998] and Asparouhov [2006] propose approaches based on pseudo-maximum likelihood, particularly through weighted generalized least squares procedures. Alternatively, Rabe-Hesketh and Skrondal [2006] propose methods based on EM algorithms for the estimation of hierarchical models with weights.

From a general perspective, the maximum likelihood approach consists of estimating the parameters of the model by identifying those values that maximize the probability of observing the available data under the assumed specification. An important extension of this approach is restricted maximum likelihood (REML), which improves the estimation of variance components by correcting the bias associated with the estimation of degrees of freedom, producing in many cases more stable estimates than conventional maximum likelihood [Kreft and De Leeuw, 1998].

In the context of multilevel models with survey data, an additional difficulty lies in the coherent incorporation of sampling weights across the different levels of the hierarchy. As Pfeffermann et al. [1998] point out, the clustered structure of the data implies that the observations are not independent, so the log-likelihood function cannot be decomposed as a simple sum of individual contributions. Instead, it is necessary to explicitly consider the dependence between the different levels of the sample design in order to obtain appropriate inferences.

To fit a multilevel model with data from complex surveys, it is recommended to redefine the expansion factors using the *q-weighted* weights approach proposed by Pfeffermann [2011]. This procedure seeks to eliminate the systematic part of the weights associated with the covariates included in the model, preserving only the residual component of the selection mechanisms. The steps are as follows:

1. Fit a regression model for the final expansion factors of the survey, using as explanatory variables the same set of covariates that will be used in the multilevel model. Let w_k be the original expansion factor of unit k and $\mathbf{x}_k$ the vector of covariates. Then the following model is fitted

$$w_k = f(\mathbf{x}_k, \beta) + \varepsilon_k,$$

   where $f(\cdot)$ represents the functional relationship between the weights w_k and the covariates $\mathbf{x}_k$, through the coefficients β.

2. Obtain the predicted values from the model for each observation unit, such that:

$$\hat{w}_k = f(\mathbf{x}_k, \hat{\beta}).$$

   These values represent the component of the expansion factors explained by the covariates included in the model.

3. Construct the *q-weighted* weights by dividing the original expansion factors by their corresponding predicted values,

$$q_k = \frac{w_k}{\hat{w}_k}.$$

   In this way, the new weights reflect only the residual variation of the expansion factors after controlling for the effect of the covariates.

4. Define the new sample design using the adjusted weights q_k instead of the original weights w_k, and use this design to estimate the parameters of the multilevel model.

The following code implements the *q-weighted* weight construction procedure. First, it fits a linear regression model that explains the original expansion factors based on the explanatory variable expenditure. In this way, the resulting weights retain only the component of variation not explained by the covariates included in the multilevel model.

```
mod_qw <- lm(wk ~ Expenditure, data = survey_data)
survey_data$qk <- survey_data$wk / predict(mod_qw)
```

III Linear Multilevel Model

The multilevel models most commonly used in practice are extensions of the classical linear model for data with a hierarchical structure. In their basic formulation, these models assume that the response variable is continuous and follows a normal distribution conditional on the fixed and random effects included in the model. It is also assumed that the random effects and residual errors are independent of each other and normally distributed with zero mean and constant variances.

A Null Model

In multilevel regression analysis, two types of parameters are distinguished: regression coefficients, known as fixed parameters, and variance components associated with random effects. A fundamental initial stage in this type of analysis consists of decomposing the total variability of the dependent variable into the different levels of the hierarchical structure. In the context of the previous example, this decomposition makes it possible to separate income variation into a component attributable to differences within strata and another associated with differences between strata.

The starting point for this decomposition is the so-called null model, which does not incorporate explanatory variables and is expressed as

$$y_{kj} = \beta_{0j} + \epsilon_{kj}$$

where y_{kj} denotes the observed value of income for unit k in stratum j. The term β_{0j} represents the stratum-specific intercept, capturing the differences in the average levels of the variable between

groups. Finally, ϵ_{kj} corresponds to the error at the unit level, which captures the unexplained variability within each stratum and is assumed to have zero mean and constant variance conditional on the group.

In turn, the stratum intercept can be decomposed into a part common to all strata and a specific deviation for each of them, as follows:

$$\beta_{0j} = \gamma_{00} + \tau_{0j}$$

where γ_{00} represents the global average intercept and τ_{0j} is the random effect for the intercept that captures the deviation of stratum j, allowing the heterogeneity between strata to be explicitly modeled. Furthermore, the random components of the model are assumed normally distributed with zero mean and specific variances for each level of the hierarchy. In particular, the random effect associated with the intercept by stratum satisfies

$$\tau_{0j} \sim N\left(0, \sigma_\tau^2\right),$$

which implies that the deviations of each stratum from the global intercept are distributed around zero, with a variability determined by σ_τ^2. Analogously, the unit-level error term follows the distribution

$$\epsilon_{kj} \sim N\left(0, \sigma_\epsilon^2\right),$$

where σ_ϵ^2 represents the residual variability within the strata. From this model, the intraclass correlation coefficient (ICC) is defined, which quantifies the proportion of the total variance explained by the differences between strata:

$$\rho = \frac{\sigma_\tau^2}{\sigma_\tau^2 + \sigma_\epsilon^2}$$

A high ICC value indicates that a significant proportion of the total variability of the variable of interest is due to differences between strata, which suggests the presence of relevant heterogeneity between groups and the need to explicitly incorporate that structure in the model. In contrast, a low ICC implies that the variability is mainly concentrated within the strata, which show greater relative homogeneity among themselves.

Continuing with the analysis of the example survey, when the variable of interest is income, the null model is the starting point for quantifying what proportion of the total variability is due to differences between strata and what proportion corresponds to differences between households within the same stratum.

Estimation of multilevel models in `R` is performed using the `lme4` [Bates et al., 2026] package. In particular, the `lmer()` function makes it possible to specify random effects using expressions of the form `(effect | group)`. In this context, the expression `(1 | Stratum)` indicates that the model includes a random intercept for each stratum. The value `1` represents the model intercept, while `Stratum` identifies the grouping variable.

```
mod_null <- lmer(
  Income ~ (1 | Stratum),
  data = survey_data,
  weights = qk
)
```

Table 7.1 presents the estimated intercepts for each stratum from the null model. Since this model does not incorporate explanatory variables, each intercept can be interpreted as the expected average income in the corresponding stratum. The results show substantial differences between groups: while the `idStrt004` stratum presents the highest estimated average income (959.6), the `idStrt009` stratum registers the lowest value (207.6). These differences suggest the existence of substantial variability between strata, justifying the incorporation of random effects to explicitly model the hierarchical structure of the data.

```
coef_mod_null <- coef(mod_null)$Stratum
coef_mod_null %>%
  slice(1:12L)
```

Table 7.1: Estimated intercepts by stratum under q-weighted and senate-weighted weights (null model, first 12 strata)

	(Intercept)
idStrt001	635
idStrt002	507
idStrt003	486
idStrt004	960
idStrt005	518
idStrt006	439
idStrt007	477
idStrt008	377
idStrt009	218
idStrt010	594
idStrt011	590
idStrt012	362

Based on the variance components estimated in the null model, the intraclass correlation is calculated using the `icc()` function from the `performance` [Lüdecke et al., 2026] package. This indicator quantifies the proportion of total income variability that can be attributed to differences between strata, constituting a measure of the dependence between observations belonging to the same group. The results obtained are presented in Table 7.2.

```
performance::icc(mod_null) %>%
  as.data.frame()
```

Table 7.2: Intraclass correlation of the null model (income by stratum)

ICC_adjusted	ICC_unadjusted	optional
0.329	0.329	FALSE

The estimated intraclass correlation of 32% indicates that approximately one third of the total observed variability in income is due to differences between strata, while the remaining percentage corresponds to differences between households within the same strata. Finally, since the null model does not include any predictors, the estimate of income within each stratum is constant and equal to the estimated intercept for that stratum.

B Random Intercept Model

The simplest multilevel model that incorporates covariates is the random intercept model. In this specification, it is assumed that the effects of the explanatory variables are common to all strata, while the average level of the response variable may vary between them. The model is expressed as

$$y_{kj} = \beta_{0j} + \mathbf{x}_{kj}\beta + \epsilon_{kj}$$

where y_{kj} represents the observed value of the response variable for unit k belonging to stratum j, $\mathbf{x}_{kj}$ is the vector of covariates associated with that unit, β is the vector of regression coefficients common to all strata and ϵ_{kj} corresponds to the random error at the unit level.

As in the null model, the intercept is modeled as $\beta_{0j} = \gamma_{00} + \tau_{0j}$; where γ_{00} represents the global average intercept and τ_{0j} is the random effect associated with stratum j, which measures the deviation of that stratum from the overall average.

The random components of the model are assumed to be independent and normally distributed, so that $\tau_{0j} \sim N(0, \sigma^2_\tau)$ and $\epsilon_{kj} \sim N(0, \sigma^2_\epsilon)$. Under these conditions, the total variability of the response variable can be decomposed into a component between strata, quantified by σ^2_τ, and a component within the strata, represented by σ^2_ϵ.

The following code fits a multilevel random intercept model, where income (`Income`) is explained by household expenditure (`Expenditure`), also incorporating a random effect associated with the stratum (`Stratum`) and using the *q-weighted* weights stored in the variable `qk`. Once the model is fitted, the intraclass correlation (ICC) is estimated from the estimated variance components.

```
random_intercept_model <- lmer(
  Income ~ Expenditure + (1 | Stratum),
  data = survey_data,
  weights = qk
)
performance::icc(random_intercept_model)
```

```
## # Intraclass Correlation Coefficient
##
##     Adjusted ICC: 0.203
##   Unadjusted ICC: 0.109
```

The adjusted intraclass correlation of 0.196 indicates that, after controlling for household expenditure (`Expenditure`), approximately 20% of the residual variability in income remains attributable to differences between strata. Although this value is lower than that obtained with the null model, the magnitude of the ICC continues to justify the use of a multilevel model to adequately represent the hierarchical structure of the data. The estimated coefficients by stratum are presented in Table 7.3:

```
coef(random_intercept_model)$Stratum %>%
  slice(1:8L)
```

Table 7.3: Coefficients of the model with random intercept by stratum (first 8 strata)

	(Intercept)	Expenditure
idStrt001	250.4	1.19
idStrt002	156.8	1.19
idStrt003	142.9	1.19
idStrt004	296.8	1.19
idStrt005	-32.4	1.19
idStrt006	53.8	1.19
idStrt007	12.5	1.19
idStrt008	107.1	1.19

To facilitate the visualization of the model results, the reduced subsample presented above is used, which contains only three selected strata.

```
estimated_coef <- inner_join(
  coef(random_intercept_model)$Stratum %>%
    tibble::rownames_to_column(var = "Stratum"),
  plot_data %>%
    select(Stratum) %>%
    distinct()
)
```

Figure 7.5 shows the estimated regression lines for each stratum under the random intercept model. The lines share the same slope, reflecting the common effect of expenditure on income, but they differ in their vertical position due to the specific intercepts estimated for each stratum.

```
ggplot(data = plot_data,
       aes(y = Income, x = Expenditure, colour = Stratum)) +
  geom_jitter() +
  geom_abline(data = estimated_coef,
              mapping = aes(slope = Expenditure,
                            intercept = `(Intercept)`,
                            colour = Stratum)) +
  theme(
    legend.position = "none",
    plot.title = element_text(hjust = 0.5)
  )
```

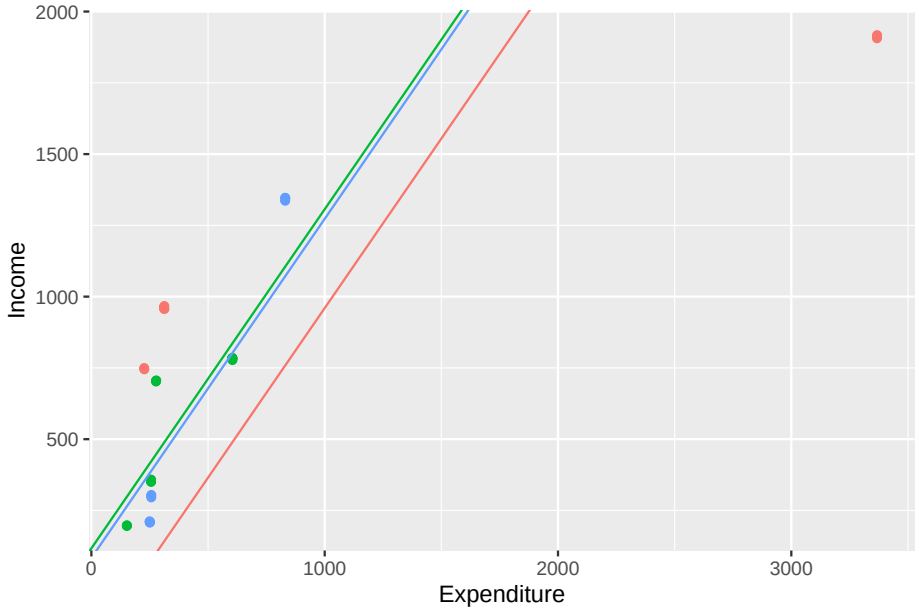

Figure 7.5: Estimated regression lines by stratum: model with random intercept

C Random Intercept and Slope Model

A natural extension of the previous model is the random intercept and slope model. In this specification, not only is the average level of the response variable allowed to vary between strata, but the effect of one or more covariates is also allowed to differ between groups. In this way, each stratum can have its own regression line, both in terms of intercept and slope. The model can be expressed as

$$y_{kj} = \beta_{0j} + x_{kj}\beta_{1j} + \mathbf{z}_{kj}\beta + \epsilon_{kj},$$

where y_{kj} represents the observed value of the response variable for unit k belonging to stratum j, x_{kj} corresponds to the covariate whose slope is allowed to vary between strata, $\mathbf{z}_{kj}$ contains the covariates with fixed effects common to all groups, and ϵ_{kj} is the error term at the unit level. In

this model, both the intercept and the slope associated with x_{kj} are considered random and are decomposed as follows:

$$\beta_{0j} = \gamma_{00} + \tau_{0j}, \qquad \beta_{1j} = \gamma_{10} + \tau_{1j}$$

where γ_{00} and γ_{10} represent, respectively, the average intercept and slope in the population, while τ_{0j} and τ_{1j} capture the specific deviations of stratum j from those averages. These random effects are assumed normally distributed as follows:

$$\begin{pmatrix} \tau_{0j} \\ \tau_{1j} \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2_{\tau_0} & \sigma_{\tau_{01}} \\ \sigma_{\tau_{01}} & \sigma^2_{\tau_1} \end{pmatrix} \right),$$

while the unit-level error satisfies $\epsilon_{kj} \sim N(0, \sigma^2_\epsilon)$. Consequently, the model makes it possible to quantify not only the variability between strata in the average levels of the response variable, but also the existing heterogeneity in the effect of the covariate whose coefficient is modeled as random.

The following code fits a multilevel model with random intercept and slope, in which income (`Income`) is explained by household expenditure (`Expenditure`) and area of residence (`Zone`). In this case, both the intercept and the effect of expenditure may vary between strata through random effects associated with `Stratum`, also using the *q-weighted* weights stored in the variable `qk`.

```
random_slope_model <- lmer(
  Income ~ 1 + Expenditure + (1 + Expenditure | Stratum),
  data = survey_data,
  weights = qk
)

performance::icc(random_slope_model)
```

```
## # Intraclass Correlation Coefficient
##
##     Adjusted ICC: 0.690
##   Unadjusted ICC: 0.457
```

Note that the code specification simultaneously includes fixed effects and random effects for the intercept and the variable `Expenditure`. The terms outside the parentheses, `1 + Expenditure`, correspond to the fixed effects of the model and make it possible to estimate the global average intercept and the global average slope associated with expenditure. For its part, the expression `(1 + Expenditure | Stratum)` incorporates the specific deviations of each stratum from those averages. The explicit inclusion of fixed effects is fundamental, since random effects are interpreted as deviations around a population mean. The model coefficients by stratum are presented in Table 7.4:

```
coef(random_slope_model)$Stratum %>%
  slice(1:10L)
```

Table 7.4: Coefficients of the model with random intercept and slope by stratum (first 10 strata)

	(Intercept)	Expenditure
idStrt001	-222.8	2.730
idStrt002	35.5	1.607
idStrt003	151.9	1.164
idStrt004	224.2	1.353
idStrt005	-89.7	1.286
idStrt006	28.6	1.217
idStrt007	41.0	1.079
idStrt008	163.8	0.928
idStrt009	16.6	0.838
idStrt010	89.9	1.830

Once again, in order to facilitate the visualization of the results, the reduced subsample defined previously is used, which includes only three selected strata.

```
estimated_coef <- inner_join(
  coef(random_slope_model)$Stratum %>%
    tibble::rownames_to_column(var = "Stratum"),
  plot_data %>%
    select(Stratum) %>%
    distinct()
)
```

Figure 7.6 presents the estimated regression lines for each stratum under the random intercept and slope model. Unlike the random intercept model, in this case the lines differ both in their vertical position and in their slope, reflecting that strata not only have different average income levels, but also different strengths in the relationship between income and expenditure.

```
ggplot(data = plot_data,
       aes(y = Income, x = Expenditure, colour = Stratum)) +
  geom_jitter() +
  geom_abline(data = estimated_coef,
              mapping = aes(slope = Expenditure,
                            intercept = `(Intercept)`,
                            colour = Stratum)) +
  theme(
    legend.position = "none",
    plot.title = element_text(hjust = 0.5)
  )
```

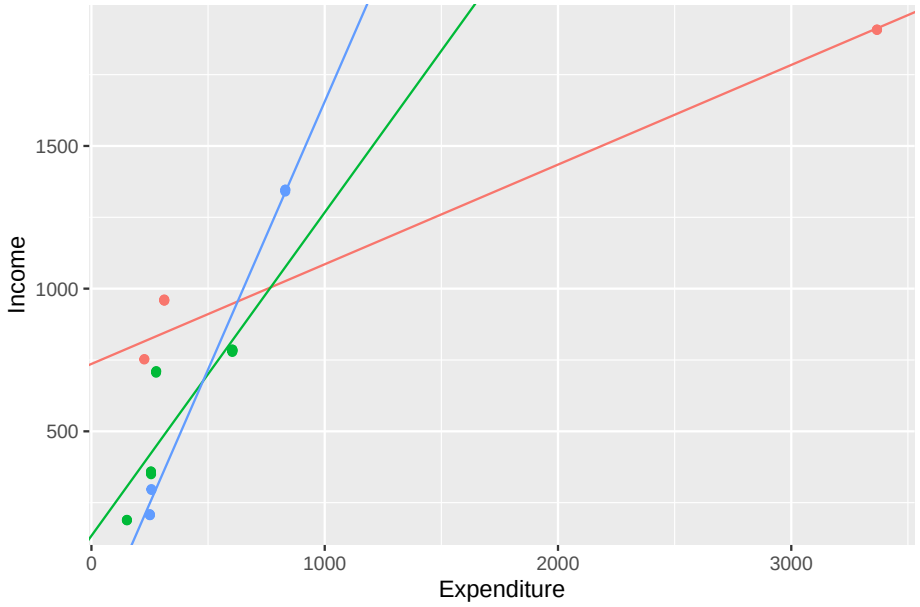

Figure 7.6: Estimated regression lines by stratum: model with random intercept and slope

IV Multilevel Logistic Model

Multilevel logistic models extend multilevel models for continuous variables to the case in which the response variable is dichotomous. Instead of directly modeling the expected value of a continuous variable, these models estimate the probability of occurrence of an event, simultaneously incorporating individual covariates and random effects associated with the groups to which units belong. In the context of household surveys, this formulation is especially useful when analyzing binary outcomes, such as being in poverty or not, accessing a service or not, participating or not in the labor market, or presenting a certain sociodemographic characteristic.

As in multilevel models for continuous variables, the starting point is to recognize that the observation units are not independent when they belong to the same stratum, cluster, or domain. However, in the logistic case the response is not represented by a normal distribution with an additive residual error, but by a Bernoulli distribution conditional on the probability of occurrence of the event. This probability is linked to the predictors through the logit function, which makes it possible to express the model on a linear scale.

$$y_{kj} \mid \pi_{kj} \sim \text{Bernoulli}(\pi_{kj})$$

where y_{kj} takes the value one if unit k of stratum j presents the event of interest and zero otherwise. The conditional probability of the event is denoted by $\pi_{kj} = Pr(y_{kj} = 1)$ and is related to the linear predictor through

$$\text{logit}(\pi_{kj}) = \log\left(\frac{\pi_{kj}}{1-\pi_{kj}}\right) = \eta_{kj}$$

In this formulation, η_{kj} plays a role analogous to the linear expected value of models for continuous variables. The central difference is that the fixed and random effects act on the logit scale and

not directly on the probability. Therefore, the differences between strata are interpreted as shifts in the *log-odds* of the event, which are then transformed into probabilities through the logistic function. Figure 7.7 presents the relationship between expenditure and the probability of poverty, with a fitted logistic curve.

```
ggplot(data = survey_data,
       aes(y = poverty, x = Expenditure)) +
  geom_point(alpha = 0.5) +
  geom_smooth(
    formula = y ~ x,
    method = "glm",
    se = FALSE,
    method.args = list(family = binomial(link = "logit"))
  ) +
  labs(y = "Poverty (1 = poor)", x = "Expenditure")
```

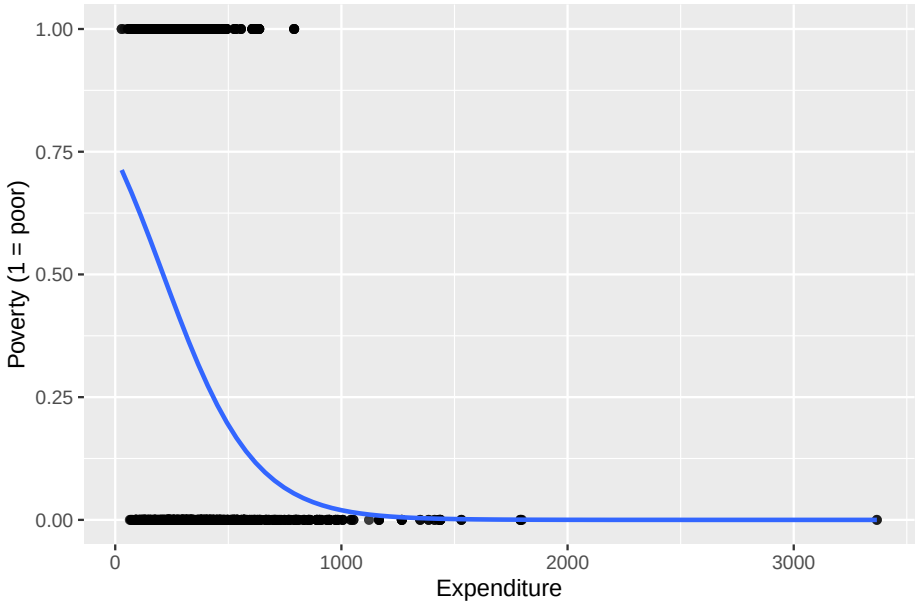

Figure 7.7: Relationship between expenditure and poverty status with fitted logistic curve

The fitted curve summarizes the marginal association between expenditure and poverty status, without yet incorporating the hierarchical structure of the strata. In general terms, the descending shape of the curve indicates that, as household expenditure increases, the estimated probability of being in poverty decreases. However, this average relationship can hide relevant differences between strata, especially when households belong to different socioeconomic contexts.

A Logistic Null Model

As in the case of continuous variables, the logistic null model constitutes the starting point to study the hierarchical structure of the response variable. This model does not incorporate covariates

and allows us to evaluate whether the average probability of the event varies between strata. In the example considered, the event of interest corresponds to the poverty status, so that the model allows us to separate the heterogeneity attributable to differences between strata from the individual variation inherent to a binary response.

In the null model, the variable of interest follows the following Bernoulli distribution:

$$y_{kj} \mid \pi_{kj} \sim \text{Bernoulli}(\pi_{kj})$$

where the probability of success for unit k belonging to group j is modeled through

$$\text{logit}(\pi_{kj}) = \beta_{0j},$$

where β_{0j} represents the specific intercept of stratum j on a logit scale. In turn, this intercept is decomposed as

$$\beta_{0j} = \gamma_{00} + \tau_{0j}$$

where γ_{00} is the global average intercept of the population and τ_{0j} the random effect associated with stratum j, which captures the deviations of each group from the overall average. As in the null model for continuous variables, this random effect allows heterogeneity between groups to be explicitly modeled:

$$\tau_{0j} \sim N(0, \sigma^2_\tau)$$

Unlike the multilevel linear model, here an additive residual term ϵ_{kj} is not incorporated into the individual-level equation. Within-group variability is determined by the Bernoulli distribution of the response. To quantify the dependence between units belonging to the same stratum, the latent variable approach is often used, under which this residual variance is approximated by the variance of the standard logistic distribution, that is, $\pi^2/3$ [Snijders and Bosker, 2011]. Thus, the intraclass correlation of the logistic model is expressed as

$$\rho = \frac{\sigma^2_\tau}{\sigma^2_\tau + \frac{\pi^2}{3}}$$

A high value of ρ indicates that the probability of the event has an important clustering structure, that is, that two units belonging to the same stratum tend to be more similar to each other than two units taken from different strata. In contrast, a low value suggests that most of the variation is concentrated at the individual level. The fit in `R` is carried out as follows:

```
logistic_null_model <- glmer(
  poverty ~ (1 | Stratum),
  data = survey_data,
  weights = qk,
  family = binomial(link = "logit")
)
```

The coefficients of the logistic null model by stratum are presented in Table 7.5. The intercepts estimated in the null model show considerable heterogeneity between strata. In the first reported strata, some intercepts are strongly negative, such as `idStrt004` and `idStrt003`, which corresponds to very low baseline probabilities of poverty. In contrast, strata such as `idStrt009` and `idStrt006` present positive and high intercepts, associated with much higher baseline probabilities.

```
coef(logistic_null_model)$Stratum %>%
  slice(1:12L)
```

Table 7.5: Estimated intercepts by stratum in the logistic null model (first 12 strata)

	(Intercept)
idStrt001	-0.852
idStrt002	-0.038
idStrt003	-2.378
idStrt004	-2.618
idStrt005	-1.037
idStrt006	0.902
idStrt007	-1.018
idStrt008	0.156
idStrt009	1.913
idStrt010	-0.609
idStrt011	-1.275
idStrt012	0.203

The intraclass correlation amounts to 0.315. Since the model does not include covariates, these differences exclusively reflect between-strata variation in the prevalence of the event.

```
performance::icc(logistic_null_model)
```

```
## # Intraclass Correlation Coefficient
##
##     Adjusted ICC: 0.315
##   Unadjusted ICC: 0.315
```

B Logistic Model with Random Intercept

The logistic model with a random intercept incorporates individual or household covariates, allowing the baseline level of the event to vary between strata. Its logic is parallel to that of the model with a random intercept for continuous variables: the effects of the covariates are considered common to all groups, while each stratum can have its own intercept. In this case, the differences between intercepts are interpreted as differences in the baseline *log-odds* of the event.

The model is expressed as $y_{kj} \mid \pi_{kj} \sim \text{Bernoulli}(\pi_{kj})$, where $\text{logit}(\pi_{kj}) = \beta_{0j} + \mathbf{x}_{kj}\beta$ and $\beta_{0j} = \gamma_{00} + \tau_{0j}$. In this case, $\mathbf{x}_{kj}$ represents the covariate vector of unit k in stratum j, β is the vector of fixed effects common to all strata, and τ_{0j} captures the stratum-specific deviation from the global average intercept. As before, it is assumed that $\tau_{0j} \sim N(0, \sigma_\tau^2)$.

Under this specification, two strata with the same values of the covariates can present different baseline probabilities of the event due to their random intercepts. However, the effect of each covariate on the logit scale remains constant between strata. In the example, this is equivalent to allowing the baseline probability of poverty to change across strata, while the association between expenditure and poverty is summarized by a common slope.

```
random_intercept_logit_model <- glmer(
  poverty ~ Expenditure + (1 | Stratum),
  data = survey_data,
  family = binomial(link = "logit"),
  weights = qk
)
performance::icc(random_intercept_logit_model)
```

```
## # Intraclass Correlation Coefficient
##
##     Adjusted ICC: 0.298
##   Unadjusted ICC: 0.171
```

The estimated coefficients by stratum are shown in Table 7.6. When incorporating household expenditure as a covariate, the fixed coefficient associated with `Expenditure` is negative, indicating that higher levels of expenditure are associated with lower *log-odds* of poverty. This implies a progressive reduction in the estimated probability of poverty as expenditure increases, holding the stratum effect constant. In this model, the effect of expenditure is common, but each stratum has its own baseline level of poverty. Thus, strata with high positive intercepts have a higher baseline probability of poverty for the same level of expenditure, while strata with negative intercepts show a lower baseline probability.

Even after controlling for expenditure, the adjusted intraclass correlation remains high, around 0.298, showing that differences between strata remain relevant.

```
coef(random_intercept_logit_model)$Stratum %>%
  slice(1:10L)
```

Table 7.6: Coefficients of the logistic model with random intercept by stratum (first 10 strata)

	(Intercept)	Expenditure
idStrt001	1.044	-0.007
idStrt002	1.918	-0.007

	(Intercept)	Expenditure
idStrt003	-0.454	-0.007
idStrt004	0.062	-0.007
idStrt005	1.760	-0.007
idStrt006	3.194	-0.007
idStrt007	0.658	-0.007
idStrt008	1.711	-0.007
idStrt009	3.721	-0.007
idStrt010	1.175	-0.007

Figure 7.8 presents the probability curves predicted by a logistic model with a random intercept by stratum. An inverse relationship is observed between expenditure and the probability of poverty, such that households with higher expenditure levels have a lower probability of being classified as poor. Likewise, the differences between the curves reflect the existing heterogeneity between strata, captured by the random intercepts of the model.

```
prediction_data <- plot_data %>%
  group_by(Stratum) %>%
  summarise(
    Expenditure = list(seq(min(Expenditure), max(Expenditure), len = 100))
  ) %>%
  tidyr::unnest_legacy()

prediction_data <- prediction_data %>%
  mutate(
    probability = predict(
      random_intercept_logit_model,
      newdata = prediction_data,
      type = "response"
    )
  )
```

```
ggplot(data = prediction_data,
       aes(y = probability, x = Expenditure, colour = Stratum)) +
  geom_line() +
  geom_point(data = plot_data,
             aes(y = poverty, x = Expenditure)) +
  labs(y = "Poverty probability", x = "Expenditure") +
  theme(legend.position = "none")
```

C Logistic Model with Random Intercept and Slope

The logistic model with random intercept and slope extends the previous specification by allowing not only the baseline level of the event to vary between strata, but also the effect of a specific

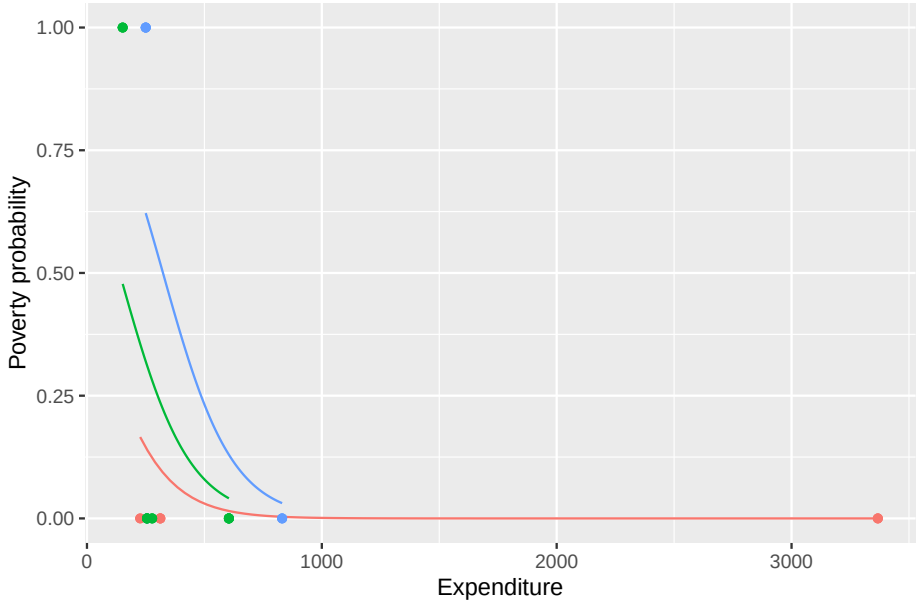

Figure 7.8: Predicted probability curves by stratum: logistic model with random intercept

covariate. This formulation is analogous to the random intercept and slope model for continuous variables, with the exception that the differences between strata are expressed on the logit scale and are translated into nonlinear probability curves.

Let x_{kj} be the covariate whose effect is allowed to vary between strata, and let $\mathbf{z}_{kj}$ be the set of covariates with common fixed effects. In this model, the probability of success is assumed such that $\text{logit}(\pi_{kj}) = \beta_{0j} + x_{kj}\beta_{1j} + \mathbf{z}_{kj}\beta$. Here, β_{0j} is the stratum-specific intercept for stratum j and β_{1j} is the specific slope associated with the covariate x_{kj}.

Both parameters are decomposed into a population-average part and a stratum-specific deviation, such that $\beta_{0j} = \gamma_{00} + \tau_{0j}$ and $\beta_{1j} = \gamma_{10} + \tau_{1j}$. The terms γ_{00} and γ_{10} represent, respectively, the global average intercept and the global average slope on a logit scale. For their part, τ_{0j} and τ_{1j} capture how much stratum j deviates from these averages. The implementation in R is as follows:

```
random_slope_logit_model <- glmer(
  poverty ~ 1 + Expenditure + (1 + Expenditure | Stratum),
  data = survey_data,
  weights = qk,
  family = binomial(link = "logit")
)
performance::icc(random_slope_logit_model)
```

```
## # Intraclass Correlation Coefficient
##
##     Adjusted ICC: 0.875
##   Unadjusted ICC: 0.631
```

The estimation of the model with random intercept and slope indicates much more pronounced heterogeneity between strata. In this case, both the intercepts and slopes associated with

expenditure can change between groups. The adjusted intraclass correlation is very high, suggesting that the between-strata component dominates the latent variation in the model. The model coefficients by stratum are shown in Table 7.7.

```
coef(random_slope_logit_model)$Stratum %>%
  slice(1:10L)
```

Table 7.7: Coefficients of the logistic model with random intercept and slope by stratum (first 10 strata)

	(Intercept)	Expenditure
idStrt001	4.865	-0.025
idStrt002	9.862	-0.035
idStrt003	-1.133	-0.007
idStrt004	1.899	-0.015
idStrt005	8.030	-0.026
idStrt006	-1.153	0.009
idStrt007	0.974	-0.012
idStrt008	1.488	-0.006
idStrt009	3.660	-0.005
idStrt010	4.133	-0.020

The coefficients by stratum show that the relationship between expenditure and poverty changes not only in its baseline level, but also in the strength and direction of the slope. In several strata the slope of expenditure is negative, which maintains the expected interpretation that higher levels of expenditure reduce the probability of poverty. However, slopes close to zero and even positive slopes also appear in some strata, suggesting distinct local patterns. This variability is precisely what the model seeks to capture through the random effect of the slope. Figure 7.9 shows that the predicted curves are more heterogeneous between strata than in the previous model:

```
prediction_data <- plot_data %>%
  group_by(Stratum) %>%
  summarise(
    Expenditure = list(seq(min(Expenditure), max(Expenditure), len = 100))
  ) %>%
  tidyr::unnest_legacy()

prediction_data <- prediction_data %>%
  mutate(
    probability = predict(
      random_slope_logit_model,
      newdata = prediction_data,
      type = "response"
    )
  )
```

```
ggplot(data = prediction_data,
       aes(y = probability, x = Expenditure, colour = Stratum)) +
  geom_line() +
  geom_point(data = plot_data,
             aes(y = poverty, x = Expenditure)) +
  labs(y = "Poverty probability", x = "Expenditure") +
  theme(legend.position = "none")
```

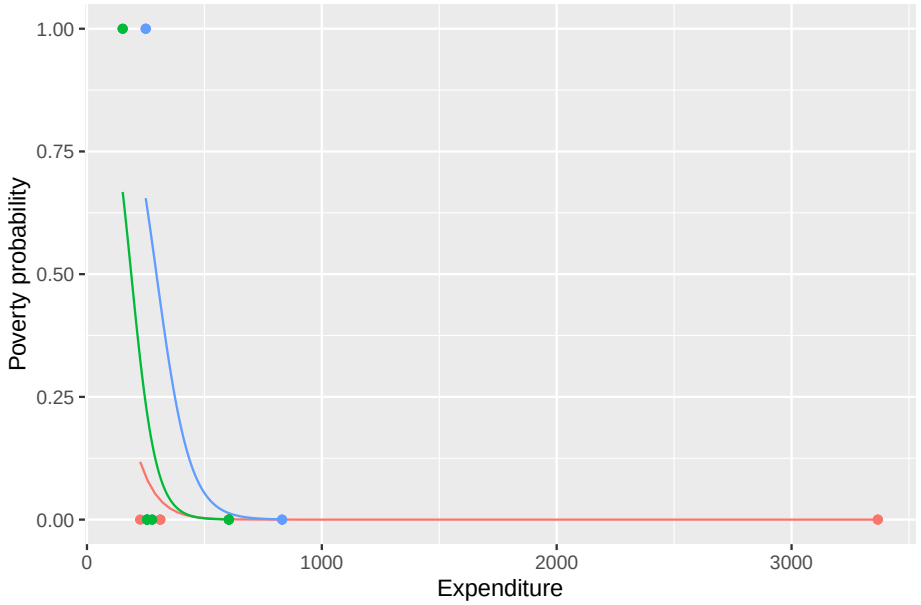

Figure 7.9: Predicted probability curves by stratum: logistic model with random intercept and slope

Appendix A

Data Wrangling with R

R is statistical software widely used for data analysis and graph generation, and it provides an integrated environment for statistical analysis, particularly for databases. This means that analysis, programming, visualization, and data management processes can be carried out within the same environment. This appendix uses the `tidyverse` approach, and in particular the `dplyr` package, to organize a logical workflow for working with databases: loading the data, inspecting it, selecting variables, filtering records, sorting rows, creating variables, and producing summaries.

I Preparing the Working Environment

R is a collaborative programming language, which allows its community of users and developers to contribute continuously by creating specialized functions, packages, and libraries. Thanks to this open ecosystem, it is possible to expand its capabilities for different types of statistical analysis and data processing. The following are among the libraries most commonly used for database analysis:

- The `tidyverse` library [Wickham et al., 2026a] is a set of R packages designed to support different stages of data work, from import and transformation to exploration, analysis, and visualization. Its use has become widespread in data science because it offers a consistent and orderly way of working.
- The `dplyr` package [Wickham et al., 2026c] provides tools for organizing and transforming rectangular databases. Its usefulness lies in bringing together, in a simple and consistent syntax, the most frequent operations in analytical work with data: selecting variables, filtering records, creating new columns, sorting observations, and summarizing information.
- The `TeachingSampling` package [Gutiérrez, 2020] provides functions for selecting probability samples and carrying out inference on finite populations under various sampling designs.

Before using the different functions provided by each library, they must first be downloaded from the web. The `install.packages` command performs this task. Note that some libraries may depend on others, so additional dependencies must also be installed in order to use them.

```
install.packages("dplyr")
install.packages("tidyverse")
install.packages("TeachingSampling")
```

Once the packages have been installed, they are loaded with `library()`. The first command clears the existing objects in the working session, which helps ensure that the chapter results depend only on the code executed here. Keep in mind that a package can only be loaded if it has previously been installed on the system.

```
rm(list = ls())

library(tidyverse)
library(dplyr)
library(TeachingSampling)
```

After the libraries or packages are downloaded and installed in R, the recommended next step is to carry out all processing through the creation of projects. An R project is defined as a file that contains the source files and content associated with the work being done. In addition, it contains information that allows each R file to be compiled, maintains information for integration with source code control systems, and helps organize processing into logical components.

II Loading and Initial Inspection of the Database

When working on projects in R, it is very common to import databases containing information relevant to a particular study. R can import databases in several formats, including `xlsx`, `csv`, `txt`, `STATA`, and others. Once the database has been read in the relevant format, it is advisable to transform it into the native R format, that is, `.RDS`. This is a more efficient format specific to R.

Once the database has been loaded, R functions are used to obtain results from aggregate processing and graphs of interest. To illustrate the use of functions that produce aggregate results, we will use the `BigCity` database from the `TeachingSampling` package. This database contains a set of socioeconomic variables for 150266 people in a particular year.

```
data(BigCity)
bigcity_data <- BigCity
```

The `bigcity_data` object contains a copy of `BigCity`. The `head()` function makes it possible to view the first rows and obtain a quick impression of the database structure.

```
head(bigcity_data)
```

```
##        HHID PersonID   Stratum     PSU  Zone    Sex Age MaritalST Income
## 1 idHH00001  idPer01 idStrt001 PSU0001 Rural   Male  38   Married    555
## 2 idHH00001  idPer02 idStrt001 PSU0001 Rural Female  40   Married    555
```

```
## 3 idHH00001  idPer03 idStrt001 PSU0001 Rural Female  20    Single    555
## 4 idHH00001  idPer04 idStrt001 PSU0001 Rural   Male  19    Single    555
## 5 idHH00001  idPer05 idStrt001 PSU0001 Rural   Male  18    Single    555
## 6 idHH00002  idPer01 idStrt001 PSU0001 Rural   Male  35   Married    298
##   Expenditure Employment  Poverty
## 1         488   Employed  NotPoor
## 2         488   Employed  NotPoor
## 3         488   Inactive  NotPoor
## 4         488   Employed  NotPoor
## 5         488   Inactive  NotPoor
## 6         217   Employed Relative
```

Once the database has been loaded into `R`, processing can begin according to the needs of each researcher. In this sense, one of the first checks performed when loading a database is to verify its dimensions; that is, to identify the number of rows and columns it contains. This can be done with the `nrow` function, which identifies the number of records in the database, and with the `ncol` function, which shows the number of variables. The computational code is as follows:

```
nrow(bigcity_data)
```

```
## [1] 150266
```

```
ncol(bigcity_data)
```

```
## [1] 12
```

A summarized way to check the number of rows and columns in a database is through the `dim` function, which returns a vector whose first component corresponds to the number of rows and whose second component indicates the number of columns.

```
dim(bigcity_data)
```

```
## [1] 150266     12
```

In some cases, databases may be large, either because they contain a considerable number of observed variables or because the number of records is large. For this reason, to view them properly, it may be convenient to open them in a separate window and explore their content in tabular form. This is done using the `View` function, as shown below.

```
View(bigcity_data)
```

Another important check when loading a database into `R` is to identify the variables it contains. For this purpose, the `names` function can be used; it shows the names of the variables included in the database.

```
names(bigcity_data)
```

```
##  [1] "HHID"        "PersonID"    "Stratum"     "PSU"         "Zone"
##  [6] "Sex"         "Age"         "MaritalST"   "Income"      "Expenditure"
## [11] "Employment"  "Poverty"
```

The previous output shows the names of the variables contained in the database. In this case, the database has 12 variables. The first variables correspond to identifiers and sampling design elements, such as `HHID`, `PersonID`, `Stratum`, `PSU`, and `Zone`. Then come sociodemographic variables, such as `Sex`, `Age`, and `MaritalST`. Finally, economic variables and variables related to labor or social status are included, such as `Income`, `Expenditure`, `Employment`, and `Poverty`.

To examine in greater detail the type of information stored by each variable, the `str` function can be used. This function compactly displays the structure of an object and its components. For our database, this function would be used as follows:

```
str(bigcity_data)
```

```
## 'data.frame':    150266 obs. of  12 variables:
##  $ HHID       : chr  "idHH00001" "idHH00001" "idHH00001" "idHH00001" ...
##  $ PersonID   : chr  "idPer01" "idPer02" "idPer03" "idPer04" ...
##  $ Stratum    : chr  "idStrt001" "idStrt001" "idStrt001" "idStrt001" ...
##  $ PSU        : chr  "PSU0001" "PSU0001" "PSU0001" "PSU0001" ...
##  $ Zone       : chr  "Rural" "Rural" "Rural" "Rural" ...
##  $ Sex        : chr  "Male" "Female" "Female" "Male" ...
##  $ Age        : int  38 40 20 19 18 35 29 14 13 6 ...
##  $ MaritalST  : Factor w/ 6 levels "Partner","Married",..: 2 2 5 5 5 2 2 5 5 NA ...
##  $ Income     : num  555 555 555 555 555 ...
##  $ Expenditure: num  488 488 488 488 488 ...
##  $ Employment : Factor w/ 3 levels "Unemployed","Inactive",..: 3 3 2 3 2 3 3 NA NA NA .
##  $ Poverty    : Factor w/ 3 levels "NotPoor","Extreme",..: 1 1 1 1 1 3 3 3 3 3 ...
```

As seen in the previous output, the `str` function makes it possible to identify the class of each variable included in the database. For example, the variable `HHID` is of character type, as is the variable `Sex`, whereas the variables `Income` and `Expenditure` are numeric. The other attributes associated with the variables can also be reviewed directly in the code output. This function is especially useful for obtaining an overview of the content and structure of a database.

III Chaining Operations

One of the most useful contributions of `R` is the incorporation of pipes (*pipelines*), which make it possible to build queries, transformations, and new objects from a database in a more intuitive, orderly, and easy-to-interpret way. The chaining operator `%>%`, from the `magrittr` package [Bache and Wickham, 2026] and loaded automatically when using the `tidyverse` library, links several

successive operations so that the result obtained in one step becomes the input for the next step. Below is a simple example of its use with the `BigCity` database, in which the total number of elements contained in the database is calculated using the `count` function.

```
bigcity_data %>%
  count()
```

```
##        n
## 1 150266
```

The previous line of code can be interpreted simply: the database is taken as the starting point, and a count of its records is performed on it. In addition to the `count` function, there is a wide variety of functions that can be combined with the `%>%` operator to perform queries, transformations, and summaries on the data. The `dplyr` package takes advantage of this operator, making it possible to organize the most common operations on a rectangular database. The following subsections present some of these operations, following a typical work sequence: reducing observations, keeping variables, sorting the information, creating new columns, and producing aggregate results.

A `filter`: selecting records

`filter()` keeps the rows that meet one or more conditions. In the following example, two subsets of `bigcity_data` are created: one with men and another with women. The result is two new objects. `male_data` contains only records with `Sex == "Male"`, while `female_data` contains records with `Sex == "Female"`.

```
male_data <- bigcity_data %>%
  filter(Sex == "Male")
female_data <- bigcity_data %>%
  filter(Sex == "Female")
```

It is also possible to filter by poverty status. In this case, only people classified as not poor are kept. The `nonpoor_data` object contains the subset of records whose `Poverty` variable takes the value `"NotPoor"`.

```
nonpoor_data <- bigcity_data %>%
  filter(Poverty == "NotPoor")
```

The `filter()` function also makes it possible to keep records whose values belong to a specific set. In the following examples, records are filtered by particular income values. Note that `%in%` evaluates whether `Income` belongs to the indicated set. Thus, `working_data` keeps incomes equal to 265 or 600, and `income_filter_data` keeps incomes equal to 1000 or 2000.

```
working_data <- bigcity_data %>%
  filter(Income %in% c(265, 600))

income_filter_data <- bigcity_data %>%
  filter(Income %in% c(1000, 2000))
```

B select: selecting variables

The `select()` function keeps specific columns. To continue with the examples, variables identifying the person and household are selected, along with others such as zone, sex, age, and income. The result is a database with fewer columns. `Age` is kept because it will later be used to sort records by age.

```
working_data <- bigcity_data %>%
  select(PersonID, HHID, PSU, Zone, Sex, Age, Income)
```

In addition, `select()` also makes it possible to exclude variables. To remove columns, a minus sign (`-`) is placed before the name of each variable. The `compact_data` object keeps the variables from `working_data`, except for `HHID` and `PersonID`. This operation is useful when working with a more compact database.

```
compact_data <- working_data %>%
  select(-HHID, -PersonID)
```

C arrange: sorting rows

The `arrange()` function sorts the rows of a database according to one or more variables. In the first example, `working_data` is sorted from lowest to highest income, while `sorted_data` contains the same columns as `working_data`, but its rows are sorted by `Income`. The `head()` function shows the first records after sorting.

```
sorted_data <- working_data %>%
  arrange(Income)

sorted_data %>%
  head()
```

```
##   PersonID      HHID     PSU  Zone    Sex Age Income
## 1  idPer01 idHH01522 PSU0112 Rural Female  82   10.0
## 2  idPer01 idHH22167 PSU0112 Rural Female  82   10.0
## 3  idPer01 idHH10745 PSU0893 Rural   Male  68   11.9
## 4  idPer01 idHH31390 PSU0893 Rural   Male  68   11.9
## 5  idPer01 idHH17632 PSU1432 Rural   Male  58   12.1
## 6  idPer02 idHH17632 PSU1432 Rural Female  50   12.1
```

Sorting can also be performed by considering more than one variable. In this example, the database is first organized according to the variable `Sex` and then, within each group, according to the variable `Age`. The following output makes it possible to review the first records after applying this sorting criterion.

```
working_data %>%
  arrange(Sex, Age) %>%
  head()
```

```
##   PersonID      HHID     PSU  Zone    Sex Age Income
## 1  idPer04 idHH00009 PSU0001 Rural Female   0    152
## 2  idPer04 idHH20654 PSU0001 Rural Female   0    152
## 3  idPer06 idHH00042 PSU0004 Rural Female   0    528
## 4  idPer06 idHH20687 PSU0004 Rural Female   0    528
## 5  idPer04 idHH00191 PSU0014 Urban Female   0    355
## 6  idPer04 idHH20836 PSU0014 Urban Female   0    355
```

To sort records from highest to lowest, the `desc()` option is used within the `arrange()` function. In the following example, the database is organized so that the oldest people appear in the first records. This type of sorting makes it possible to quickly identify the highest values of a variable within the reduced database.

```
working_data %>%
  arrange(desc(Age)) %>%
  head()
```

```
##   PersonID      HHID     PSU  Zone    Sex Age Income
## 1  idPer02 idHH01024 PSU0076 Urban Female  98    135
## 2  idPer02 idHH21669 PSU0076 Urban Female  98    135
## 3  idPer01 idHH02916 PSU0219 Rural Female  98    137
## 4  idPer01 idHH23561 PSU0219 Rural Female  98    137
## 5  idPer01 idHH03863 PSU0290 Rural   Male  98    245
## 6  idPer01 idHH24508 PSU0290 Rural   Male  98    245
```

D `mutate`: creating or transforming variables

The `mutate()` function creates new variables or modifies existing variables. Because `BigCity` does not include an explicit geographic region variable, `Region` is constructed from `Stratum`. First, the numeric part of the stratum is extracted with `gsub("\\D", "", Stratum)`, then it is converted to a number with `as.numeric()`, and finally it is divided into five intervals with `cut()`. The result is a new `Region` column in `bigcity_data`, whose categories make it possible to produce regional summaries in the following steps.

```
bigcity_data <- bigcity_data %>%
  mutate(
    Region = cut(
      as.numeric(gsub("\\D", "", Stratum)),
      breaks = 5,
      labels = c("North", "South", "Center", "West", "East")
    )
  )
```

In database analysis, it is also common to recode categorical variables. The following code creates `region_id`, a coded version of `Region` that preserves the order of the regions and assigns two-digit codes. This form of recoding facilitates presentation or linkage with other sources of information.

```
bigcity_data <- bigcity_data %>%
  mutate(
    region_id = factor(
      Region,
      levels = c("North", "South", "Center", "West", "East"),
      labels = c("01", "02", "03", "04", "05")
    )
  )
```

The following example creates `log_income`, which corresponds to the natural logarithm of income (a transformation of the income scale). This operation is frequently used when a monetary variable has high dispersion. The first rows of the relevant variables are then shown.

```
log_income_data <- working_data %>%
  mutate(log_income = log(Income))

log_income_data %>%
  select(PersonID, Income, log_income) %>%
  head()
```

```
##   PersonID Income log_income
## 1  idPer01    555       6.32
## 2  idPer02    555       6.32
## 3  idPer03    555       6.32
## 4  idPer04    555       6.32
## 5  idPer05    555       6.32
## 6  idPer01    298       5.70
```

Note that `mutate()` can create more than one variable in the same step. In addition, variables created at the beginning of `mutate()` can be used in later calculations within the same call. For example, `income_2` doubles the original income and `income_4` doubles `income_2`. The following output shows that `income_4` is equal to four times the initial income.

```
income_transform_data <- log_income_data %>%
  mutate(
    income_2 = 2 * Income,
    income_4 = 2 * income_2
  )

income_transform_data %>%
  select(PersonID, income_2, income_4) %>%
  head()
```

```
##   PersonID income_2 income_4
## 1  idPer01     1110     2220
## 2  idPer02     1110     2220
## 3  idPer03     1110     2220
## 4  idPer04     1110     2220
## 5  idPer05     1110     2220
## 6  idPer01      597     1193
```

A categorical variable can also be created using conditions. In this example, the `case_when()` function classifies age into ranges and then converts the result into an ordered factor. The result is `age_group`, an ordered age-group variable. This transformation facilitates the production of tables by age range.

```
bigcity_data <- bigcity_data %>%
  mutate(
    age_group = case_when(
      Age <= 5 ~ "0-5",
      Age <= 15 ~ "6-15",
      Age <= 30 ~ "16-30",
      Age <= 45 ~ "31-45",
      Age <= 60 ~ "46-60",
      TRUE ~ "Over 60"
    ),
    age_group = factor(
      age_group,
      levels = c(
        "0-5", "6-15", "16-30",
        "31-45", "46-60", "Over 60"
      ),
      ordered = TRUE
    )
  )
```

E `summarise`: descriptive summaries

The `summarise()` function makes it possible to build a summary table from one or more variables in the database. In the following example, two descriptive statistics are calculated for the variable `Income`: the total and the mean. As a result, a table with a single row is obtained, containing the sum of observed incomes and the average income of the records included in the database.

```
bigcity_data %>%
  summarise(
    total_income = sum(Income),
    mean_income = mean(Income)
  )
```

```
##   total_income mean_income
## 1     87893117         585
```

Location measures can also be calculated to describe the position of values within the distribution. In this case, the median, the first decile, the ninth decile, and the difference between the latter two are obtained. The result summarizes different points in the income distribution, while the decile range shows the distance between the bottom 10% and the top 10% of observed incomes.

```
bigcity_data %>%
  summarise(
    median_income = median(Income),
    decile_1 = quantile(Income, 0.1),
    decile_9 = quantile(Income, 0.9),
    decile_range = decile_9 - decile_1
  )
```

```
##   median_income decile_1 decile_9 decile_range
## 1           449      165     1126          961
```

With `summarise()`, it is also possible to calculate dispersion measures, such as the variance, standard deviation, minimum value, maximum value, range, and interquartile range. These statistics make it possible to describe the degree of variability in the incomes observed in the sample. In particular, the range is obtained as the difference between the maximum value and the minimum value, while the interquartile range summarizes the dispersion of the central 50% of the distribution.

```
bigcity_data %>%
  summarise(
    variance_income = var(Income),
    sd_income = sd(Income),
    min_income = min(Income),
    max_income = max(Income),
    range_income = max_income - min_income,
    iqr_income = IQR(Income)
  )
```

```
##   variance_income sd_income min_income max_income range_income iqr_income
## 1          332463       577         10      32920        32910        468
```

F group_by: grouping cases

The `group_by()` function makes it possible to group the database according to one or more variables. When combined with `summarise()`, statistics are calculated independently within each defined group. In the following example, the number of records by region is counted and the result is then sorted from highest to lowest. The following output makes it possible to identify how many records are associated with each region in the database.

```
bigcity_data %>%
  group_by(Region) %>%
  summarise(n = n()) %>%
  arrange(desc(n))
```

```
## # A tibble: 5 x 2
##   Region     n
##   <fct>  <int>
## 1 East   39160
## 2 West   33868
## 3 Center 25944
## 4 South  25898
## 5 North  25396
```

For simple counts, `count()` provides a shorthand way to write the combination of `group_by()` and `summarise(n = n())`. In practice, both alternatives produce the same result, since they group the database according to a categorical variable and calculate how many records belong to each group. For example, the following code reproduces the same results as the previous code.

```
bigcity_data %>%
  count(Region) %>%
  arrange(desc(n))
```

```
##   Region     n
## 1   East 39160
## 2   West 33868
## 3 Center 25944
## 4  South 25898
## 5  North 25396
```

To obtain the number of completed surveys by sex, the same grouping and counting logic is applied. In this case, the database is grouped according to the variable `Sex`, and then the number of records associated with each of its categories is calculated. The result makes it possible to identify how many observations correspond to each sex within the database.

```
bigcity_data %>%
  count(Sex) %>%
  arrange(desc(n))
```

```
##      Sex     n
## 1 Female 79190
## 2   Male 71076
```

The count can also be performed by geographic zone. In this case, the database is grouped by the variable `Zone`, and the number of records corresponding to each of its categories is calculated. The following output shows how the observations are distributed across the different zones in the database.

```
bigcity_data %>%
  count(Zone) %>%
  arrange(desc(n))
```

```
##    Zone     n
## 1 Urban 78164
## 2 Rural 72102
```

In addition to counts, it is also possible to calculate summary measures by group, such as the mean. In this case, to obtain average income by region together with the total number of records, the database is grouped according to the variable `Region` and then `summarise()` is applied. The result is a regional table that presents, for each region, the total number of observations and the corresponding mean income.

```
bigcity_data %>%
  group_by(Region) %>%
  summarise(
    n = n(),
    mean_income = mean(Income)
  )
```

```
## # A tibble: 5 x 3
##   Region     n mean_income
##   <fct>  <int>       <dbl>
## 1 North  25396        509.
## 2 South  25898        644.
## 3 Center 25944        701.
## 4 West   33868        571.
## 5 East   39160        530.
```

The same procedure can be applied to calculate mean income by sex. In this case, the database is grouped according to the variable `Sex`, and then the average of `Income` is obtained for each category. The resulting table makes it possible to compare the observed mean income across the different groups defined by sex.

```
bigcity_data %>%
  group_by(Sex) %>%
  summarise(
    n = n(),
    mean_income = mean(Income)
  )
```

```
## # A tibble: 2 x 3
##   Sex        n mean_income
##   <chr>  <int>       <dbl>
## 1 Female 79190        579.
## 2 Male   71076        591.
```

Finally, the mean, standard deviation, and interquartile range of income are calculated according to employment status. To do this, the database is grouped by the variable `Employment`, and then the descriptive statistics for `Income` are obtained within each group. The following output presents a summary of income behavior for each employment category.

```
bigcity_data %>%
  group_by(Employment) %>%
  summarise(
    n = n(),
    mean_income = mean(Income),
    sd_income = sd(Income),
    iqr_income = IQR(Income)
  )
```

```
## # A tibble: 4 x 5
##   Employment     n mean_income sd_income iqr_income
##   <fct>      <int>       <dbl>     <dbl>      <dbl>
## 1 Unemployed  4630        429.      375.       392.
## 2 Inactive   44104        532.      553.       439.
## 3 Employed   62188        661.      606.       529.
## 4 <NA>       39344        541.      558.       407.
```

Appendix B

Finite Population Inference

A fundamental question in survey analysis is to identify which probability measure governs the inference. In classical statistical inference, the available observations are usually assumed to be independent and identically distributed (IID) random variables. Under this approach, randomness comes from the model that generates the data. However, as Kish [1987] notes, this assumption does not adequately describe information from surveys with complex sample designs, where observations are selected through stratification, clustering, and unequal inclusion probabilities.

I Two Possible Inferences

In household surveys, it is useful to distinguish three objects. First, the finite population, denoted by $U = \{1, 2, \dots, N\}$, which contains all units of interest. Second, the study variable y_k, observed or defined for each unit $k \in U$. Third, the sample design $p(s)$, which assigns probabilities to the possible samples $s \subset U$. In the notation used throughout this document, a unit can be identified as k within cluster or PSU i, belonging to stratum h, so weighted estimators are often written as sums of the form

$$\hat{t}_y = \sum_h \sum_i \sum_k w_{hik}\, y_{hik}$$

where w_{hik} represents the expansion factor or sampling weight associated with the observed unit. In sampling theory, the values y_k are considered fixed once the population has been defined. Randomness is not in the values of the variable, but in the inclusion indicators

$$I_k = \begin{cases} 1, & \text{if } k \in s, \\ 0, & \text{if } k \notin s, \end{cases}$$

which induce the inclusion probability $\pi_k = \Pr_p(I_k = 1)$ and, in turn, the corresponding sampling weight, defined as $w_k = \frac{1}{\pi_k}$.

An estimator that ignores the sampling weights may not adequately represent the target population, especially when the sample comes from a complex design. In design-based inference, the population is considered fixed and the source of randomness comes from the sample selection

mechanism; therefore, its central elements are the inclusion probabilities π_k and the sampling weights w_k.

By contrast, model-based inference considers the values of the variable of interest across the population as realizations of a stochastic process, usually denoted by ξ. Thus, while model-based inference studies properties such as $E_\xi(Y_k)$ and $Var_\xi(Y_k)$, design-based inference is oriented toward incorporating the characteristics of the sample plan into the estimators, so that they are unbiased $E_p(\hat{t}_y) = t_y$.

This distinction makes it possible to understand why, even when there is a reasonable model for the variable of interest, the sample design must be incorporated explicitly if the goal is to obtain valid inferences for the target population. To understand the distinction between model-based inference and design-based inference, we start with a simple example, adapted from Binder [2011]. Suppose that $N = 100$ independent realizations of a Bernoulli variable with parameter $\theta = 0.3$ are generated, where θ represents the expected proportion of unemployed people in a population generated by a model. If $Y_k \sim Bernoulli(\theta)$, then the expectation under the model is $E_\xi(Y_k) = \theta$, while the variance under the model is $Var_\xi(Y_k) = \theta(1-\theta)$.

In the specialized literature, the model ξ is called a superpopulation model, because it generates finite populations. Thus, for a finite population generated by this model, the population mean is $\bar{Y}_U = \frac{1}{N}\sum_{k\in U} Y_k$. Note that, under the model ξ, this mean is unbiased for θ, since

$$E_\xi(\bar{Y}_U) = E_\xi\left(\frac{1}{N}\sum_{k\in U} Y_k\right) = \frac{1}{N}\sum_{k\in U} E_\xi(Y_k) = \theta.$$

The following Monte Carlo simulation reproduces this process. In each repetition, a complete population of size $N = 100$ is generated and its mean is calculated. The average of these population means should approximate θ.

```
set.seed(2026)
population_size <- 100
theta <- 0.3
n_sim_model <- 1000
model_estimates <- rep(NA, n_sim_model)

for (sim in seq_len(n_sim_model)) {
  outcome <- rbinom(population_size, 1, theta)
  model_estimates[sim] <- mean(outcome)
}

cbind(
  theta,
  expected_model_mean = mean(model_estimates),
  bias = mean(model_estimates) - theta
)
```

```
##      theta expected_model_mean    bias
## [1,]   0.3               0.302 0.00225
```

The code sets a seed so that the exercise is reproducible. It then generates `n_sim_model` independent populations using `rbinom()`. The output reports the true parameter `theta`, the average of the simulated means `expected_model_mean`, and the Monte Carlo bias. As expected under the IID Bernoulli model, `expected_model_mean` is very close to 0.3 and the bias is close to zero. The small differences occur because 1000 simulations are used, not an infinite number of repetitions.

In sampling theory, by contrast, the characteristics of interest are fixed population parameters. If a person is unemployed, their status is considered a fixed value of the finite population. What is random is the mechanism through which the sample is selected. The population total is defined as $t_y = \sum_{k \in U} y_k$, while the population mean is $\bar{y}_U = \frac{t_y}{N}$. If a sample s is observed, the Horvitz-Thompson estimator of the total is

$$\hat{t}_y = \sum_{k \in s} \frac{y_k}{\pi_k} = \sum_{k \in s} w_k y_k.$$

As mentioned in previous chapters, when the population size N is unknown, the population mean can be estimated using the weighted ratio estimator, defined as

$$\hat{\bar{y}} = \frac{\sum_{k \in s} w_k y_k}{\sum_{k \in s} w_k}$$

These estimators incorporate the inclusion probabilities into their functional form, and this is the central difference from the simple unweighted average of a sample. To illustrate the impact of the design, suppose that the previous population is divided into N_I PSUs. PSU i has size N_i, total $t_{y_i} = \sum_{k \in U_i} y_{ik}$, and mean $\bar{y}_i = t_{y_i}/N_i$. If n_I PSUs are selected with probabilities proportional to size, then their inclusion probabilities take the following form:

$$\pi_{Ii} = \frac{n_I N_i}{N}$$

When all members of the selected PSUs are observed, the weighted estimator of the population mean, which is approximately unbiased, can be written as

$$\hat{\bar{y}} = \frac{1}{N} \sum_{i \in S_I} \frac{t_{y_i}}{\pi_{Ii}} = \frac{1}{N} \sum_{i \in S_I} \frac{N_i \bar{y}_i}{n_I N_i / N} = \frac{1}{n_I} \sum_{i \in S_I} \bar{y}_i.$$

This expression illustrates why, when the sampling design is complex (in this case, probability proportional to size selection), the average of the cluster means provides unbiasedness. By contrast, the simple average of the people observed in the selected PSUs would be:

$$\bar{y}_s = \frac{\sum_{i \in S_I} t_{y_i}}{\sum_{i \in S_I} N_i},$$

which treats the sample as if it were self-weighting. This estimator is not unbiased when the selection probabilities are unequal or when cluster size is related to the variable of interest. The following simulation fixes a finite population and repeatedly selects households with probabilities

proportional to size using `S.piPS()` from the `TeachingSampling` package [Gutiérrez, 2020]. In each sample, two estimators are calculated: `design_estimates`, which corresponds to $\hat{\bar{y}}$, and `simple_estimates`, which corresponds to the simple average of the people observed in the selected PSUs, $\bar{y}_s$. In addition, these estimators are compared with the model parameter `theta_population`, corresponding to $\theta = 0.3$.

```
library(TeachingSampling)

set.seed(2026)
population_size <- 100
theta <- 0.3
n_sim_sampling <- 1000
design_estimates <- simple_estimates <- rep(NA, n_sim_sampling)

cluster_size <- rep(2:6, each = 5)
n_clusters <- length(cluster_size)
cluster_id <- rep(seq_len(n_clusters), cluster_size)
size_center <- weighted.mean(cluster_size, cluster_size)
prob_person <- theta + rep((cluster_size - size_center) / 12, cluster_size)
outcome <- rbinom(population_size, 1, prob_person)
theta_population <- mean(outcome)
cluster_totals <- tapply(outcome, cluster_id, sum)
cluster_means <- tapply(outcome, cluster_id, mean)

sampled_cluster_count <- floor(n_clusters * 0.3)
for (sim in seq_len(n_sim_sampling)) {
  pps_sample <- S.piPS(sampled_cluster_count, cluster_size)
  sampled_clusters <- pps_sample[, 1]
  sampled_cluster_size <- cluster_size[sampled_clusters]
  sampled_cluster_total <- cluster_totals[sampled_clusters]
  sampled_cluster_mean <- cluster_means[sampled_clusters]
  design_estimates[sim] <- mean(sampled_cluster_mean)
  simple_estimates[sim] <-
    sum(sampled_cluster_total) / sum(sampled_cluster_size)
}

cbind(
  theta_population,
  expected_pps = mean(design_estimates),
  bias_pps = mean(design_estimates) - theta_population,
  expected_simple = mean(simple_estimates),
  bias_simple = mean(simple_estimates) - theta_population
)
```

```
##      theta_population expected_pps bias_pps expected_simple bias_simple
## [1,]              0.3        0.304  0.00368           0.358      0.0585
```

The first block of code generates households of unequal sizes, between 2 and 6 people. To make the design effect visible, the individual probability of unemployment varies smoothly with household size, but is centered so that the expected population average remains close to `theta`. Then `theta_population` is calculated, which is the actual proportion of unemployment in that specific population. In each repetition, `S.piPS()` selects PSUs with probabilities proportional to size; then the mean of the means of the selected PSUs and the simple average of the observed people are calculated. The output compares both estimators with `theta_population`. The Monte Carlo bias of `expected_pps` is close to zero, while that of `expected_simple` is not, showing the result of ignoring the design in this exercise.

II The Integration of Both Inferences

Nevertheless, in many applications these two inferential frameworks are not presented as mutually exclusive approaches, but as complementary components of the same problem. In particular, the sample may be considered to have been obtained through a complex probability design while, at the same time, the values of the variable of interest respond to a superpopulation model. This formulation leads to a double-inference framework, in which uncertainty does not come from a single source, but from the combination of the mechanism that generates the population values (superpopulation model) and the mechanism that determines which units are observed (sampling design).

Formally, two probability measures are involved: ξ, associated with the process that generates the values Y_k, and $p(\cdot)$, associated with the sample design that selects the sample s. Therefore, the properties of the estimators must be evaluated while simultaneously considering both sources of variation. Under this framework, the combined expectation can be expressed as

$$E_{\xi p}(\hat{\theta}) = E_{\xi}\{E_p(\hat{\theta} \mid U)\},$$

where the behavior of the estimator is first evaluated under the design, conditional on the finite population generated, and then averaged over the superpopulation model. In this context, the full joint likelihood may be intractable, especially when the inclusion probabilities depend on variables related to the response. The solution proposed by Pfeffermann [1993] is the maximum pseudo-likelihood (MPV) technique, which weights each unit's contribution by the inverse of its inclusion probability.

A Maximum Likelihood Method

The maximum likelihood method starts from a random sample $y_1, \ldots, y_n$ generated by a distribution with density or probability function $f(y;\theta)$. If the observations are independent and identically distributed (IID), the likelihood function is defined as

$$L(\theta) = \prod_{k=1}^{N} f(y_k;\theta).$$

Because products can be difficult to manipulate, the log-likelihood is used:

$$\ell(\theta) = \sum_{k=1}^{N} \log f(y_k; \theta)$$

The maximum likelihood estimator $\hat{\theta}$ is the value of θ that maximizes $\ell(\theta)$. If the function is differentiable, this value satisfies the score equations:

$$U(\theta) = \frac{\partial\ \ell(\theta)}{\partial\ \theta} = \sum_{k=1}^{N} u_k(\theta) = 0$$

where $u_k(\theta) = \frac{\partial}{\partial\theta} \log f(y_k; \theta)$ is the contribution of unit k to the total score. For a Bernoulli distribution with parameter θ, the probability function is $f(y_k; \theta) = \theta^{y_k}(1-\theta)^{1-y_k}$, for $y_k \in \{0, 1\}$. The log-likelihood is

$$\ell(\theta) = \sum_{k=1}^{N} [y_k \log(\theta) + (1 - y_k) \log(1 - \theta)]$$

By differentiating, setting equal to zero, and solving, the MV estimator is obtained, defined by:

$$\hat{\theta}_{MV} = \frac{1}{N} \sum_{k=1}^{N} y_k$$

Thus, the sample proportion is the natural estimator of the Bernoulli parameter. The key point is that this result depends on the assumption that the observations have the same distribution and do not come from a design with unequal probabilities.

On the other hand, in a multiple linear regression model with normal errors, we have $\mathbf{y} \sim N(\mathbf{X}\beta, \sigma^2 I)$. In this case, the log-likelihood, omitting constants, can be written as

$$\ell(\beta, \sigma^2) = -\frac{n}{2} \log(\sigma^2) - \frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta)$$

Maximizing this expression with respect to β is equivalent to minimizing the sum of squared residuals. Therefore, the maximum likelihood estimator of β coincides with the ordinary least squares estimator $\widehat{\beta}_{MV} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$. This result is important because it shows that many common statistical procedures can be interpreted as maximum likelihood under specific model assumptions. However, if the data come from a complex survey, the matrix $\mathbf{X}'\mathbf{X}$ does not by itself reflect the sample selection mechanism.

B Maximum Pseudo-Likelihood Method

When the sample design is complex, the observed data do not necessarily satisfy the assumptions of independence and identical distribution. Some units may represent many people in the population, and others only a few. In addition, two units selected within the same cluster may be correlated. In this context, directly using the ordinary log-likelihood can produce estimates that describe the sample, but not the target population.

Maximum pseudo-likelihood, developed from the results of Binder [1983], incorporates the sampling weights into the log-likelihood. For observed units $k \in s$, it is defined as

$$\ell_p(\theta) = \sum_{k \in s} w_k \log f(y_k; \theta)$$

The weight $w_{hik} = 1/\pi_{hik}$ indicates how many population units the observation y_{hik} represents. Therefore, the contribution of a unit with a low inclusion probability is amplified in the pseudo-likelihood. The resulting estimating equations are

$$U_p(\theta) = \sum_{k \in s} w_k \; u_k(\theta) = 0$$

MPV estimation preserves the logic of maximum likelihood, but replaces the ordinary *score* with a weighted *score*. The solution $\widehat{\theta}_{MPV}$ is the value that makes the weighted sum of individual contributions equal to zero. For a Bernoulli distribution, the pseudo-log-likelihood is

$$\ell_p(\theta) = \sum_{k \in s} w_k \left[y_k \log(\theta) + (1 - y_k) \log(1 - \theta) \right].$$

By differentiating and setting equal to zero, we obtain:

$$\frac{\partial l_p(\theta)}{\partial \theta} = \sum_{k \in s} w_k \left[\frac{y_k}{\theta} - \frac{1 - y_k}{1 - \theta} \right] = 0$$

The solution is:

$$\hat{\theta}_{MPV} = \frac{\sum_{k \in s} w_k y_k}{\sum_{k \in s} w_k} = \hat{p}_d$$

Therefore, for a binary variable, MPV leads to the weighted estimator of the proportion, equivalent to the Hájek estimator. This result directly connects pseudo-likelihood theory with the proportion estimators presented in the chapters on categorical variables.

Likewise, for a multiple linear regression model, the weighted pseudo-log-likelihood implies minimizing a weighted sum of squares:

$$(\mathbf{y} - \mathbf{X}\beta)' \mathbf{W} (\mathbf{y} - \mathbf{X}\beta),$$

where

$$\mathbf{W} = \begin{pmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{pmatrix}.$$

The solution is the weighted least squares estimator:

$$\hat{\beta}_{MPV} = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{y}$$

This result naturally extends the ordinary estimator to the context of complex surveys: each observation contributes to the model fit in proportion to its sampling weight. Nevertheless, for inference it is not enough to weight the point estimate; the variance must also be estimated correctly, taking strata, clusters, and weights into account.

The following nested Monte Carlo simulation illustrates double inference. In the outer loop, a new population is generated under the Bernoulli model. In the inner loop, many samples are selected by design and the estimator that incorporates the design is compared with the simple sample average.

```
set.seed(2026)
population_size <- 100
theta <- 0.3
n_sim_population <- 100
expected_design_estimates <- expected_simple_estimates <- population_means <-
  rep(NA, n_sim_population)

cluster_size <- rep(2:6, each = 5)
n_clusters <- length(cluster_size)
cluster_id <- rep(seq_len(n_clusters), cluster_size)
size_center <- weighted.mean(cluster_size, cluster_size)
prob_person <- theta + rep((cluster_size - size_center) / 12, cluster_size)
sampled_cluster_count <- floor(n_clusters * 0.3)

for (population_sim in seq_len(n_sim_population)) {
  outcome <- rbinom(population_size, 1, prob_person)
  population_means[population_sim] <- mean(outcome)
  cluster_totals <- tapply(outcome, cluster_id, sum)
  cluster_means <- tapply(outcome, cluster_id, mean)

  design_estimates <- simple_estimates <- rep(NA, 100)
  for (sample_sim in seq_len(100)) {
    pps_sample <- S.piPS(sampled_cluster_count, cluster_size)
    sampled_clusters <- pps_sample[, 1]
    sampled_cluster_size <- cluster_size[sampled_clusters]
    sampled_cluster_total <- cluster_totals[sampled_clusters]
    sampled_cluster_mean <- cluster_means[sampled_clusters]
    design_estimates[sample_sim] <- mean(sampled_cluster_mean)
    simple_estimates[sample_sim] <-
      sum(sampled_cluster_total) / sum(sampled_cluster_size)
  }
  expected_design_estimates[population_sim] <- mean(design_estimates)
  expected_simple_estimates[population_sim] <- mean(simple_estimates)
}
```

```
cbind(
  theta,
  mean_population = mean(population_means),
  expected_pps = mean(expected_design_estimates),
  bias_pps = mean(expected_design_estimates) - theta,
  expected_simple = mean(expected_simple_estimates),
  bias_simple = mean(expected_simple_estimates) - theta
)
```

```
##      theta mean_population expected_pps bias_pps expected_simple bias_simple
## [1,]   0.3           0.304        0.302  0.00248           0.332      0.0317
```

The code reproduces the two sources of randomness. The outer loop represents the model ξ, because it generates new finite populations with `rbinom()`. The inner loop represents the design p, because multiple samples are selected for each population with `S.piPS()`. The output includes `mean_population`, which is the average of the generated population means; `expected_pps`, which summarizes the behavior of the estimator that incorporates the design; and `expected_simple`, which summarizes the simple average of the selected sample. When the design is incorporated correctly, the Monte Carlo bias with respect to `theta` should be small. The comparison with `expected_simple` shows why survey-based inference requires weights and design variances, even when starting from a simple statistical model.

In summary, ordinary maximum likelihood is appropriate when observations can be treated as IID under a probabilistic model. In complex surveys, maximum pseudo-likelihood offers a natural extension: it preserves the structure of the estimating equations, but weights each individual contribution by its sampling weight. Thus, statistical models are fitted in a way that is more coherent with the population the survey seeks to represent.

References

Bibliography

Tihomir Asparouhov. General multi-level modeling with sampling weights. *Communications in Statistics, Theory and Methods*, 35(3):439–460, 2006. doi: 10.1080/03610920500476598.

Stefan Milton Bache and Hadley Wickham. *magrittr: A Forward-Pipe Operator for R*, 2026. URL https://CRAN.R-project.org/package=magrittr. R package version 2.0.5.

Douglas Bates, Martin Maechler, Ben Bolker, and Steven Walker. *lme4: Linear Mixed-Effects Models using 'Eigen' and S4*, 2026. URL https://CRAN.R-project.org/package=lme4. R package version 2.0-1.

Leonardo Bautista. Diseños de muestreo estadístico. *Universidad Nacional de Colombia*, 1998.

David A Binder. On the variances of asymptotically normal estimators from complex surveys. *International Statistical Review/Revue Internationale de Statistique*, pages 279–292, 1983.

David A. Binder. Estimating model parameters from a complex survey under a model-design randomization framework. *Pakistan Journal of Statistics*, 27(4), 2011.

David A Binder and Milorad S Kovacevic. Estimating some measures of income inequality from survey data: an application of the estimating equations approach. *Survey Methodology*, 21: 137–146, 1995.

William J. Browne and David Draper. A comparison of bayesian and likelihood-based methods for fitting multilevel models. *Bayesian Analysis*, 1(3):473–514, 2006. doi: 10.1214/06-BA117.

Christian Bruch, Ralf Münnich, and Stefan Zins. Variance estimation for complex surveys. *Deliverable D3*, 1, 2011.

Tianji Cai. Investigation of ways to handle sampling weights for multilevel model analyses. *Sociological Methodology*, 43(1):178–219, 2013. doi: 10.1177/0081175013490035.

William Gemmell Cochran. *Sampling techniques*. John Wiley & Sons, 1977.

Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1): 1–26, 1979. doi: 10.1214/aos/1176344552.

Robert E. Fay. On adjusting the pearson chi-square statistic for cluster sampling. *Proceedings of the Social Statistics Section, American Statistical Association*, pages 402–405, 1979.

Ivan P. Fellegi. Approximate tests of independence and goodness of fit based on stratified multistage samples. *Journal of the American Statistical Association*, 75(370):261–268, 1980. doi: 10.2307/2287405.

W. Holmes Finch, Jocelyn E. Bolin, and Ken Kelley. *Multilevel Modeling Using R*. CRC Press, Boca Raton, FL, 2 edition, 2019.

Greg Freedman Ellis and Ben Schneider. *srvyr: 'dplyr'-Like Syntax for Summary Statistics of Survey Data*, 2024. URL https://CRAN.R-project.org/package=srvyr. R package version 1.3.1.

Wayne A. Fuller. Regression analysis for sample survey. *Sankhya, Series C*, 37:117–132, 1975.

Wayne A. Fuller. Regression estimation for survey samples. *Survey Methodology*, 28(1):5–23, 2002.

Jack G Gambino and Pedro Luis do Nascimento Silva. Sampling and estimation in household surveys. In *Handbook of Statistics*, volume 29, pages 407–439. Elsevier, 2009.

Andrew Gelman and Jennifer Hill. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, New York, 2006.

Harvey Goldstein. *Multilevel Statistical Models*. John Wiley & Sons, Chichester, 4 edition, 2011.

Hugo Andrés Gutiérrez. *Estrategias de muestreo: diseño de encuestas y estimación de parámetros*. Ediciones de la U, segunda edición edition, 2016. ISBN 978-958-762-586-8.

Hugo Andrés Gutiérrez. *TeachingSampling: Selection of Samples and Parameter Estimation in Finite Population*, 2020. URL https://CRAN.R-project.org/package=TeachingSampling. R package version 4.1.1.

Hugo Andrés Gutiérrez and Giovany Babativa-Márquez. Efectos de diseño para indicadores sociales en américa latina: función generalizada de varianza para estimadores directos provenientes de encuestas de hogares, 2023. ISSN 1680-8789 (versión electrónica), 1994-7364 (versión impresa). URL https://repositorio.cepal.org/server/api/core/bitstreams/feb8c8be-6434-41f2-8fd9-f12f71a0739a/content.

Morris H Hansen, William N Hurwitz, William G Madow, et al. Sample survey methods and theory. 1953.

Steven G Heeringa, Brady T West, Steve G Heeringa, and Patricia A Berglund. *Applied survey data analysis*. chapman and hall/CRC, 2017.

Lionel Henry, Hadley Wickham, and Winston Chang. *ggstance: Horizontal 'ggplot2' Components*, 2024. URL https://CRAN.R-project.org/package=ggstance. R package version 0.3.7.

Guilherme Jacob, Djalma Pessoa, and Anthony Damico. *convey: Income Concentration Analysis with Complex Survey Samples*, 2024. URL https://CRAN.R-project.org/package=convey. R package version 1.0.1.

Graham Kalton and Ismael Flores-Cervantes. Weighting methods. *Journal of official statistics*, 19 (2):81, 2003.

Leslie Kish. Survey sampling. 1965.

Leslie Kish. *Statistical Design for Research*. John Wiley & Sons, New York, 1987.

Leslie Kish and Martin R. Frankel. Inference from complex samples. *Journal of the Royal Statistical Society, Series B*, 36:1–37, 1974.

Edward L Korn and Barry I Graubard. Analysis of large health surveys: accounting for the sampling design. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 158 (2):263–295, 1995.

Milorad S Kovacevic and David A Binder. Variance estimation for measures of income inequality and polarization-the estimating equations approach. *Journal of Official Statistics*, 13(1):41, 1997.

JG Kovar, JNK Rao, and CFJ Wu. Bootstrap and other methods to measure errors in survey estimates. *Canadian Journal of Statistics*, 16(S1):25–45, 1988.

Ita G. G. Kreft and Jan De Leeuw. *Introducing Multilevel Modeling*. Sage Publications, London, 1998.

Matti Langel and Yves Tillé. Variance estimation of the gini index: revisiting a result several times published. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 176(2): 521–540, 2013.

Jianzhu Li and Richard Valliant. Linear regression diagnostics in cluster samples. *Journal of Official Statistics*, 31(1):61–75, March 2015. ISSN 0282-423X, 2001-7367. doi: 10.1515/jos-2015-0003.

Jacob A. Long. *jtools: Analysis and Presentation of Social Scientific Data*, 2026. URL https://CRAN.R-project.org/package=jtools. R package version 2.3.1.

Dana Loomis, David B Richardson, and Leslie Elliott. Poisson regression analysis of ungrouped data. *Occupational and environmental medicine*, 62(5):325–329, 2005.

Thomas Lumley. *Complex Surveys: A Guide to Analysis Using R: A Guide to Analysis Using R*. John Wiley and Sons, 2010.

Thomas Lumley. survey: analysis of complex survey samples, 2024. R package version 4.5.

Thomas Lumley. *svyVGAM: Design-Based Inference in Vector Generalised Linear Models*, 2026. URL https://CRAN.R-project.org/package=svyVGAM. R package version 1.3.

Daniel Lüdecke, Mattan S. Ben-Shachar, Indrajeet Patil, Philip Waggoner, and Dominique Makowski. *performance: Assessment of Regression Models Performance*, 2026. URL https://CRAN.R-project.org/package=performance. R package version 0.17.0.

Juan Merlo, Basile Chaix, Henrik Ohlsson, Anders Beckman, Kristina Johnell, Per Hjerpe, Lennart Råstam, and Klaus Larsen. 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, 2006. doi: 10.1136/jech.2004.029454.

Stefano Meschiari. *latex2exp: Use LaTeX Expressions in Plots*, 2026. URL https://CRAN.R-project.org/package=latex2exp. R package version 0.9.8.

Elizabeth A. Molina and Chris J. Skinner. Pseudo-likelihood and quasi-likelihood estimation for complex sampling schemes. *Computational Statistics & Data Analysis*, 13:395–405, 1992.

Naciones Unidas. *Diseño de Muestras para Encuestas de Hogares: Directrices Prácticas.* Number 98 in Estudios de Métodos, Serie F. Naciones Unidas, Nueva York, 2009. URL https://unstats.un.org/unsd/publication/seriesf/seriesf_98s.pdf.

John A. Nelder and Robert W. M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384, 1972. doi: 10.2307/2344614.

Guillaume Osier. Variance estimation for complex indicators of poverty and inequality using linearization techniques. In *Survey research methods*, volume 3, pages 167–195, 2009.

Inho Park, Marianne Winglee, Jay Clark, Keith Rust, Andrea Sedlak, and David Morganstein. Design effects and survey planning. In *Proceedings of the Section on Survey Research Methods*, pages 3179–3186, 2003. URL https://www.asasrms.org/Proceedings/y2003/Files/JSM2003-000156.pdf. American Statistical Association, Alexandria, VA.

Edzer Pebesma, Roger Bivand, Etienne Racine, Michael Sumner, Ian Cook, Tim Keitt, Robin Lovelace, Hadley Wickham, Jeroen Ooms, and Kirill Müller. *sf: Simple Features for R*, 2026. URL https://CRAN.R-project.org/package=sf. R package version 1.1-1.

Thomas Lin Pedersen. *patchwork: The composer of plots*, 2025. URL https://CRAN.R-project.org/package=patchwork. R package version 1.3.2.

Danny Pfeffermann. The role of sampling weights when modeling survey data. *International Statistical Review*, 61(2):317–337, 1993. doi: 10.2307/1403631.

Danny Pfeffermann. Modelling of complex survey data: Why model? Why is it a problem? How can we approach it? *Survey Methodology*, 37(2):115–136, 2011.

Danny Pfeffermann, Chris J. Skinner, David J. Holmes, Harvey Goldstein, and Jon Rasbash. Weighting for unequal selection probabilities in multilevel models. *Journal of the Royal Statistical Society: Series B*, 60(1):23–40, 1998. doi: 10.1111/1467-9868.00106.

Christopher Prener, Timo Grossenbacher, and Angelo Zehr. *biscale: Tools and Palettes for Bivariate Thematic Mapping*, 2025. URL https://CRAN.R-project.org/package=biscale. R package version 1.1.0.

Sophia Rabe-Hesketh and Anders Skrondal. Multilevel modelling of complex survey data. *Journal of the Royal Statistical Society: Series A*, 169(4):805–827, 2006. doi: 10.1111/j.1467-985X.2006.00426.x.

Sophia Rabe-Hesketh and Anders Skrondal. *Multilevel and Longitudinal Modeling Using Stata.* Stata Press, College Station, TX, 3 edition, 2012.

J. N. K. Rao, C. F. J. Wu, and K. Yue. Some recent work on resampling methods for complex surveys. *Survey Methodology*, 18:209–217, 1992.

Jon N. K. Rao and Alastair J. Scott. On chi-squared tests for multiway contingency tables with cell proportions estimated from survey data. *The Annals of Statistics*, 12(1):46–60, 1984.

David Robinson, Alex Hayes, Simon Couch, and Emil Hvitfeldt. *broom: Convert Statistical Objects into Tidy Tibbles*, 2026. URL https://CRAN.R-project.org/package=broom. R package version 1.0.13.

Carl-Erik Särndal and Sixten Lundström. *Estimation in surveys with nonresponse*. John Wiley & Sons, 2005.

Carl-Erik Särndal, Bengt Swensson, and Jan Wretman. *Model assisted survey sampling*. Springer Science & Business Media, 2003.

Babubhai V. Shah, Maurice M. Holt, and Ralph F. Folsom. Inference about regression models from sample survey data. *Bulletin of the International Statistical Institute*, 41(3):43–57, 1977.

Chris J. Skinner, David Holt, and Tom M. F. Smith, editors. *Analysis of complex surveys*. John Wiley & Sons, New York, 1989.

Tom AB Snijders and Roel Bosker. Multilevel analysis: An introduction to basic and advanced multilevel modeling. 2011.

Frederick Solt and Yue Hu. *dotwhisker: Dot-and-Whisker Plots of Regression Results*, 2025. URL https://CRAN.R-project.org/package=dotwhisker. R package version 0.8.4.

Cristian Fernando Tellez Piñerez and Diego Fernando Lemus Polanía. *Estadística Descriptiva y Probabilidad con aplicaciones en R*. Fundación Universitaria Los Libertadores, 2015.

Martijn Tennekes. *tmap: Thematic Maps*, 2026. URL https://CRAN.R-project.org/package=tmap. R package version 4.3.

D. Roland Thomas and Jon N. K. Rao. Small-sample comparisons of level and power for simple goodness-of-fit statistics under cluster sampling. *Journal of the American Statistical Association*, 82(398):630–636, 1987.

United Nations Statistics Division. *Handbook of Surveys on Households and Individuals Foundations and Emerging Approaches*. United Nations, 2026.

Richard Valliant. *svydiags: Regression Model Diagnostics for Survey Data*, 2024. URL https://CRAN.R-project.org/package=svydiags. R package version 0.7.

Gregory R. Warnes, Ben Bolker, Thomas Lumley, Arni Magnusson, Bill Venables, Genei Ryodan, and Steffen Moeller. *gtools: Various R Programming Tools*, 2023. URL https://CRAN.R-project.org/package=gtools. R package version 3.9.5.

Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, Alex Hayes, Lionel Henry, Jim Hester, Max Kuhn, Thomas Lin Pedersen, Evan Miller, Stephan Milton Bache, Kirill Müller, Jeroen Ooms, David Robinson, Dana Paige Seidel, Vitalie Spinu, Kohske Takahashi, Davis Vaughan, Claus Wilke, Kara Woo, and Hiroaki Yutani. *tidyverse: Easily Install and Load the Tidyverse*, 2026a. URL https://CRAN.R-project.org/package=tidyverse. R package version 2.0.0.

Hadley Wickham, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, Dewey Dunnington, and Teun van den Brand. *ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*, 2026b. URL https://CRAN.R-project.org/package=ggplot2. R package version 4.0.3.

Hadley Wickham, Romain François, Lionel Henry, Kirill Müller, and Davis Vaughan. *dplyr: A Grammar of Data Manipulation*, 2026c. URL https://CRAN.R-project.org/package=dplyr. R package version 1.2.1.

Claus O. Wilke. *cowplot: Streamlined Plot Theme and Plot Annotations for 'ggplot2'*, 2025. URL https://CRAN.R-project.org/package=cowplot. R package version 1.2.0.

Leland Wilkinson. *The grammar of graphics*. Springer, New York, 2 edition, 2005.

Kirk M Wolter and Kirk M Wolter. *Introduction to variance estimation*, volume 53. Springer, 2007.