eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

PhD Thesis:

Development and validation of prediction models for complex sampling data

Amaia Iparragirre Letamendia

2024

Supervised by:

Irantzu Barrio

Inmaculada Arostegui

A thesis submitted for the degree of Philosophy Doctor in Mathematics & Statistics

This Ph.D. thesis was financially supported by the University of the Basque Country (UPV/EHU) by the Predoctoral Grant PIF18/213, by Biostit Research Group (IT620-13), MATHMODE: Group on Applied Mathematical Modeling, Statistics and Optimization (IT1294-19, IT1456-22), S3M1P4R (PID2020-115882RB-I00) and MTM2016-74931-P.

Badira barrutik besterik ireki ezin diren ate batzuk,
besterik ez.

Gorka Urbizu (2024). *Hasiera bat.* Besterik ez.

Esker Onak - Acknowledgements

"Badira barrutik besterik ireki ezin diren ate batzuk". Badira, baita ere, ateak irekitzen laguntzen dizuten pertsonak. Baita ateak hor daudela ikustarazten dizun jendea ere. Doktorego tesia izan den harrizko bide honetan horrelako pertsona askorekin gurutzatzeko zortea izan dut, aurretik pentsatu gabe, ohartu gabe, bide hori leunagoa bihurtu didatenak. Horiei guztiei nire esker ona adierazi nahiko nieke lerro hauetan. "Besterik ez".

Ezingo nuke esker onez betetako lerro hauek idazten hasi nire bi zuzendariak aipatu gabe. Dena den, aitortzen dut paragrafo hau idatzi dudan azkena izan dela: ezin hitzik aurkitu nigatik egin duzuen guztia eskertzeko. Irantzu, eskerrik asko egunero-egunero nire ondoan egoteagatik, konfiantzagatik eta nigan sinisteagatik (nik neure buruan baino askoz gehiago). Elkarrekin hasi ginen bide honetan, eta elkarrekin hazi gara. Inma, eskerrik asko hasieratik emandako babesagatik eta aholku guztiengatik. Izan nitzakeen zuzendaririk onenak izan ditut. Zuek ateak hor daudela ikustarazi eta irekitzen lagundu bakarrik ez, ateak ere eraiki dituzue niretzat.

I would also like to thank the two reviewers who have revised and made a favourable report of this thesis.

Oso eskertuta nago, baita ere, EUSTAT-eko kideei, batez ere, Jorge eta Marinari, urte hauetan gugan izan duzuen konfiantzagatik eta zuekin lanean jarraitzeko aukera emateagatik.

During the last few years, I have also had the opportunity to do research stays at two different universities under the guidance of two researchers whom I really

appreciate. Muchísimas gracias Lupe, por acogerme en Barcelona y hacer que mi primera estancia de investigación fuera tan agradable. Muchas gracias también por la sabiduría, por el finde en Pals y, cómo no, por los gintonics. Thank you very much Thomas, Kia Ora. Working with you has been an incredible experience. No me gustaría olvidarme de las compañeras y los compañeros que me han acompañado en estas estancias. Moltes gràcies Dani, Jordi, Klaus, Leire i Xavier pels riures, els cafès i fer-me sentir com a casa mentre vaig estar a Barcelona. Muchísimas gracias a ti también, Claudia, estar en la otra punta del mundo ha sido un poco más fácil gracias a ti. Y a Dani, por compartir tu experiencia previa conmigo, por todos los buenos consejos y contagiarme el entusiasmo.

Nire eskerrik beroenak nire familia zientifikoari. Bereziki, mila mila esker Arantza, orain dela urte batzuk familia hau osatzen hasteagatik, eta hainbeste urtetan zehar lanean jarraitzeagatik. Gu guztion eredu izan zara eta zara, gaur egun ere. Eskerrik asko gu guztiongan sinesteagatik. Eskerrik asko Josu, azken urtebete pasatxoan gure egunerokotasuna hobea egiteagatik, barreengatik eta gauzak perspektiban jartzen laguntzeagatik. Eskerrik asko Maider, gertutasunagatik eta urte hauetan bizitako momentu bakoitzarengatik. Tampoco me gustaría olvidarme de las amigas y los amigos de la Sociedad Española de Bioestadística (SEB), Biostatnet, y del mundo de la investigación, en general, que he podido hacer durante los últimos años. Marta, Natalia, Joaquín, Coté, Anabel, David, Roi... entre otros muchos. Las vivencias compartidas con vosotras y vosotros han hecho de este camino un poco más fácil de recorrer y, sin duda, muchísimo más divertida. Tengo un montón de mentores y mentoras sin haberlo pedido. Muchas gracias por todo el apoyo y cariño.

Graduko lagunei (aurkitu nitzakeen bidelagun onenak izan zarete): Aitor, Elene, Ibon, Jone, Lara, Lore, Rut eta Sara. Eskerrak kuadrilako lagunei (zuek bai, "ardue bezela...") eta familiari. Osaba Balere, badakit norbaitek tesi hau irakurtzekotan zu izango zinatekeela irakurriko lukeena. Ze pena urte hauetako abenturak zurekin konpartitu ezin izana. Aitona, badakit zeinen harro egongo zinatekeen (izeba Karmen, bai, zu ere bai). Dena den, amona, badakit zure partez, eta falta direnen partez, sentituko duzula poza. Ama, aita eta Julen, eskerrik asko eman didazuen babes guztiagatik. Baita gure heziketan jarri duzuen indarragatik ere. Eta Mikel, eskerrik asko pazientzia infinituagatik, ulertu ezin diren gauzak ulertzeagatik, eta ateak aukeratzeko emandako askatasunagatik.

Eskerrik asko, guztioi. Jarrai dezagun bidea egiten, ateak irekitzen.

"Hasiera bat. Besterik ez".

Contents

Laburpena

Diseinu konplexuetatik eratorritako datuen erabilera nabarmen hazi da azken urteetan, Estatistika Ofizialaren testuinguruan inkestak burutzeko bereziki. Inkestetako datuak lortzeko, inkestaren helburuak betetzeko interesekoa den populazioa lagindu ohi da aurretik definitutako diseinu konplexu zehatz bat jarraituz. Laginketa prozesu hau etapa batean edo gehiagotan burutu daiteke laginketa teknika desberdinak euren artean konbinatuz. Laginketa teknika ezagunenetako batzuk estratifikazioa (populazioko indibiduoak talde ezberdinetan banatzea) eta *cluster* edo multzokatze teknika (populazioko indibiduoak multzoka elkartzea) dira. Zorizko laginketa sinplearen bidez lagindutako datuekin alderatuta, diseinu konplexuetan oinarrituta lortutako datuetan populazioko indibiduo bakoitzari lagindua izateko probabilitate bat esleitzen zaio. Probabilitate hauek guztiak 0-ren desberdinak dira eta, horrez gain, populazioko indibiduo guztien probabilitateak ez dira berdinak. Ondorioz, lagindua izan den indibiduo bakoitzari laginketa-pisu bat esleitzen zaio laginketa prozesuaren amaieran. Laginketa-pisu hauetako bakoitzak, lagindutako indibiduo horrek ordezkatzen dituen populazioko indibiduo kopurua adierazten du, eta lagindua izateko esleitu zaion probabilitatearen alderantzizkoa kalkulatuz definitzen dira. Diseinu konplexuko laginketak oso baliagarriak dira, adibidez, populazioko talde gutxituen ordezkapen hobeak lortzeko. Horrez gain, datu-bilketaren gastu ekonomikoak murrizteko ere oso erabilgarriak dira. Hau dela eta, gizarte eta osasun zientzietan besteak beste, gero eta gehiago biltzen dituzte datuak laginketa diseinu konplexuetan oinarrituz.

Gaur egun, beste helburu batzuez gain, diseinu konplexuko laginketetan oina-

rrituz bildutako datuak eredu aurresaleak garatzeko ere erabiltzen dira. Eredu aurresaleen garrantzia azpimarragarria da hainbat eta hainbat arlotan. Eredu aurresaleen helburu nagusia, aldagai azaltzaile batzuen inguruko informazioa erabilita, intereseko erantzun aldagaiaren balioak aurresatea da, aurretik behatutako datuetan oinarrituta. Baina eredu aurresale hauek egunerokotasunean erabili ahal izateko, beharrezkoa da eredu aurresale hauen garapen prozesuan zehar pausu ezberdinak jarraitzea, lortu dugun eredua ona eta praktikan erabilgarria dela bermatu ahal izateko. Adibidez, eskura ditugun aldagai azaltzaile guztien artetik garrantzitsuenen aukeraketa egitea pauso garrantzitsua da eredu aurresale onak garatu ahal izateko. Gainera, doitutako ereduaren aurresateko gaitasuna aztertu behar da eredu aurresaleak praktikan erabiltzen hasi aurretik. Aurretik aipatutako pauso hauek eta beste batzuk jarraitzea da gakoa praktikan erabilgarriak izango diren eredu aurresaleak lortu ahal izateko.

Dena dela, eredu aurresale onak garatzeko erabili ohi diren teknika estatistiko gehienek erabiltzen ari garen datuak askeak eta berdinki banatuak direla asumitzen dute, eta ez dira egokiak baldintza hauek betetzen ez dituzten datuentzat. Diseinu konplexuko laginketetan oinarritutako datuek ordea, datuak biltzeko jarraitutako prozedura konplexua dela-eta, ez dituzte baldintza hauek betetzen. Ondorioz, ez da egokia teknika estatistiko tradizional hauek zuzenean aplikatzea mota honetako datuekin lan egiten ari garenean. Hori dela eta, doktorego tesi honen helburu nagusia eredu aurresaleen garapen prozesurako proposamen berriak egitea da, diseinu konplexuko laginketetan oinarritutako datuekin eredu aurresale onak garatu ahal izateko baliagarriak izango direnak. Zehazki, lau ekarpen egin dira doktorego tesi honetan, eta ekarpen hauetako bakoitza eredu aurresaleen garapen prozesuaren atal ezberdin bat hobetzeaz arduratzen da.

Bereziki, diseinu konplexuko laginketetan oinarritutako datuak erabilita, eredu aurresaleen parametroak estimatzeko moduak eztabaida izugarria piztu du ikertzaileen artean. Laginketa-pisuak ereduen parametroak estimatzeko erabili beharko liratekeen edo ez zalantzan jartzen dute arlo honetako ikertzaile askok, gaur egun ere. Hortaz, lehenik eta behin, simulazio-ikerketa bat egin dugu parametroak estimatzeko teknika ezberdinen jokaera konparatzeko asmoz, non teknika hauetako batzuk laginketa-pisuak erabiltzen dituzten eta beste batzuek ez, diseinu konplexuko laginketetan oinarritutako datuentzat eredu aurresaleak doitzerako orduan. Ikerketa honetan, erantzun aldagai dikotomikoen jokaera azaltzeko hain ohikoak diren erregresio logistikoko ereduetan zentratu gara. Simulazio-ikerketan erabilitako datuak, inkesta errealetako datuetan oinarrituta simulatu ditugu, diseinatu diren eszenari-

oek praktikan eskuartean izan ditzakegun arazo errealak ondo ordezkatzen dituztela bermatu ahal izateko. Simulazio-ikerketa honen emaitzetan oinarrituta, laginketa-pisuak kontuan hartzen dituzten estimazio teknikak erabiltzea gomendatuko genuke, ereduen koefizienteentzat estimazio alboragabeak lortu ahal izateko.

Gainera, doktorego tesi honetan, aldagai azaltzaile garrantzitsuenen aukeraketa egiterako orduan diseinu konplexuko laginketek duten eragina ere aztertu dugu. Zehazki, LASSO erregresio ereduetan oinarritu gara helburu hau betetzeko. Eredu mota hauetan penalizazio parametro bat aukeratu behar da aldagai azaltzaile guztien artetik erantzun aldagaiaren jokaera ondoen azaltzen dutenak aukeratu ahal izateko. Penalizazio parametro hau balidazio-teknika ezberdinak erabiliz aukeratu ohi da, besteak beste, *cross-validation* edo balidazio-gurutzatuaren teknika da erabiliena arlo honetan. Balidazio-gurutzatuaren teknika tradizionalak ordea, ez du kontuan hartzen datuak biltzeko erabilitako laginketaren izaera konplexua jatorrizko laginaren entrenamendu eta balidazio azpimultzoak sortzerako orduan. Hortaz, lan honetan, diseinuan oinarritzen den balidazio-gurutzatuaren teknika berri bat proposatu dugu (*design-based cross-validation* izena eman diogu teknika berri honi), datuak biltzeko erabili den laginketa prozesuaren izaera konplexua kontuan hartzen duena. Horretaz gain, laginketa-pisuak erabiltzea proposatzen dugu LASSO erregresio ereduak doitzerako orduan. Simulazio-ikerketa sakon bat burutu dugu proposamen hauen baliozkotasuna aztertzeko asmoz. Ikerketa honen emaitzetan oinarrituta, bi proposamen hauen arteko elkarlanak emaitzak modu esanguratsuan hobetzea dakarrela ikusi dugu, diseinu konplexuko datuekin aldagai aurresale garrantzitsuenen aukeraketa egitean.

Lehen esan bezala, doitutako eredu aurresaleak praktikan aplikatu aurretik, beharrezkoa da euren aurresateko gaitasuna aztertzea. Erregresio logistikoko ereduen aurresateko gaitasuna ROC kurba (bere ingeleseko sigletatik dator izena, *receiver operating characteristic curve*) eta AUC (*area under the ROC curve*) parametroaren bidez neurtzen da. Edonola ere, ROC kurba eta AUC parametroaren estimatzaile tradizionalak zorizko laignketa sinpleetan oinarritutako datuekin erabiltzeko sortu ziren eta ez daude prest diseinu konplexuko laginketetako datuekin lan egiteko. Hau dela eta, diseinua kontuan hartzen duten estimatzaileak proposatu ditugu lan honetan ROC kurba eta AUC-a estimatzeko laginketa-pisuak erabilita. Proposatutako estimatzaile hauen jokaera aztertzeko simulazio-ikerketa bat egin dugu. Ikerketa honen emaitzetan oinarrituta, laginketa-pisuak kontuan hartzen dituzten proposatutako estimatzaile berriak erabili beharko liratekeela ondorioztatu dugu, erregresio logistikoko ereduen aurresateko gaitasuna neurtzeko diseinu konplexuko datuetan

oinarrituta.

Doktorego tesi honetan egin dugun laugarren, eta azken, ekarpena indibiduoen sailkapenari lotuta dago erregresio logistikoko ereduen testuinguruan. Erregresio logistikoko ereduen bidez, indibiduo batek aztertzen ari garen gertaera jasateko duen probabilitatea estimatzen da. Dena dela, kasu batzuetan probabilitate hauetan oinarrituta, indibiduoaren gaineko erabakiak hartu behar ditugu eta, horretarako, indibiduo hau gertaera jasan duen edo gertaera jasan ez duen indibiduo bezala sailkatzea komeni zaigu. Erabaki hau hartu ahal izateko, mozketa-puntu bat aukeratu ohi da, eta indibiduoarentzako estimatutako probabilitatea aukeratutako mozketa-puntua baino handiagoa edo berdina bada, indibiduo hau gertaera jasan duen indibiduo bezala sailkatzen da, eta estimatutako probabilitatea aukeratutako mozketa-puntua baino txikiagoa bada aldiz, gertaera jasan ez duen indibiduo gisa sailkatu ohi da. Ahalik eta mozketa-puntu hoberena aukeratu ahal izateko (mozketa-puntu "optimoa") teknika ezberdinak proposatu dira literaturan eta teknika hauetako bakoitzak kriterio jakin bat maximizatzen du. Teknika hauek, dena den, zorizko laginketa sinpleetatik lortutako datuetan erabili izan dira orain arte, eta ondorioz, ez dituzte laginketa diseinu konplexuak kontuan hartzen. Hau horrela izanik, doktorego tesi honetan, laginketa-pisuak kontuan hartzen dituzten estimatzaile berriak proposatu ditugu mozketa-puntu optimoak estimatu ahal izateko. Estimatzaile hauen jokaera simulazio ikerketa baten bidez aztertu eta teknika tradizionalen jokaerarekin konparatu da diseinu konplexuko datuetan. Simulazio ikerketa honen emaitzetan oinarrituta, pisuak kontuan hartzen dituzten estimatzaileak erabiltzea gomendatuko genuke mota honetako datuekin lanean diharduenean.

Horrez gain, doktorego tesi honetan proposatutako metodoak inkesta ezberdinetako datu errealetara aplikatu dira eta R software estatistikoko bi pakete berritan inplementatu dira (`wlasso` eta `wROC`). R software libreko edozein erabiltzailerentzat eskuragarri daude bi pakete hauek, inkestetako datuak aztertu behar dituzten estatistikari, teknikari eta ikertzaileek tesi honetan proposatu diren metodoak modu erraz batean aplikatu ahal izan ditzaten euren eguneroko jardunean.

Laburtuz, doktorego tesi honetan hainbat ekarpen egin dira eredu aurresaleen garapen prozesua hobetzeko asmoz, diseinu konplexuko laginketetan oinarrituta jasotako datuekin lan egitean. Bereziki, ereduaren parametroen estimazioan, aldagai garrantzitsuenen aukeraketa prozesuan, ereduen aurresateko gaitasunaren estimazioan eta indibiduoen sailkapenaren arloan proposatu dira hobekuntza hauek. Proposatutako teknika guztiak simulazio ikerketen bidez balioztatu dira, eta ikerketa guzti hauetako emaitzek proposatutako teknika berrien erabilera gomendatzen

dute laginketa diseinua kontuan hartzen ez duten teknika tradizionalen ordez. Proposamen hauek R paketean daude eskuragarri, eta modu honetan ekarpen garrantzitsua egin zaio gizarteari, erabiltzeko errazak diren tresnen bidez metodo hauek inplementatu ahal izatea baimenduz.

Summary

Complex survey data are being increasingly used nowadays, especially in the context of Official Statistics. In this framework, complex survey data are usually obtained by sampling the population of interest for the survey, following some particular complex sampling design. This sampling process may be carried out in one or more sampling stages, for which the combination of techniques such as stratification and clustering is very common practice. One of the most special characteristics of complex sampling data, in comparison to data obtained based on simple random sampling, is that each individual in the population has a probability (different from 0) of being included in the sample. These inclusion probabilities are not equal for all the individuals in the population. Then, each individual that finally ends up in the sample is assigned a sampling weight, calculated as the inverse of its inclusion probability, which indicates the number of units from the finite population that is represented by this individual in the sample. Complex sampling designs are a good way of obtaining better representations of minority groups in the population and can also help reduce the cost of data collection. As a result, they are becoming more and more relevant in various fields, such as social and health sciences, among others.

Currently, complex survey data are being widely used to fit prediction models, among other purposes. The relevance of prediction models in numerous areas is undeniable. The main goal of prediction models is to make predictions on a response variable of interest based on the information observed for several explanatory variables or covariates. In order to end up with good prediction models that can be applied in daily practice, it is necessary to consider taking several steps. For exam-

ple, the selection of the most relevant covariates and the analysis of the predictive performance of the fitted model are some important key points in the development process of prediction models.

However, most of the statistical techniques developed for fitting good prediction models are based on a number of assumptions, such as the units that are being analyzed are independent and identically distributed (iid). The problem comes from the fact that complex survey data, due to the way in which it was collected, do not satisfy iid conditions. Hence, the straightforward application of traditional statistical techniques, including those techniques developed for fitting prediction models, is not appropriate in this context. Therefore, the main objective of this Ph.D. thesis is to make new proposals on the development process of prediction models, so that they can be used to develop prediction models for complex survey data. In particular, four contributions are made, each of them regarding a different part of the development process of prediction models.

In the first place, the estimation of model parameters in the context of complex survey data has been the source of a large debate in the literature. Specifically, whether sampling weights should or should not be considered in the estimation process of model parameters has generated many doubts among statisticians. Thus, a simulation study is carried out in order to compare the performance of several coefficient estimation techniques, some of which consider the sampling weights and some which do not, for estimating the prediction model coefficients based on complex survey data. In particular, this study focuses on logistic regression models for dichotomous response variables. Data is generated based on real surveys in order to design more realistic scenarios. The results suggest the use of weighted estimation techniques in order to obtain unbiased model coefficient estimates.

In addition, in this Ph.D. thesis, we also analyze the impact of complex sampling designs on the selection of the most relevant predictor variables. In particular, we lay on LASSO regression models for this purpose. In LASSO regression models, a tuning parameter must be selected to obtain a subset of the most important variables that best explain the behavior of the response variable. This tuning parameter is usually selected based on validation techniques such as cross-validation. However, the traditional cross-validation does not consider the complex sampling design in the way in which it generates training and test sets. Thus, we propose a new design-based cross-validation technique that accounts for the complex nature of the sampling process. Moreover, we also suggest considering sampling weights for estimating LASSO regression model coefficients. An extensive simulation study

has been carried out in order to analyze the validity of these proposals. The results suggest that the combination of these two proposals results in a considerable improvement in the variable selection process with complex survey data.

Before implementing prediction models in practice, their predictive performance should be analyzed. In the context of logistic regression, the predictive performance of the models is commonly quantified in terms of discrimination ability and calibration. In this Ph.D. thesis, we focus our attention on the discrimination ability, which is usually measured by means of the receiver operating characteristic (ROC) curve and the area under that curve (AUC). However, the traditional ROC curve and AUC estimators are not thought to be applied to complex survey data and, hence, do not account for complex sampling designs. Therefore, we propose new design-based estimators considering sampling weights. The performance of the proposed estimators is analyzed by means of a simulation study. The results suggest that the proposed weighted estimators should be used in order to estimate the discrimination ability of logistic regression models in the context of complex survey data.

The fourth contribution of this Ph.D. thesis is related to the classification of individuals in the context of logistic regression models. When logistic regression models are used for making predictions, a predicted probability of event can be estimated for an individual. However, in some cases, we should decide whether we classify this unit as event or non-event, based on its estimated predicted probability. In this context, a cut-off point is usually selected, and if the predicted probability is greater or equal to the selected cut-off point, the unit is classified as event, while if the predicted probability is lower than that cut-off point, then the unit is classified as non-event. Several techniques are proposed in the literature to select the optimal cut-off point that maximizes a particular criterion. However, those techniques are thought to be applied to simple random samples, and hence, they do not consider complex sampling designs. In this Ph.D. thesis, we propose new weighted estimators for estimating optimal cut-off points, and the performance of those estimators is compared to the traditional unweighted ones by means of a simulation study. The results suggest the use of weighted estimators in the context of complex survey data.

In addition, the methods proposed in this Ph.D. thesis have been applied to real survey data and implemented in two R-packages (`wlasso` and `wROC`) that are freely available for any R user so that survey statisticians can easily apply those proposals in their particular surveys in daily practice.

In summary, in this Ph.D. thesis, we make several contributions to the improvement of the development process of prediction models for complex survey data, in

particular, regarding the estimation of the model, variable selection, estimation of the discrimination ability, and the classification of individuals. All these proposals have been validated by means of simulation studies, the results of which recommend the use of the proposed methods against the traditional ones. The new methodological proposals have been incorporated into two R-packages, providing society with accessible tools for easy implementation of these proposals.

CHAPTER 1

Introduction

To explain the context in which this Ph.D. thesis emerged, we need to go back to September 2017. At that time, the Official Statistics Basque Office EUSTAT (Euskal Estatistika Erakundea - Instituto Vasco de Estadística, hereinafter EUSTAT) contacted the Biostit Research Group due to their experience in the development of prediction models to help them with the modeling of a data set, specifically, the data set obtained from the Survey on the Information Society in Companies (ESIE, due to its Spanish acronym). Biostit Research Group hired the present Ph.D. candidate to work as a research technician in this collaboration.

When we (the Ph.D. candidate and the supervisors of this dissertation) started working with the ESIE survey data set, we realized that the sample in which the survey was carried out was obtained following a particular complex sampling design. Given that we had neither previous experience nor knowledge related to complex survey data, we first started reviewing the literature on the analysis of this type of data. We realized that the conditions that are assumed in order to apply the traditional statistical techniques to develop prediction models were not satisfied for

the data we were handling, and therefore, we had to question all the steps we would usually follow when developing prediction models for *traditional* data.

Hence, this Ph.D. thesis covers two important fields within statistics: prediction models and complex survey data. Therefore, we will start this chapter by introducing these two topics. In Section 1.1, prediction models are introduced, while complex survey data are described in Section 1.2. In Section 1.3, we define the objectives we have addressed in this thesis, and finally, in Section 1.4, we describe the organization of the subsequent chapters of this dissertation.

1.1 Prediction models

The relevance of prediction models is undeniable in many areas nowadays. Health sciences, biology, ecology, official statistics, meteorology, and finance are just some examples of a large number of fields in which prediction models play an essential role. Prediction models are used to explain the behavior of the characteristic of interest, response variable, or outcome by means of a set of explanatory variables or covariates. One of the main goals of prediction models is to make predictions for unobserved data based on the information and patterns that have been previously observed. For example, the evolution that a patient may have can be estimated considering the evolution that other patients with similar characteristics have previously had. There are different types of prediction models, and depending on the distribution of the response variable and its association with the covariates, the most appropriate model should be chosen to describe the relationship between the response variable and the covariates. Throughout this document, in particular, we will consider linear regression models (for response variables following a Gaussian distribution) and logistic regression models (appropriate for response variables following a Bernoulli distribution).

Prediction models are widely used as a tool for decision-making. For example, in finance, they can be useful for predicting loan defaults (Li et al. 2022); in ecology, particularly in fisheries, they are often used to make conservation decisions (Guisan et al. 2013, Li et al. 2020); medicine is another field in which prediction models are widely implemented as a support for decision-making, where, they can be helpful for deciding whether a patient should be admitted to an intensive care unit or not, among other purposes (Arostegui et al. 2019).

Given the impact of these models in daily practice, it is essential to ensure that the prediction models are valid and useful before implementing them in new

data, especially if the goal is to use them as support to make decisions. Thus, the development process and validation of models before making predictions for new data are key points when working in this context. Steyerberg (2008) and Steyerberg and Vergouwe (2014) offer interesting guidelines to end up with good prediction models that can be effectively applied in practice. Several steps need to be followed for this purpose. First, the most appropriate model that will properly fit our data needs to be selected. Another important step is a proper selection of the variables to be considered in the model. How to treat these covariates and how to deal with missing values (in case they are present in the data set being managed) should be considered during the process of developing the prediction models. The predictive performance of the fitted models also needs to be checked before implementing them in new data sets, particularly if they are going to be used as support for decision-making. Finally, the final model needs to be validated before being implemented into new data sets by means of internal and external validation techniques.

Different techniques to handle the development process of prediction models are available for independent and identically distributed (iid) data (see, e.g., Steyerberg (2008)). But what happens if our data violate iid conditions? This dissertation focuses on this context.

1.2 Complex survey data

The history of complex survey data dates back to a little over a century ago, when Kaier (1895) defined the first theoretical basis for obtaining representative samples of a population (for more information on the origins of the sampling theory and the social situation of that time we recommend Lie (2002)). In the following years, the basis of randomization (Fisher 1992) and different sampling techniques were proposed, and the basis of the theory on the estimation of population statistics and their variance estimation were set (see, e.g., Cochran (1977), Horvitz and Thompson (1952), Kish (1965), Neyman (1934)). Nowadays, complex survey data have become an important tool for society, being the basis of most large-scale national surveys (see, e.g., Fisher et al. (2020)), which cover a wide range of social topics, including but not limited to: health (see, e.g., National Health and Nutrition Examination Survey (NHANES)[1] carried out in the United States, the European health interview survey[2], or the Hortega Study in Spain (Tellez-Plaza et al. 2019)); labor force or pop-

[1] https://www.cdc.gov/nchs/nhanes/about_nhanes.htm
[2] https://ec.europa.eu/eurostat/web/microdata/european-health-interview-survey

ulation activity surveys (such as the one conducted by Eurostat[3] or by the Spanish National Statistics Institute (INE)[4]); or income and economic related surveys used to analyze, among other purposes, the poverty indicators among the inhabitants (such as the Living Conditions Monitoring Survey conducted in Zambia[5] or the Poverty and Social Inequalities Survey[6] carried out by the Basque Government).

Complex survey data are obtained by carrying out the survey on a representative sample of the population of interest, which in this context is assumed to be finite. These samples are usually drawn following complex sampling designs, such as stratification, clustering, or a combination of them in different stages of the sampling process. In order to compensate for the unbalanced sample that is usually obtained based on these sampling schemes, a sampling weight is usually assigned to each sampled observation, which indicates the number of elements it represents in the population.

Due to their special characteristics, complex survey data do not satisfy iid conditions (see, e.g., Skinner et al. (1989)). Therefore, the analysis of this kind of data has been the source of a great deal of discussion. In particular, the way that prediction models should be fitted based on this kind of data has generated a large discussion among researchers and statisticians. Fuller (1975) and Binder (1981; 1983) proposed accounting for complex sampling designs when fitting regression models. However, the researchers of that time began having serious doubts about the best way of proceeding with the estimation of the models, wondering whether the sampling weights were needed in this context.

In order to introduce the problem very briefly, this discussion can be seen as a large debate of two different perspectives for facing the same question. These approaches are usually known as model-based and design-based perspectives in the literature (Binder and Roberts 2009, Chambers and Skinner 2003). The researchers with the model-based point of view claim that if the prediction model is well specified, then sampling weights are not necessary, and thus, the probability distribution induced by the sampling design is usually ignored by these researchers. This perspective can be seen as a theoretical way of facing the problem, given that it assumes the existence of a prediction model that perfectly explains the relationship between

[3] https://ec.europa.eu/eurostat/web/microdata/public-microdata/labour-force-survey

[4] https://www.ine.es/dyngs/INEbase/en/operacion.htm?c=Estadistica_C&cid=1254736176918&menu=ultiDatos&idp=1254735976595

[5] https://catalog.ihsn.org/catalog/7105/study-description

[6] https://www.euskadi.eus/gobierno-vasco/-/informacion/encuesta-de-pobreza-y-desigualdades-sociales-epds/

the outcome and a set of covariates that are available (also known as the *super-population* model, where the *superpopulation* is the infinite population from which the finite population on which the survey is conducted is derived (Chambers and Skinner 2003)) and refuses the adequacy of fitting a model that does not satisfy this condition. In contrast, the researchers adopting the design-based perspective warn that biased estimates may be obtained if the sampling design is ignored in the estimation process. Hence, they recommend the use of sampling weights in order to correct this bias. This point of view is more focused on the real needs of daily practice, given that the researchers that adopt this perspective note that not always exists a theoretical model that perfectly explains the behavior of the response variable based on the available covariates, and they are more inclined to adjust the best possible model according to the circumstances of each situation.

As an example of this discussion, Brewer and Mellor (1973) published an entertaining paper in the form of a dialogue between a survey statistician and a mathematical statistician, which makes it clear that the questions and issues related to how to work with this type of data are far from being trivial. Some years later, Smith (1988) gave a name to the question asked by many researchers when working with complex survey data by publishing an article entitled *"To weight or not to weight, that is the question"*. This debate is still ongoing today, as we will discuss throughout this dissertation, and many researchers continue to hesitate about how to treat this type of data. As an example, it is worth reading the title of an article published by Gelman (2007) that needs no further explanation: *"Struggles with survey weighting and regression modeling"*. In the paper, the author defines survey weighting as "a mess". This discussion will be analyzed in more detail in the subsequent chapters.

The point we aim to emphasize here is that, due to the debate between these two approaches, the researchers who need to work with survey data have serious doubts about which one would be the best way for them to proceed with their analysis. Even though Heeringa et al. (2017) states that the more practical design-based perspective is the one that most survey statisticians currently adopt, there still are researchers that question this point.

Lumley (2010) points out that one of the potential reasons why most researchers may have adopted the model-based perspective in the past was the lack of software available to do the analysis from the design-based perspective. As the same author states, this is not a very important problem nowadays since most of the frequently used statistical programs such as R (by means of the `survey` package (Lumley 2020)), SAS (Lewis 2016) or Stata are prepared to perform those analyses (see, e.g., Heeringa

et al. (2017) for a summary of the available statistical software to work under the design-based perspective).

However, the Ph.D. candidate and her advisors would like to add another reason why some researchers may still be in favor of avoiding performing the analysis from a design-based point of view when possible. The analysts who adopt the model-based perspective have available all the traditional statistical techniques that are proposed to develop good prediction models to iid data. In contrast, there is a considerable lack of proposals to develop good prediction models following the design-based approach. As stated above, one of the most discussed topics in the context of complex surveys is the effect of the sampling design on the estimation of model parameters (see, e.g., Binder and Roberts (2009), Holt et al. (1980), Lumley and Scott (2017), Reiter et al. (2005), Scott and Wild (1986; 2002)). Nevertheless, beyond the estimation of the model parameters, complex survey data have shown to have a great impact on the development of prediction models, and numerous advances have been made in the last years in this field, including the following ones, among others: Lumley and Scott (2015) proposed new design-based estimators for estimating two widely used parameters for model selection, Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC), considering the sampling design; focusing on the evaluation of model performance, Archer et al. (2007) and Lumley (2017) proposed goodness-of-fit tests that consider complex sampling designs to analyze the calibration of the models fitted to complex survey data; in the context of the discrimination ability, Yao et al. (2015) proposed a modification of the Mann-Whitney U-statistic in order to consider the sampling design to estimate the AUC of the models. Nevertheless, we believe that there is still a lack of design-based techniques to develop good prediction models, and this fact can make some researchers avoid adopting the design-based perspective. Therefore, the research work performed in the framework of this thesis was defined in order to provide new tools and make improvements for developing good prediction models for complex survey data from the design-based perspective.

1.3 Objectives of the thesis

Coming back to our particular situation in which we signed a collaboration agreement with EUSTAT for the modeling of the ESIE survey, we realized we would have problems responding to their petitions due to the lack of tools to develop good, valid, and useful prediction models that consider the complex sampling design of the data.

Later on, more questions related to other datasets were raised based on different complex sampling surveys that our collaboration with EUSTAT required answering. Specifically, the collaboration has continued with two other surveys besides the ESIE survey: the Population in Relation to Activity (PRA) survey and the Innovation survey. This EUSTAT collaboration has given rise to numerous new questions that have led us to open a new research line, the results of which have begun to be visible in this thesis and will continue to be reflected in future works. We noticed that it would be necessary to continue researching this line, not only to be able to respond to the petitions of EUSTAT properly but also because we realized that, given the increasing interest of society in the analysis of survey data, many researchers would have the same problems as us. Therefore, we set the goals of this thesis in order to resolve the questions, doubts, and problems we came across when we began to work in this collaboration, and thus, all the questions addressed in this work are fully motivated by the real needs and gaps we found when we started trying to fit prediction models to complex survey data in practice.

In summary, the main objective of this thesis is to make new proposals for developing good prediction models with complex survey data from the design-based perspective, as well as to provide computational tools in order for other researchers to use the methods we propose in this thesis. Among the different types of prediction models, we have focused, in particular, on linear and logistic regression models. Within this main objective, four specific goals have been defined, each of them related to a different section of the development process of prediction models. Specifically, in this Ph.D. thesis, we have focused on (1) the estimation of prediction model parameters, (2) variable selection, (3) analysis of the predictive performance, and (4) classification of individuals. These four goals are defined in more detail below.

Goal 1. Estimation of model parameters.

The first goal of this thesis is to analyze the performance of different existing estimation methods, some of which consider the sampling design and others do not, for estimating prediction model coefficients. Specifically, we focus on the framework of logistic regression models for dichotomous response variables. The hypothesis we aim to check is that the use of sampling weights may be crucial when fitting logistic regression models.

When we started working with complex survey data, the first questions that came to our minds were related to the estimation methods for fitting prediction

models. We realized that there are plenty of works in the literature addressing this issue which are commonly carried out under two different situations:

(a) Theoretical works and simulation studies based on artificial data sets generated based on previously established known theoretical models, but which do not work under realistic scenarios with real surveys (see, e.g., Pfeffermann (1993), Scott and Wild (1986)).

(b) Studies with fully practical objectives, in which the performance of different estimation methods for estimating the model parameters are compared among them, but the true parameters are not known (see, e.g., Chambless and Boyle (1985), Masood et al. (2016)).

However, none of these papers completely solved our concerns, given that our main focus was analyzing the impact of the issues of the above-mentioned studies in real survey data but under controlled scenarios in which the theoretical values of the parameters were known. Therefore, we aim to conduct a simulation study based on real survey data to address this goal.

Goal 2. Variable selection.

The second goal of this thesis is to analyze the impact of complex sampling designs on the variable selection process of prediction models and to propose new design-based methods for this purpose. In particular, we focus on the variable selection by means of the Least Absolute Shrinkage and Selection Operator (LASSO) regression, which allows the selection of a subset of predictor variables that best describes the behavior of the response variable based on a tuning parameter. In addition, both linear and logistic regression are considered (in particular, this is the only goal we address for linear regression models in this thesis). We expect the new design-based proposal to make a difference in the selection of variables based on either linear or logistic LASSO regression models.

Goal 3. Analysis of the predictive performance.

The third goal of this dissertation is to make new proposals for estimating the predictive performance of prediction models considering complex sampling designs. In particular, we focus on the logistic regression framework, for which discrimination ability may be used to analyze the predictive performance of the models. The discrimination ability of logistic regression models is usually measured by means of the Receiver Operating Characteristic (ROC) curve and

the area under it (AUC). We aim to propose new estimators that account for complex sampling designs to estimate the ROC curve and AUC of the models. We expect these new proposals to be unbiased estimators of the ROC curve and AUC, and to outperform the traditional estimators, which do not account for complex sampling designs.

Goal 4. Classification of individuals.

The fourth and last goal of this dissertation is to make new proposals that consider complex sampling designs for selecting the optimal cut-off points for individual classification. Remaining in the context of logistic regression models, the probability of event can be estimated for each individual based on the fitted model. However, it is sometimes interesting to classify these individuals as individuals with or without the event of interest. For example, a doctor may obtain a probability indicating the risk that a patient will have to be admitted to the Intensive Care Unit (ICU) but may want to make a decision about whether or not he or she should ultimately admit that patient to the ICU. The individuals can then be classified as events or non-events based on their probability of event. Usually, if the probability is greater than a pre-specified cut-off point, the individual is classified as event and, otherwise, as non-event. In this context, different methods can be applied in order to select the optimal cut-off points for individual classification. However, the traditional methods are not thought to be applied to complex survey data, and hence, they do not consider complex sampling designs. Thus, the hypothesis is that these new design-based estimators would be a better option to estimate optimal cut-off points than the traditional ones.

The four goals described above have been addressed throughout this Ph.D. thesis thanks to the financial contract offered by Biostit Research Group (IT620-13), the Predoctoral Grant PIF18/213 of the University of the Basque Country (UPV/EHU), and the financial support of MATHMODE: Group on Applied Mathematical Modeling, Statistics and Optimization (IT1294-19, IT1456-22), S3M1P4R (PID2020-115882RB-I00) and MTM2016-74931-P.

1.4 Organization of the subsequent chapters

This section describes the organization of the following chapters, in which the goals described in Section 1.3 have been addressed.

Chapter 2 mathematically defines the basic notation and definitions of fundamental concepts related to complex survey data that will be necessary to understand the rest of the dissertation. In addition, we introduce the methodological background to estimate prediction models for complex survey data from the design-based perspective. In particular, throughout this document, we work with linear and logistic regression models, so we introduce the design-based approach to fit these two types of models.

In Chapter 3, we introduce the two real survey data sets that have motivated this work and are continuously referenced throughout the document. In particular, the Survey on the Information Society in Companies (ESIE) and the Population in Relation to Activity (PRA) Survey are described. Both surveys have been designed and collected by EUSTAT.

Chapter 4 offers an extensive bibliographical review and all the details addressing the first goal of this dissertation regarding the estimation of logistic regression model parameters. In this chapter, we conduct a simulation study based on real survey data. With this simulation study, we aim to analyze the impact of complex sampling designs when estimating logistic regression model parameters under realistic but controlled scenarios, in which the true parameter values are known. We generate simulated data based on ESIE and PRA surveys and sample these populations by mimicking the corresponding true sampling designs. Since the outcomes of these two real surveys are dichotomous response variables, we compare the performance of different estimation methods for estimating logistic regression parameters. In addition, beyond the simulation study, we also compare the estimates obtained by means of different estimation methods in the true real surveys. The work presented in this chapter has been accepted for publication:

> Iparragirre, A., Barrio, I., Aramendi, J., & Arostegui, I. (2024). Estimation of logistic regression parameters for complex survey data: simulation study based on real survey data. *SORT - Statistics and Operations Research Transactions*, in press.

In Chapter 5, we address the second goal of this dissertation by analyzing the

impact of complex sampling designs on the variable selection of linear and logistic LASSO regression models. On the one hand, the impact of the sampling weights in the estimation of LASSO regression models is analyzed. On the other hand, new methods that consider complex sampling designs for selecting the optimal tuning parameter for LASSO regression models fitted to complex survey data are proposed. The performance of different methods is compared by means of a simulation study. The work presented in this chapter has been published in the following scientific paper:

> Iparragirre, A., Lumley, T., Barrio, I., & Arostegui, I. (2023). Variable selection with LASSO regression for complex survey data. *Stat*, 12(1), e578.

In Chapter 6, we address the third goal of this dissertation. In particular, we analyze the impact of complex sampling designs on the estimation of the discrimination ability of logistic regression models fitted to complex survey data. Two new estimators that consider sampling weights are proposed for estimating the ROC curve and AUC of the models. The performance of the proposals is validated and compared to the traditional estimators by means of a simulation study. The proposed estimators are also applied to ESIE survey data. The work presented in this chapter has been published in the following scientific paper:

> Iparragirre, A., Barrio, I., & Arostegui, I. (2023). Estimation of the ROC curve and the area under it with complex survey data. *Stat*, 12(1), e635.

In Chapter 7, we face the fourth goal of this thesis, in which we propose new weighted estimators for estimating the optimal cut-off points for individual classification in the context of logistic regression models. A simulation study is conducted in order to analyze the performance of the proposed methods, which are compared to the traditional unweighted estimators. The proposed estimators are also applied to ESIE real survey data. The work described in this chapter has been published in the following scientific paper:

> Iparragirre, A., Barrio, I., Aramendi, J. & Arostegui, I. (2022) Estimation of cut-off points under complex-sampling design data. *SORT-Statistics and Operations Research Transactions*, 46(1), 137–158.

All the proposals developed in the previous chapters have been programmed in the statistical software R. Specifically, two packages have been created and set publicly available to any user of R. The `wlasso` package allows to carry out the selection of variables by means of the proposals described in Chapter 5 (**Goal 2**). The `wROC` package can be used to estimate the ROC curve and the AUC of logistic regression models fitted to complex survey data and to estimate the optimal cut-off points for the classification of individuals in this context, as described in Chapter 6 (**Goal 3**) and Chapter 7 (**Goal 4**), respectively. In Chapter 8, these two packages are described in detail.

Finally, we close this document in Chapter 9 with a discussion regarding the main conclusions and limitations of this dissertation, further research topics that we aim to address in the future in relation to the matter related to the development of prediction models to complex survey data and the main contributions emerged from this Ph.D. thesis.

CHAPTER 2

Basic notation of complex survey data and background of design-based prediction models

This work addresses two important statistical fields that need to be introduced in detail: complex survey data and prediction models. The goal of this chapter is to introduce these two fields to the reader and set the basic notation that will be used throughout the rest of the document.

In particular, in Section 2.1, we set the notation related to complex surveys, and we describe the two sampling designs that are considered throughout the document: the one-stage stratified sampling and the two-stage stratified cluster sampling. It should be noted that the context of complex surveys is very broad, and the goal of this chapter is not to make a summary of it but to define the basic terms related to complex surveys that are necessary to understand the concepts considered and the methods proposed in this work. Therefore, in this chapter, we focus on describing only those terms that will be used in the following chapters, while many important aspects of complex survey data (which include other kinds of sampling designs, post-

stratification, or the modification of sampling weights to account for non-responses, among others) are not explained here, as they have not been treated in this work. For more information on these topics, we recommend Cochran (1977), Heeringa et al. (2017), Kalton (1983), Kish (1965), Kott (2012), Little and Vartivarian (2003) and Särndal et al. (2003).

In Section 2.2, we describe the existing methodological background to fit prediction models to complex survey data from the design-based framework. In particular, throughout this document, we work with linear and logistic regression models. Thus, in this section, we first briefly introduce these models traditionally defined for iid data. The main objective of this section is to familiarize the reader with the estimation of predictive models from the design perspective, and we thus introduce the methodological background for fitting linear and logistic regression models in this framework. It should be noted that the methodological background described in this section will be necessary to understand the methodology implemented and proposed in the following chapters. However, some fundamental concepts in the development of prediction models (such as variable selection or the predictive ability of the fitted models) have been omitted from this chapter and are explicitly addressed in the corresponding chapter.

2.1 Basic notation of complex survey data

The objective of this section is to define the basic concepts related to complex survey data and to establish the notation that will be used throughout the rest of the document. Specifically, we mathematically define the basic elements of this type of data, which include the finite population, complex sampling designs, inclusion probabilities, and sampling weights.

A survey is usually conducted with one objective in mind: to analyze a particular characteristic in a specific population. For example, we may be interested in analyzing the employment status of inhabitants in a certain city, in the amount of money invested by companies in a particular country in research, development, and innovation, or in studying the evolution of pain in patients in a region who are receiving a particular treatment. The first task is usually to correctly define the population of interest where we want to carry out the research. In the framework that we have considered in this work, we work with a finite population for which all the individuals that are part of it are identifiable in some way. For example, the inhabitants of a city can be identified by the registration documents of that city,

the companies of a country are registered by means of a tax identification number, and the patients of a region who are receiving that treatment are registered by the health system of that region. Hereinafter, we will denote as U the finite population of interest and N its size.

In practice, it is often unfeasible and/or inefficient to survey all individuals in the population, and it is decided to take a sample representing the population in which to conduct the survey instead. The simplest way to sample the population is to choose individuals at random. Let n be the number of individuals to be sampled. In this way, each individual has an identical probability of n/N of being sampled. This sampling design is known in the literature as simple random sampling. However, in most cases, survey statisticians decide to use other types of sampling in order to obtain more efficient and/or economically more feasible samples by implementing more complex sampling techniques. More information about survey sampling techniques can be found in Cochran (1977), Kish (1965) and Särndal et al. (2003). A good summary can also be found in Kalton (1983).

In particular, in this work, we focus on probability-based sampling designs. In this type of sampling, each unit of the population is assigned a probability (different from zero) to be included in the sample (also known as "inclusion probability"), and these probabilities can be different for each population unit. Methods such as stratification or clustering can be applied at one or several stages of the sampling process. The sampled individuals are assigned a sampling weight, which is defined as the inverse of their inclusion probability (thus, can also be denoted as "inverse probability weights") and indicates the number of units that this sampled observation represents in the finite population. Throughout this document, we work with two different complex sampling designs. In particular, in Section 2.1.1, we describe the one-stage stratified sampling design. Briefly, this sampling design consists in defining several non-overlapping population subsets (denoted as strata) and sampling individuals by simple random sampling in each of the strata. In Section 2.1.2, the two-stage stratified cluster sampling design is described, based on which, in the first stage, some clusters or groups of individuals are selected to be included in the sample from each stratum, and in the second stage, only a number of individuals are sampled by simple random sampling process from each cluster selected in the first stage to be finally included into the sample. We proceed to describe both sampling designs in detail below.

2.1.1 One-stage stratified sampling

In this section, we describe the one-stage stratified sampling process. In order to sample the finite population based on this design, we need to follow the next steps.

First, the finite population U is divided into several strata, which are usually defined based on the information on a variable or a combination of some of them. These strata are mutually excluding subsets of the population, that is,

$$U = \bigcup_{h=1}^{H} U_h, \quad \text{where} \quad U_h \cap U_{h'} = \varnothing,\ \forall h, h' = 1, \ldots, H\ :\ h \neq h', \tag{2.1}$$

being H the total number of defined strata. Let us denote as N_h the size of stratum h, $\forall h = 1, \ldots, H$, where,

$$\sum_{h=1}^{H} N_h = N. \tag{2.2}$$

Then, a previously specified number of individuals is randomly sampled from each population stratum. Let us denote as n_h the number of units to be sampled from U_h, $\forall h = 1, \ldots, H$ and form the subset S_h. Then, each individual $i \in U_h$ has the inclusion probability π_i, where,

$$\pi_i = P(i \in S) = P(1_{S_h}(i) = 1) = \frac{n_h}{N_h}, \quad \forall i \in U_h,\ \forall h = 1, \ldots, H, \tag{2.3}$$

where,

$$1_{S_h}(i) = \begin{cases} 1 & \text{if} \quad i \in S_h, \\ 0 & \text{if} \quad i \notin S_h, \end{cases} \qquad \forall h = 1, \ldots, H. \tag{2.4}$$

Once n_h elements have been selected from U_h they form the subset $S_h \subset U_h$, $\forall h = 1, \ldots, H$. Finally, the sample S is constructed by joining all the subsets S_h,

$$S = \bigcup_{h=1}^{H} S_h, \quad \text{where} \quad S_h \cap S_{h'} = \varnothing,\ \forall h, h' = 1, \ldots, H\ :\ h \neq h', \tag{2.5}$$

and the total sample size is denoted as $n = \sum_{h=1}^{H} n_h$.

One of the major advantages of stratified sampling is that a different proportion of individuals can be sampled from each stratum, oversampling some of them so that a sufficiently large sample size can be ensured in each subset. Thus, a sampling weight is assigned to each sampled unit in order to solve the imbalance that this act would imply. These sampling weights are defined as the inverse of inclusion

probabilities, and they indicate the number of units that each sampled observation represents in the finite population. For each unit i sampled from U_h, the corresponding sampling weight is calculated as follows:

$$w_i = \frac{1}{\pi_i} = \frac{N_h}{n_h}, \quad \forall i \in S_h, \tag{2.6}$$

or equivalently, in a more general way, $\forall i \in S$:

$$w_i = \sum_{h=1}^{H} 1_{S_h}(i) \cdot \frac{N_h}{n_h}, \quad \forall i \in S. \tag{2.7}$$

Note that, following the notation set above, the sum of the sampling weights is equal to the population size,

$$\begin{aligned} \sum_{i \in S} w_i = \sum_{i \in S} \sum_{h=1}^{H} 1_{S_h}(i) \cdot \frac{N_h}{n_h} = \sum_{h=1}^{H} \sum_{i \in S} 1_{S_h}(i) \cdot \frac{N_h}{n_h} = \sum_{h=1}^{H} \sum_{i \in S_h} \frac{N_h}{n_h} = \\ = \sum_{h=1}^{H} \frac{N_h}{n_h} \cdot n_h = \sum_{h=1}^{H} N_h = N. \end{aligned} \tag{2.8}$$

A graphical example of the stratified simple random sampling scheme can be found in Figure 2.1. In this figure, it can be seen how the finite population (depicted as the rectangle painted in dark blue) is partitioned into different strata (which are defined by means of gray lines). Two of these population strata are zoomed in, and all the individuals in each of them are shown. On the one hand, the greatest stratum of both that are zoomed in (which is depicted on the left side, let us indicate this stratum as U_{h^*}) is formed by $N_{h^*} = 20$ individuals, from which $n_{h^*} = 4$ of them are sampled. On the other hand, from the smallest stratum depicted on the right side (which will be indicated as $U_{h^{**}}$), $n_{h^{**}} = 2$ out of $N_{h^{**}} = 4$ individuals are sampled. Note that when sampling 4 out of 20 individuals or 2 out of 4, the sampling proportions of these strata are not equal, and thus, the inclusion probabilities for the units in these strata are not the same. Following eq. (2.3), the inclusion probability for the units in U_{h^*} is:

$$\pi_i = \frac{4}{20} = 0.2, \quad \forall i \in U_{h^*}, \tag{2.9}$$

while the inclusion probability for the units in $U_{h^{**}}$ is calculated as follows:

$$\pi_i = \frac{2}{4} = 0.5, \quad \forall i \in U_{h^{**}}. \tag{2.10}$$

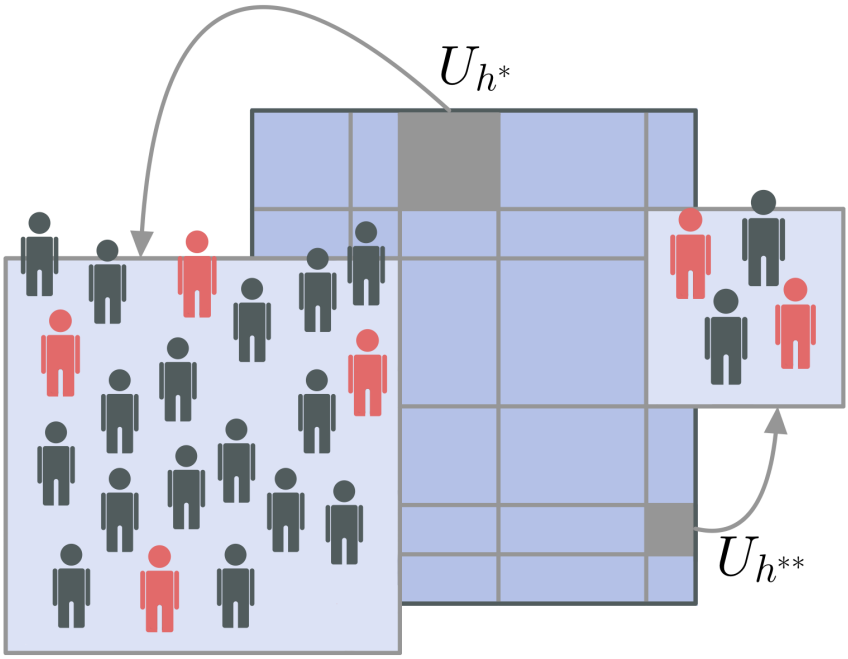


Figure 2.1: Stratified simple random sampling without replacement. The dark blue rectangle indicates the finite population. The gray lines separate the strata. The sampled individuals are represented in red.

Therefore, the sampling weights assigned to these sampled units also differ. Following eq. (2.6), the sampling weight for the 4 units sampled from U_{h^*} is the following:

$$w_i = \frac{20}{4} = 5, \quad \forall i \in S_{h^*}, \tag{2.11}$$

indicating that each of them represents 5 units as themselves in the finite population (given that since they belong to the same stratum, they are assumed to be similar). In the same way, the sampling weight for the 2 units sampled from $U_{h^{**}}$ is,

$$w_i = \frac{4}{2} = 2, \quad \forall i \in S_{h^{**}}. \tag{2.12}$$

Finally, note that the sum of the weights corresponding to the units sampled from these two strata gives the total number of units in the corresponding finite popula-

tion, i.e.,

$$\sum_{i \in S_{h^*}} w_i + \sum_{i \in S_{h^{**}}} w_i = 4 \cdot 5 + 2 \cdot 2 = 20 + 4 = 24. \tag{2.13}$$

In this example, we only show the sampling process in two of them for illustration purposes. However, it is important to note that, in order to properly obtain a representative sample for the whole finite population, the sampling must be done one by one in each of the strata. In that case, the sum of all the sampling weights of the sampled units must result in the total population size N, as shown in eq. (2.8).

2.1.2 Two-stage stratified cluster sampling

In this section, we describe the two-stage stratified cluster sampling process, which, as its name indicates, is carried out in two stages. In the first place, as described for the one-stage stratified sampling in eq. (2.1), the finite population U is partitioned into H mutually excluding strata,

$$U = \bigcup_{h=1}^{H} U_h, \quad \text{where} \quad U_h \cap U_{h'} = \varnothing, \, \forall h, h' = 1, \ldots, H \, : \, h \neq h'. \tag{2.14}$$

In addition, in each population stratum $U_h, \forall h = 1, \ldots, H$ individuals are grouped into non-overlapping clusters, that is,

$$U_h = \bigcup_{\alpha=1}^{A_h} U_{h,\alpha}, \quad \text{where} \quad U_{h,\alpha} \cap U_{h,\alpha'} = \varnothing, \, \forall \alpha, \alpha' = 1, \ldots, A_h \, : \, \alpha \neq \alpha', \tag{2.15}$$

being A_h the total number of clusters in U_h, $\forall h = 1, \ldots, H$ and $N_{h,\alpha}$ the population size of each cluster $U_{h,\alpha}$.

In the first stage of the sampling, a pre-specified number of clusters a_h (being $a_h \leq A_h$, $\forall h = 1, \ldots, H$) are randomly selected from each stratum. The units (in this case, clusters) that are sampled in the first stage of the sampling process are usually also known as primary sampling units (PSU). Let us define the next indicator function:

$$1_h(\alpha) = \begin{cases} 1 & \text{if the cluster } U_{h,\alpha} \text{ is selected in the first stage,} \\ 0 & \text{otherwise.} \end{cases} \tag{2.16}$$

Let $\pi_{h,\alpha}$ be the inclusion probability for the population cluster $U_{h,\alpha}$, to be selected.

Then, note that,

$$\pi_{h,\alpha} = P(1_h(\alpha) = 1) = \frac{a_h}{A_h}, \quad \forall \alpha = 1, \ldots, A_h, \; \forall h = 1, \ldots, H. \tag{2.17}$$

In the second stage of the sampling, a given number of individuals is sampled from each cluster selected in the first stage, which form in this way the sample subset $S_{h,\alpha} \subset U_{h,\alpha}$. Then, if cluster $U_{h,\alpha}$ is selected in the first stage (i.e., $1_h(\alpha) = 1$), $\forall i \in U_{h,\alpha}$ the probability for being included into the sample is,

$$\pi_{i|h,\alpha} = P(i \in S_{h,\alpha} | 1_h(\alpha) = 1) = P(1_{S_{h,\alpha}}(i) = 1 | 1_h(\alpha) = 1) = \frac{n_{h,\alpha}}{N_{h,\alpha}}, \quad \forall i \in U_{h,\alpha}, \tag{2.18}$$

where,

$$1_{S_{h,\alpha}}(i) = \begin{cases} 1 & \text{if} \quad i \in S_{h,\alpha}, \\ 0 & \text{if} \quad i \notin S_{h,\alpha}, \end{cases} \qquad \forall \alpha = 1, \ldots, A_h, \; \forall h = 1, \ldots, H. \tag{2.19}$$

Finally, note that two conditions need to be satisfied for $\forall i \in U_{h,\alpha}$ to be included in the sample. First, the cluster $U_{h,\alpha}$ needs to be selected. Then, unit i needs to be one of the units from $U_{h,\alpha}$ selected for being sampled. Therefore, the inclusion probability for each unit $i \in U_{h,\alpha}$ can be calculated as follows:

$$\begin{aligned} \pi_i &= P(i \in S) = P\left[(1_h(\alpha) = 1) \cap (1_{S_{h,\alpha}}(i) = 1)\right] = \\ &= P\left[1_{S_{h,\alpha}}(i) = 1 | 1_h(\alpha) = 1\right] \cdot P\left[1_h(\alpha) = 1\right] = \pi_{i|h,\alpha} \cdot \pi_{h,\alpha} = \\ &= \frac{n_{h,\alpha}}{N_{h,\alpha}} \cdot \frac{a_h}{A_h}, \quad \forall i \in U_{h,\alpha}, \; \forall h = 1, \ldots, H, \; \forall \alpha = 1, \ldots, A_h, \end{aligned} \tag{2.20}$$

which can also be rewritten in more general terms as,

$$\pi_i = \sum_{h=1}^{H} \sum_{\alpha=1}^{A_h} \pi_{i|h,\alpha} \cdot \pi_{h,\alpha} \cdot 1_{U_{h,\alpha}}(i) = \sum_{h=1}^{H} \sum_{\alpha=1}^{A_h} \cdot \frac{n_{h,\alpha}}{N_{h,\alpha}} \cdot \frac{a_h}{A_h} \cdot 1_{U_{h,\alpha}}(i), \quad \forall i \in U, \tag{2.21}$$

where,

$$1_{U_{h,\alpha}}(i) = \begin{cases} 1 & \text{if} \quad i \in U_{h,\alpha}, \\ 0 & \text{if} \quad i \notin U_{h,\alpha}, \end{cases} \qquad \forall \alpha = 1, \ldots, A_h, \; \forall h = 1, \ldots, H. \tag{2.22}$$

In summary, the subset $S_{h,\alpha} \subset U_{h,\alpha}$ is formed by the $n_{h,\alpha}$ units of $U_{h,\alpha}$ that are selected in the second stage of the sampling, assuming that the cluster $U_{h,\alpha}$ has been selected in the first stage, that is, $1_h(\alpha) = 1$. Note that if the cluster $U_{h,\alpha}$ has not

been selected in the first stage, then none of the units of $U_{h,\alpha}$ are selected to be part of the sample, that is, $\forall h \in \{1,\dots,H\}$ and $\forall \alpha \in \{1,\dots,A_h\}$,

$$\text{if } 1_h(\alpha) = 0 \Longrightarrow S_{h,\alpha} = \varnothing. \tag{2.23}$$

$\forall h \in \{1,\dots,H\}$, let us define the set $\mathbb{A}_h$ as follows:

$$\mathbb{A}_h = \{\alpha \in \{1,\dots,A_h\} \ : 1_h(\alpha) = 1\}. \tag{2.24}$$

In other words, the set $\mathbb{A}_h$ is the set of indexes indicating the clusters from stratum h, $\forall h \in \{1,\dots,H\}$ that have been selected in the first stage. Sample S is then formed as follows:

$$S = \bigcup_{h=1}^{H} \bigcup_{\alpha=1}^{A_h} S_{h,\alpha}, \tag{2.25}$$

which can also be rewritten as in eq. (2.26) taking into account the implication indicated in eq. (2.23):

$$S = \bigcup_{h=1}^{H} \bigcup_{\dot{\alpha}\in\mathbb{A}_h} S_{h,\dot{\alpha}}, \tag{2.26}$$

being $\mathbb{A}_h$ the set defined in eq. (2.24). Hereinafter, we use the indicator α to indicate the indexes of all the clusters in the population and the indicator $\dot{\alpha}$ to indicate specifically the clusters that have been selected in the first stage and end up in the sample.

The sampling weight assigned to each sampled observation is now calculated as the inverse of its inclusion probability:

$$w_i = \frac{1}{\pi_i} = \frac{N_{h,\dot{\alpha}}}{n_{h,\dot{\alpha}}} \cdot \frac{A_h}{a_h}, \quad \forall i \in S_{h,\dot{\alpha}}. \tag{2.27}$$

or equivalently,

$$w_i = \frac{1}{\pi_i} = \sum_{h=1}^{H} \sum_{\dot{\alpha}\in\mathbb{A}_h} \frac{N_{h,\dot{\alpha}}}{n_{h,\dot{\alpha}}} \cdot \frac{A_h}{a_h} \cdot 1_{S_{h,\dot{\alpha}}}(i), \quad \forall i \in S. \tag{2.28}$$

The sum of the sampling weights of all the units in the sample can then be calculated

as in eq. (2.29):

$$\begin{aligned}\sum_{i\in S} w_i &= \sum_{i\in S}\sum_{h=1}^{H}\sum_{\dot{\alpha}\in\mathbb{A}_h} \frac{N_{h,\dot{\alpha}}}{n_{h,\dot{\alpha}}}\cdot\frac{A_h}{a_h}\cdot 1_{S_{h,\dot{\alpha}}}(i) = \sum_{h=1}^{H}\sum_{\dot{\alpha}\in\mathbb{A}_h}\sum_{i\in S_{h,\dot{\alpha}}} \frac{N_{h,\dot{\alpha}}}{n_{h,\dot{\alpha}}}\cdot\frac{A_h}{a_h} = \\ &= \sum_{h=1}^{H}\sum_{\dot{\alpha}\in\mathbb{A}_h} n_{h,\dot{\alpha}}\cdot\frac{N_{h,\dot{\alpha}}}{n_{h,\dot{\alpha}}}\cdot\frac{A_h}{a_h} = \sum_{h=1}^{H}\sum_{\dot{\alpha}\in\mathbb{A}_h} N_{h,\dot{\alpha}}\cdot\frac{A_h}{a_h}.\end{aligned} \tag{2.29}$$

Let us suppose that the clusters in the same stratum are of the same size, i.e.,

$$N_{h,\alpha} = N_{h,\alpha'}, \quad \forall\alpha,\alpha' = 1,\dots,A_h \ : \alpha\neq\alpha', \quad \forall h = 1,\dots,H. \tag{2.30}$$

Then, given that $N_h = \sum_{\alpha=1}^{A_h} N_{h,\alpha}$, $N_{h,\alpha} = \dfrac{N_h}{A_h}$. Thus, note that if eq. (2.30) is satisfied, rewriting eq. (2.29), the sum of the sampling weights is equal to the population size:

$$\sum_{i\in S} w_i = \sum_{h=1}^{H}\sum_{\dot{\alpha}\in\mathbb{A}_h} N_{h,\dot{\alpha}}\cdot\frac{A_h}{a_h} = \sum_{h=1}^{H}\sum_{\dot{\alpha}\in\mathbb{A}_h}\frac{N_h}{A_h}\cdot\frac{A_h}{a_h} = \sum_{h=1}^{H}\frac{N_h}{A_h}\cdot\frac{A_h}{a_h}\cdot a_h = \sum_{h=1}^{H} N_h = N. \tag{2.31}$$

In other words, if all the clusters are the same size, then the sum of the sampling weights of all the units in the sample is equal to the population size.

Throughout this document, we will work under the assumption that eq. (2.30) is satisfied. In practice, it is recommended that the clusters of the same stratum, even if they are not exactly the same size, should not be too different in size (Kalton 1983). It should be noted that one of the greatest advantages of this type of sampling is that it can considerably reduce the economic costs of surveys, especially those conducted face-to-face, by being able to conduct the survey in specific areas, such as specific hospitals in a state or specific neighborhoods in a city.

Figure 2.2 depicts a visual description of this type of sampling design as an example. The finite population (dark blue rectangle) is partitioned into several strata (which are indicated with gray lines), and two of them are zoomed in. These strata have 5 clusters (on the left, let us denote as h^* this stratum) and 4 clusters (on the right, which will be denoted as h^{**}), respectively. From stratum h^*, two clusters are selected to be sampled. Therefore, the probability of being sampled for each cluster in this stratum can be calculated following eq. (2.17):

$$\pi_{h^*,\alpha} = \frac{2}{5} = 0.4, \quad \forall\alpha = 1,\dots,5. \tag{2.32}$$

And in the same way, given that two out of four clusters are sampled, the probability of being sampled for each cluster in stratum h^{**} is,

$$\pi_{h^{**},\alpha} = \frac{2}{4} = 0.5, \quad \forall \alpha = 1, \ldots, 4. \tag{2.33}$$

Let us denote as α' and α'' the clusters sampled from strata h^* and h^{**}. All of them are zoomed in, and the number of units in each of them is shown. In both selected clusters from stratum h^*, there are $N_{h^*,\alpha'} = N_{h^*,\alpha''} = 10$ units from which $n_{h^*,\alpha'} = 5$ and $n_{h^*,\alpha''} = 2$ are sampled, being the inclusion probability for each unit of these clusters defined as in eq. (2.18):

$$\pi_{i|h^*,\alpha'} = \frac{5}{10} = 0.5, \quad \forall i \in U_{h^*,\alpha'}, \tag{2.34}$$

$$\pi_{i|h^*,\alpha''} = \frac{2}{10} = 0.2, \quad \forall i \in U_{h^*,\alpha''}. \tag{2.35}$$

Therefore, the inclusion probability and the corresponding sampling weight for the units in the clusters α' and α'' from stratum h^* can be calculated as in eqs. (2.20) and (2.27), respectively:

$$\pi_i = \pi_{i|h^*,\alpha'} \cdot \pi_{h^*,\alpha'} = \frac{5}{10} \cdot \frac{2}{5} = 0.2 \quad \Longrightarrow \quad w_i = \frac{1}{\pi_i} = \frac{5}{2} \cdot \frac{10}{5} = 5, \forall i \in S_{h^*,\alpha'}, \tag{2.36}$$

$$\pi_i = \pi_{i|h^*,\alpha''} \cdot \pi_{h^*,\alpha''} = \frac{2}{10} \cdot \frac{2}{5} = 0.08 \quad \Longrightarrow \quad w_i = \frac{1}{\pi_i} = \frac{5}{2} \cdot \frac{10}{2} = 12.5, \forall i \in S_{h^*,\alpha''}. \tag{2.37}$$

Similarly, following the same steps as the ones described previously, we can easily calculate the sampling weights for the units sampled from clusters α' and α'' from stratum h^{**} (which in this case are equal, given that the same number of units are sampled from both clusters):

$$w_i = \frac{4}{2} \cdot \frac{4}{2} = 4, \quad \forall i \in S_{h^{**},\alpha'} \quad \text{and} \quad \forall i \in S_{h^{**},\alpha''}. \tag{2.38}$$

In this example, we only show the sampling process in two of the strata for illustration purposes. However, it is important to note that, in order to properly obtain a representative sample for the whole finite population, the sampling must be done one by one in each of the strata, and the sampling process followed in the two strata shown in the picture should be replicated in the rest of the strata of the finite population.

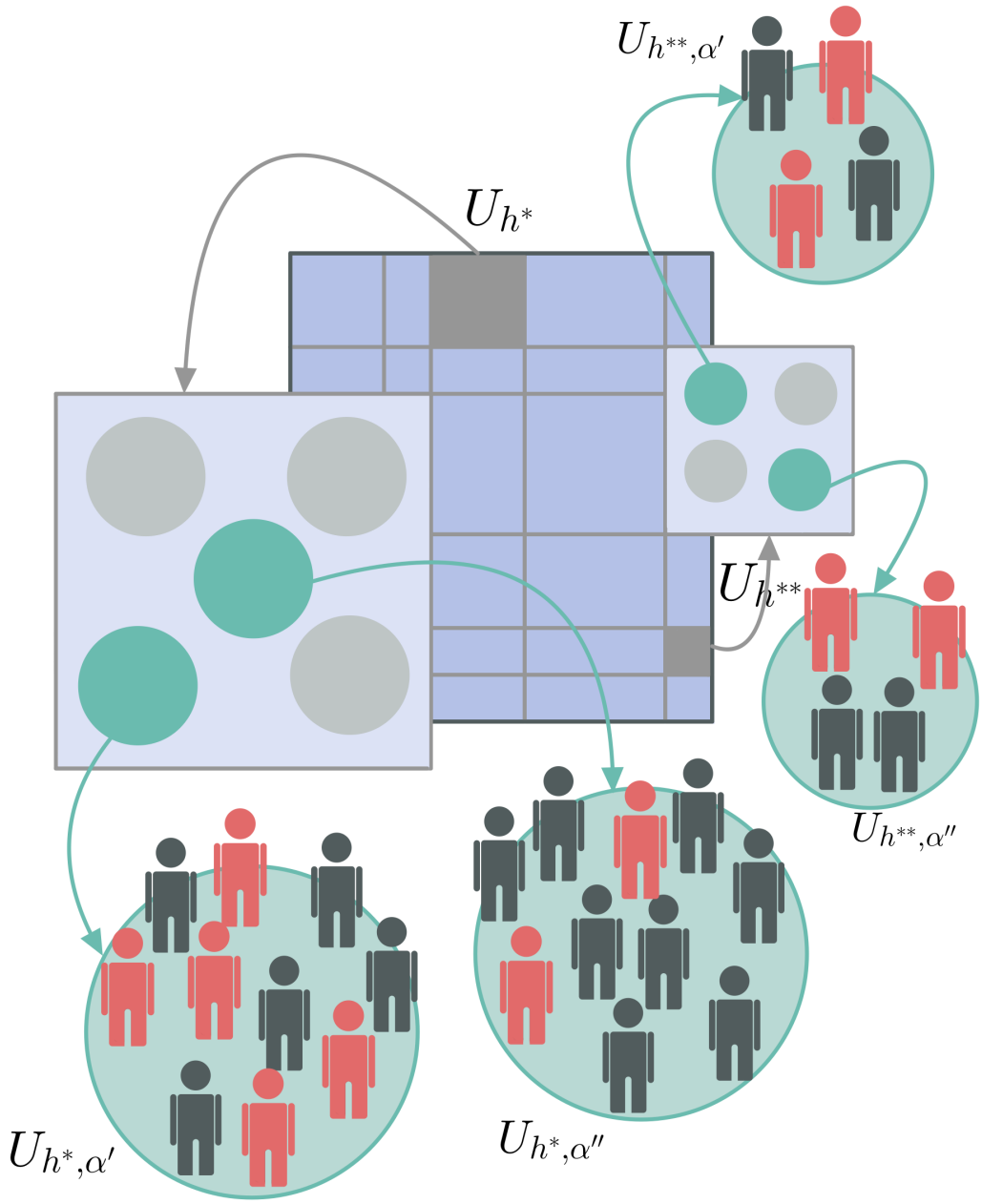


Figure 2.2: Two-stage stratified cluster sampling. The dark blue rectangle indicates the finite population. The gray lines separate the strata. The circles indicate clusters (in green, the ones that have been sampled). The sampled units from each cluster are indicated in red.

2.2 Background of the design-based estimation of prediction models

The goal of this chapter is to set the basic notation and methodological background to fit prediction models to complex survey data from the design-based perspective. In particular, throughout the document, we work with linear (Section 2.2.1) and logistic (Section 2.2.2) regression models, which will be introduced in the following lines. First, we introduce the traditional methodology to fit these prediction models to iid data. It should be noted that, in the context of survey data, when a prediction model is fitted, we aim to analyze the relationship between the characteristic of interest and the predictor variables in the finite population. Therefore, the finite population models would be the gold standard that we aim to approach based on the sampled information in this context. Thus, we first define finite population models, and finally, we introduce the methodology proposed to fit these models based on the sample obtained following complex sampling designs from the design-based perspective.

First of all, let us denote the basic notation. We denote as Y the characteristic of interest or the response variable (which follows either the Gaussian distribution for linear regression models or a Bernoulli distribution for logistic regression models) and as $\boldsymbol{X} = (1, X_1, \ldots, X_p)$ the set of covariates or predictor variables. Let $\{(y_i, \boldsymbol{x}_i)\}_{i \in S}$ indicate the set of observations of the response and predictor variables for the individuals in sample S of size n, where $\boldsymbol{x}_i = (1, x_{i1}, \ldots, x_{ip})$. The vector of regression coefficients will be denoted by $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^T$ in both linear and logistic regression models. For ease of notation, let us indicate $X_0 = 1$ and $x_{0i} = 1$, $\forall i \in S$ the variable and observations that are multiplying the intercept β_0, respectively.

2.2.1 Linear regression models

For a continuous response variable Y, the linear regression model for the observed data is defined as follows:

$$y_i = \boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2), \tag{2.39}$$

and the vector of regression coefficients is estimated (let us denote the vector of estimated regression coefficients as $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)^T$) based on sample S by

minimizing the residual sum of square (RSS):

$$\text{RSS}(\boldsymbol{\beta}) = \sum_{i \in S} (y_i - \sum_{j=0}^{p} \beta_j x_{ij})^2, \tag{2.40}$$

which can also be rewritten in a matrix form as follows:

$$\text{RSS}(\boldsymbol{\beta}) = (\boldsymbol{y} - \mathbf{X}_{n\times(p+1)}\boldsymbol{\beta})^T(\boldsymbol{y} - \mathbf{X}_{n\times(p+1)}\boldsymbol{\beta}), \tag{2.41}$$

where

$$\boldsymbol{y} = (y_1, \ldots, y_n)^T, \quad \mathbf{X}_{n\times(p+1)} = \begin{pmatrix} 1 & x_{11} & \ldots & x_{1p} \\ 1 & x_{21} & \ldots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & \ldots & x_{np} \end{pmatrix}. \tag{2.42}$$

Then, the linear regression coefficients are estimated by minimizing the RSS function described in eq. (2.40) and eq. (2.41). For this purpose, first, the function is derived with respect to each coefficient β_j, $\forall j = 0, \ldots, p$, and the resulting equations are equated to 0 to obtain the coefficient estimates, i.e.,

$$\frac{\partial\, \text{RSS}(\boldsymbol{\beta})}{\partial \beta_j} = -2 \sum_{i \in S} x_{ij}(y_i - \boldsymbol{x}_i\boldsymbol{\beta}) = 0, \quad \forall j = 0, \ldots, p, \tag{2.43}$$

which in matrix form can be rewritten as follows:

$$\mathbf{X}^T_{n\times(p+1)}(\boldsymbol{y} - \mathbf{X}_{n\times(p+1)}\boldsymbol{\beta}) = 0. \tag{2.44}$$

The coefficients that solve the previous equation are the ones that will be used for estimating the linear regression model ($\hat{\boldsymbol{\beta}}$):

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T_{n\times(p+1)}\mathbf{X}_{n\times(p+1)})^{-1}\mathbf{X}^T_{n\times(p+1)}\boldsymbol{y}. \tag{2.45}$$

Linear regression models from the design-based perspective

Let us suppose that information about the response variable Y and the vector of covariates $\boldsymbol{X}$ is available for all the units in the finite population U, i.e., $\{(y_i, \boldsymbol{x}_i)\}_{i=1}^N$. Then, the linear population model can be defined as follows:

$$y_i = \boldsymbol{x}_i\boldsymbol{\beta}^{\text{pop}} + \epsilon_i, \tag{2.46}$$

where the finite population regression coefficients $\boldsymbol{\beta}^{\text{pop}}$ can be computed by minimizing the population residual sum of square ($\text{RSS}_{\text{pop}}(\boldsymbol{\beta})$) in eq. (2.47):

$$\text{RSS}_{\text{pop}}(\boldsymbol{\beta}) = \sum_{i \in U} (y_i - \sum_{j=0}^{p} \beta_j x_{ij})^2, \tag{2.47}$$

Then, the vector of population coefficients, which will be denoted as $\boldsymbol{\beta}^{\text{pop}}$ hereinafter, could be obtained by following eqs. (2.43) and (2.45).

However, the problem arises from the fact that commonly the information on the response variable and covariates is not available for all the individuals in the finite population U, but only for those that have been incorporated into sample S, together with their respective sample weights, i.e., $\{(y_i, \boldsymbol{x}_i, w_i)\}_{i \in S}$, and thus, the population regression coefficients need to be estimated based on the sample S. In this context, Fuller (1975) proposed to minimize the weighted residual sum of square (WRSS) described in eq. (2.48) in order to estimate the population coefficients for linear regression models when working with complex survey data. Let $\mathbf{W}_{n \times n}$ be a $n \times n$ diagonal matrix with the sampling weights w_i, $\forall i = 1, \ldots, n$ in the diagonal, i.e., $\mathbf{W}_{n \times n} = \text{diag}(\{w_i\}_{i \in S})$. Then,

$$\text{WRSS}(\boldsymbol{\beta}) = \sum_{i \in S} w_i (y_i - \boldsymbol{x}_i \boldsymbol{\beta})^2 = (\boldsymbol{y} - \mathbf{X}_{n \times (p+1)} \boldsymbol{\beta})^T \mathbf{W}_{n \times n} (\boldsymbol{y} - \mathbf{X}_{n \times (p+1)} \boldsymbol{\beta}). \tag{2.48}$$

This function is differentiated with respect to β_j, $\forall j = 0, \ldots, p$, i.e.,

$$\frac{\partial \text{WRSS}(\boldsymbol{\beta})}{\partial \beta_j} = -2 \sum_{i \in S} w_i x_{ij} (y_i - \boldsymbol{x}_i \boldsymbol{\beta}) = \mathbf{X}^T_{n \times (p+1)} \mathbf{W}_{n \times n} (\boldsymbol{y} - \mathbf{X}_{n \times (p+1)} \boldsymbol{\beta}) = 0. \tag{2.49}$$

Solving the previous equations, we have that,

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T_{n \times (p+1)} \mathbf{W}_{n \times n} \mathbf{X}_{n \times (p+1)})^{-1} \mathbf{X}^T_{n \times (p+1)} \mathbf{W}_{n \times n} \boldsymbol{y}. \tag{2.50}$$

2.2.2 Logistic regression models

In a similar way, if Y is a dichotomous response variable, the logistic regression model is defined as,

$$\text{logit}(P(Y = 1 | \boldsymbol{x}_i)) = \text{logit}(p(\boldsymbol{x}_i)) = \ln \left[\frac{p(\boldsymbol{x}_i)}{1 - p(\boldsymbol{x}_i)} \right] = \boldsymbol{x}_i \boldsymbol{\beta}, \tag{2.51}$$

where,

$$p(\boldsymbol{x}_i) = P(Y = 1|\boldsymbol{x}_i) = \frac{e^{\boldsymbol{x}_i\boldsymbol{\beta}}}{1 + e^{\boldsymbol{x}_i\boldsymbol{\beta}}}, \tag{2.52}$$

and $\hat{\boldsymbol{\beta}}$ is obtained by maximizing the likelihood function $L(\boldsymbol{\beta})$:

$$L(\boldsymbol{\beta}) = \prod_{i \in S} p(\boldsymbol{x}_i)^{y_i} (1 - p(\boldsymbol{x}_i))^{1-y_i}. \tag{2.53}$$

Equivalently, for ease of calculation, the log-likelihood function $\ell(\boldsymbol{\beta})$ described in eq. (2.54) is maximized instead:

$$\begin{aligned} \ell(\boldsymbol{\beta}) = \ln(L(\boldsymbol{\beta})) &= \sum_{i \in S} \left[y_i \ln(p(\boldsymbol{x}_i)) + (1 - y_i) \ln(1 - p(\boldsymbol{x}_i)) \right] = \\ &= \sum_{i \in S} \left[y_i \ln\left(\frac{e^{\boldsymbol{x}_i\boldsymbol{\beta}}}{1 + e^{\boldsymbol{x}_i\boldsymbol{\beta}}} \right) + (1 - y_i) \ln\left(\frac{1}{1 + e^{\boldsymbol{x}_i\boldsymbol{\beta}}} \right) \right] = \\ &= \sum_{i \in S} \left\{ y_i \left[\boldsymbol{x}_i\boldsymbol{\beta} - \ln\left(1 + e^{\boldsymbol{x}_i\boldsymbol{\beta}}\right) \right] - (1 - y_i) \ln\left(1 + e^{\boldsymbol{x}_i\boldsymbol{\beta}}\right) \right\} = \\ &= \sum_{i \in S} \left\{ y_i \boldsymbol{x}_i\boldsymbol{\beta} - \ln\left(1 + e^{\boldsymbol{x}_i\boldsymbol{\beta}}\right) \right\}. \end{aligned} \tag{2.54}$$

This function is differentiated for all the coefficients to end up with the following likelihood equations:

$$\frac{\partial\, \ell(\boldsymbol{\beta})}{\partial \beta_j} = \sum_{i \in S} x_{ij} \left(y_i - \frac{e^{\boldsymbol{x}_i\boldsymbol{\beta}}}{1 + e^{\boldsymbol{x}_i\boldsymbol{\beta}}} \right) = 0, \quad \forall j = 0, \ldots, p. \tag{2.55}$$

Given that there is not a closed form that solves the likelihood equations, logistic regression coefficients are usually estimated by means of iterative methods such as the Iterative Weighted Least Squares (IWLS) (McCullagh and Nelder 1989).

Logistic regression models from the design-based perspective

Let us suppose that information about the response variable Y and the vector of covariates $\boldsymbol{X}$ is available for all the units in the finite population U, i.e., $\{(y_i, \boldsymbol{x}_i)\}_{i=1}^{N}$. Then, the logistic regression population model can be defined as follows, respectively:

$$\text{logit}(p(\boldsymbol{x}_i)) = \boldsymbol{x}_i\boldsymbol{\beta}^{\text{pop}}, \tag{2.56}$$

where the finite population regression coefficients $\boldsymbol{\beta}^{\text{pop}}$ can be computed by maximizing the population likelihood function ($L_{\text{pop}}(\boldsymbol{\beta})$) in eq. (2.57):

$$L_{\text{pop}}(\boldsymbol{\beta}) = \prod_{i \in U} p(\boldsymbol{x}_i)^{y_i} (1 - p(\boldsymbol{x}_i))^{1-y_i}. \tag{2.57}$$

Then, the population coefficients $\boldsymbol{\beta}^{\text{pop}}$ could be obtained by following the procedures previously explained in eq. (2.54) and eq. (2.55).

However, given that only the information of sampled units is available, Binder (1981; 1983) proposed to use the pseudo-likelihood function described in eq. (2.58) that considers the sampling weights to estimate the population regression coefficients ($\hat{\boldsymbol{\beta}}$):

$$PL(\boldsymbol{\beta}) = \prod_{i \in S} p(\boldsymbol{x}_i)^{y_i w_i} (1 - p(\boldsymbol{x}_i))^{(1-y_i) w_i}. \tag{2.58}$$

In order to ease the calculations, the logarithm of the pseudo-likelihood function defined in eq. (2.59) is maximized:

$$\begin{aligned} p\ell(\boldsymbol{\beta}) = \ln(PL(\boldsymbol{\beta})) &= \sum_{i \in S} [w_i y_i \ln(p(\boldsymbol{x}_i)) + w_i (1 - y_i) \ln(1 - p(\boldsymbol{x}_i))] = \\ &= \sum_{i \in S} \left[w_i y_i \ln\left(\frac{e^{\boldsymbol{x}_i \boldsymbol{\beta}}}{1 + e^{\boldsymbol{x}_i \boldsymbol{\beta}}} \right) + w_i (1 - y_i) \ln\left(\frac{1}{1 + e^{\boldsymbol{x}_i \boldsymbol{\beta}}} \right) \right] = \\ &= \sum_{i \in S} \left\{ w_i y_i \left[\boldsymbol{x}_i \boldsymbol{\beta} - \ln\left(1 + e^{\boldsymbol{x}_i \boldsymbol{\beta}}\right) \right] - w_i (1 - y_i) \ln\left(1 + e^{\boldsymbol{x}_i \boldsymbol{\beta}}\right) \right\} = \\ &= \sum_{i \in S} \left\{ w_i y_i \boldsymbol{x}_i \boldsymbol{\beta} - w_i \ln\left(1 + e^{\boldsymbol{x}_i \boldsymbol{\beta}}\right) \right\}. \end{aligned} \tag{2.59}$$

This function is differentiated for all the coefficients to end up with the following likelihood equations:

$$\frac{\partial\, p\ell(\boldsymbol{\beta})}{\partial \beta_j} = \sum_{i \in S} w_i x_{ij} \left(y_i - \frac{e^{\boldsymbol{x}_i \boldsymbol{\beta}}}{1 + e^{\boldsymbol{x}_i \boldsymbol{\beta}}} \right) = 0, \quad \forall j = 0, \ldots, p, \tag{2.60}$$

which are solved by means of iterative methods in the same way as the traditional logistic regression model (Heeringa et al. 2017).

CHAPTER 3

Motivating Data Sets

The goals raised in this dissertation are motivated by two real surveys designed and collected by the Official Statistics Basque Office (EUSTAT): the Survey on the Information Society in Companies[7] (ESIE) and the Survey on the Population with Relation to Activity[8] (PRA). In Section 3.1 and Section 3.2, we describe these two real surveys, giving information about the finite population, the sample, and the sampling process of each of them. In addition, in the upcoming chapters, we carry out several simulation studies based on these surveys. Thus, in this chapter, we also describe the generation process of the synthetic data that have been used in those studies.

[7] https://en.eustat.eus/estadisticas/tema_150/opt_1/tipo_7/temas.html
[8] https://en.eustat.eus/estadisticas/tema_37/opt_0/temas.html

3.1 Survey on the Information Society in Companies (ESIE)

The Survey on the Information Society in Companies is usually denoted as ESIE due to its Spanish acronym (specifically, *Encuesta sobre la Sociedad de la Información en Empresas*). ESIE provides regular information on the implementation of New Information and Communication Technology in the companies of Basque Country (BC, hereinafter). This survey collects information about the use of the Internet in different establishments of the companies in BC. EUSTAT conducts this survey annually. In particular, the information that will be used for illustration purposes throughout this dissertation was collected in 2010 (we would like to point out that, for the purposes of this thesis, the year in which the data was collected is not relevant and we picked one at random).

3.1.1 Descriptive analysis of real data

In the ESIE survey, from the 2010 finite population of 195 222 establishments, 7 725 were sampled by one-stage stratified sampling (see Section 2.1.1 for more information on the sampling process). Strata are defined by means of the combination of three categorical variables: the province where the company is located (3 categories), the activity of the company (65 categories), and the number of employees (categorized in 3 categories, i.e., < 10, $10 - 99$, ≥ 100). In this way, a total of $3 \times 65 \times 3 = 585$ different strata have been defined. However, it should be noted that in some of these strata there are no units in the population, so in fact, we have 515 non-empty strata in total ($h = 1, \ldots, H$, where $H = 515$). In particular, strata sizes in the finite population range from 1 to 14 535, where the median is 38 and the interquartile range is $7 - 185.5$. It can be seen that due to the large number of total strata, most of them are relatively small. The sampling probabilities for each stratum also vary considerably and range from 0.006 to 1, with a median of 0.283 and an interquartile range of $0.097 - 0.842$.

However, due to the large number of establishments in BC, EUSTAT is sometimes interested in analyzing particularly the behavior of the greatest establishments that have at least 10 employees. In this case, the finite population in 2010 was formed by 14 200 establishments from which 2 852 were sampled. When reducing the number of employees, the categories of that variable are reduced from 3 to 2 when defining the strata, and thus, in this situation, a total of 390 strata have been

defined, from which 325 have at least one company (i.e., $H = 325$). Strata sizes in the finite population (i.e., N_h, $\forall h \in \{1, \ldots, H\}$) range from 1 to 860, where the median is 12 and the interquartile range $4 - 44$. The probabilities of being sampled from each stratum (i.e., n_h/N_h, $\forall h \in \{1, \ldots, H\}$) range from 0.039 to 1 (with a median of 0.667 and an interquartile range of $0.260 - 1$). We will use both populations (the complete one with all establishments and the one with only medium and large establishments) in the different chapters of this document.

Regarding the variables of interest, even though in the survey more information about other response variables is also collected, in the data set provided by EUSTAT, we have available information about the following four dichotomous response variables, the ones that indicate whether the company:

- has access to the internet (Y_1),
- carries out online transactions (Y_2, only recorded in case the company has access to the internet, i.e., $Y_1 = 1$),
- has its own website (Y_3),
- has its own website in the Basque language (Y_4, only recorded in case the company has its own website, i.e., $Y_3 = 1$).

In order to describe more clearly the relationship between the different variables of interest, Figure 3.1 is shown below.

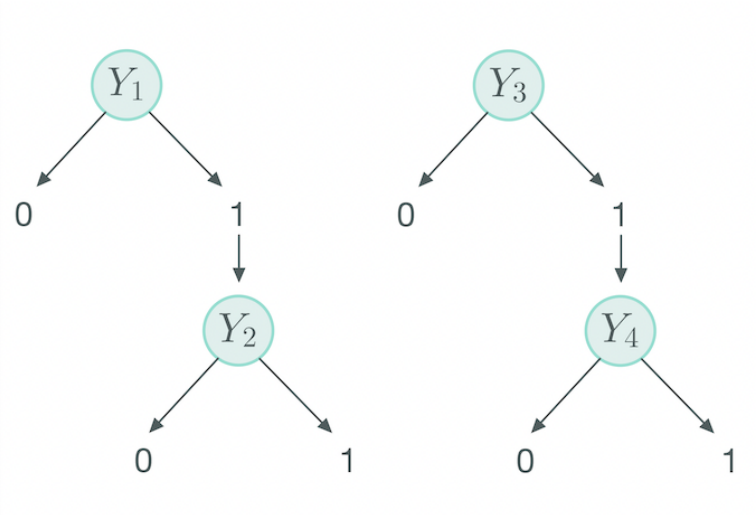


Figure 3.1: Relationship between the variables of interest of the ESIE survey data.

A descriptive analysis of the dichotomous variables of interest can be found in Table 3.1. All of them have been considered to generate the synthetic data set based on the ESIE survey data in order for it to be as similar as possible to the true finite population. Thus, all of them are described in Table 3.1. Nevertheless, only one of them has been considered as the response variable in the models fitted to this data set in the following chapters. In particular, the dichotomous response variable considered for this work indicates whether a company has its own website (1) or not (0) (i.e., the one we have previously defined as Y_3). The probability of event estimated in the sample without considering the sampling weights is 0.822 (i.e., the number of units with the event of interest, divided by the total number of units in the sample), while the weighted estimate of the probability of event is 0.754, computed by taking into account the number of units that each element represents in the finite population by means of the sampling weights as follows:

$$\hat{p}_{Y_3,w} = \frac{\sum_{i \in S} w_i \cdot I(y_{3i} = 1)}{\sum_{i \in S} w_i}, \tag{3.1}$$

where

$$I(y_{3i} = 1) = \begin{cases} 1 & \text{if} \quad y_{3i} = 1, \\ 0 & \text{if} \quad y_{3i} = 0, \end{cases} \qquad \forall i \in S. \tag{3.2}$$

Note that, regarding the notation of Table 3.1, $\hat{N}$ indicates the number of units from the finite population represented by means of the sampled units (as the sum of their weights) that take a value in a particular response variable. In particular, for Y_3, this is calculated as follows, given that this information is available for all the sampled units:

$$\hat{N}_{Y_3} = \sum_{i \in S} w_i. \tag{3.3}$$

In a similar way, $\hat{N}_{Y=0}$ and $\hat{N}_{Y=1}$ indicate the number of units represented by means of the sampled units without and with the variable of interest, which in case of Y_3 would be:

$$\hat{N}_{Y_3=0} = \sum_{i \in S} w_i \cdot I(y_{3i} = 0), \quad \text{and} \quad \hat{N}_{Y_3=1} = \sum_{i \in S} w_i \cdot I(y_{3i} = 1). \tag{3.4}$$

Thus, note that the weighted estimate of the probability of event can also be expressed as follows:

$$\hat{p}_{Y_3,w} = \frac{\hat{N}_{Y_3=1}}{\hat{N}_{Y_3}}. \tag{3.5}$$

The differences between the weighted and unweighted estimates of the probability of event are remarkable. It should also be noted that, due to the large number of strata and their relatively small size, the behavior of the response variable varies considerably depending on the strata. Particularly, the unweighted sample probability of event ranges from 0 to 1 across different strata with a median of 0.769 and an interquartile range of $0.472 - 1$. These differences, together with the imbalanced sampling across the strata, make the weighted and unweighted estimates of the probability of event differ.

Due to the large number of available covariates in the real data set, a descriptive analysis of those covariates included in the models fitted throughout this dissertation can be observed in Table 3.2 (accounting for all the establishments in BC) and Table 3.3 (considering only the establishments with at least 10 employees). Those covariates include information about the ownership of the establishments (7 categories), the type of activity they undertake (categorized in 3 categories), the number of employees they have (6 categories), and the province in which the establishment is located (3 categories). It should be noted that information on these four covariates is available not only for the sampled units but for all the establishments in the finite population, as shown in the tables. In addition, three out of four variables are the same variables previously used to define the strata, but not necessarily with the same categorization. In particular, the activity and the number of employees (both of them with a different categorization) and the province (with exactly the same categorization indicated before).

Table 3.1: Descriptive analysis of the response variables available in the ESIE survey data (corresponding to 2010), which include information about the number of sampled units with information on the response variable (n), the number of sampled units without ($n_{Y=0}$) and with ($n_{Y=1}$) the event of interest; the number (and percentage %) of units that the n sampled units represent by means of the sampling weights ($\hat{N}$, calculated as the sum of their weights), and the number of units and percentages that represent the units without ($\hat{N}_{Y=0}$ (%)) and with ($\hat{N}_{Y=1}$ (%)) the event of interest considering sampling weights.

	All the establishments in BC					
	n	$n_{Y=0}$ (%)	$n_{Y=1}$ (%)	$\hat{N}$	$\hat{N}_{Y=0}$ (%)	$\hat{N}_{Y=1}$ (%)
Y_1	7 725	1 320 (17.1)	6 405 (82.9)	195 222	58 704 (30.1)	136 518 (69.9)
Y_2	6 405	1 725 (26.9)	4 680 (73.1)	136 518	52 628 (38.6)	83 890 (61.4)
Y_3	7 725	3 444 (44.6)	4 281 (55.4)	195 222	129 124 (66.1)	66 098 (33.9)
Y_4	4 281	2 470 (57.7)	1 811 (42.3)	66 098	41 865 (63.3)	24 232 (36.7)
	Establishments with at least 10 employees					
	n	$n_{Y=0}$ (%)	$n_{Y=1}$ (%)	$\hat{N}$	$\hat{N}_{Y=0}$ (%)	$\hat{N}_{Y=1}$ (%)
Y_1	2 852	80 (2.8)	2 772 (97.2)	14 200	542 (3.8)	13 658 (96.2)
Y_2	2 772	317 (11.4)	2 455 (88.6)	13 658	2 067 (15.1)	11 591 (84.9)
Y_3	2 852	507 (17.8)	2 345 (82.2)	14 200	3 488 (24.6)	10 712 (75.4)
Y_4	2 345	1 298 (55.4)	1 047 (44.6)	10 712	6 236 (58.2)	4 476 (41.8)

Table 3.2: Descriptive analysis of the covariates considered in the models fitted to ESIE survey data considering all the establishments in BC. In particular, the following information is provided: the name of the variable (in brackets, the codification of those variables we use throughout this dissertation), the category (in brackets, their codification), the number (and percentage) of the units that take each category in the finite population (N (%)) and in the sample (n (%)), and the number of units without ($n_{Y_3=0}$) and with ($n_{Y_3=1}$) the event of interest in the sample.

Variable	**Category**	**All the establishments in BC**			
		N (%)	n (%)	$n_{Y_3=0}$	$n_{Y_3=1}$
Ownership (X_1)	Natural person (1)	105 271 (53.9)	1 810 (23.4)	1 538	272
	Working partnership (2)	459 (0.2)	46 (0.6)	15	31
	Limited partnership (3)	12 285 (6.3)	1 661 (21.5)	262	1 399
	Cooperative (4)	2 038 (1.0)	238 (3.1)	42	196
	Limited company (5)	52 748 (27.0)	2 636 (34.1)	1 123	1 513
	Public administration (6)	3 067 (1.6)	545 (7.1)	86	459
	Others (7)	19 354 (9.9)	789 (10.2)	378	411
Activity (X_2)	Industry (2)	14 498 (7.4)	1 638 (21.2)	617	1 021
	Construction (3)	30 879 (15.8)	356 (4.6)	232	124
	Services (4)	149 845 (76.8)	5 731 (74.2)	2 595	3 136
Number of employees (X_3)	0-2 (1)	171 589 (87.9)	2 720 (35.2)	1 991	729
	3-9 (2)	9 444 (4.8)	2 153 (27.9)	946	1 207
	10-19 (3)	6 991 (3.6)	842 (10.9)	261	581
	20-49 (4)	4 786 (2.5)	714 (9.2)	138	576
	50-99 (5)	1 436 (0.7)	417 (5.4)	39	378
	$\geq$ 100 (6)	976 (0.5)	879 (11.4)	69	810
Province (X_4)	Araba (1)	25679 (13.2)	1720 (22.3)	751	969
	Gipuzkoa (2)	67962 (34.8)	2715 (35.1)	1231	1484
	Bizkaia (3)	101581 (52.0)	3290 (42.6)	1462	1828

Table 3.3: Descriptive analysis of the covariates considered in the models fitted to ESIE survey data considering only those companies with at least 10 employees in the BC. In particular, the following information is provided: the name of the variable (in brackets, the codification of those variables we use throughout this dissertation), the category (in brackets, their codification), the number (and percentage) of the units that take each category in the finite population (N (%)) and in the sample (n (%)), and the number of units without ($n_{Y_3=0}$) and with ($n_{Y_3=1}$) the event of interest in the sample.

Variable	**Category**	**Establishments with at least 10 employees**			
		N (%)	n (%)	$n_{Y_3=0}$	$n_{Y_3=1}$
Ownership (X_1)	Natural person (1)	259 (1.8)	31 (1.1)	22	9
	Working partnership (2)	172 (1.2)	23 (0.8)	3	20
	Limited partnership (3)	4 397 (31.0)	1 140 (40.0)	141	999
	Cooperative (4)	601 (4.2)	159 (5.6)	13	146
	Limited company (5)	6 084 (42.9)	915 (32.1)	251	664
	Public administration (6)	1 579 (11.1)	376 (13.2)	41	335
	Others (7)	1 097 (7.7)	208 (7.3)	36	172
Activity (X_2)	Industry (2)	3 648 (25.7)	888 (31.1)	157	731
	Construction (3)	1 580 (11.1)	119 (4.2)	44	75
	Services (4)	8 961 (63.2)	1 845 (64.7)	306	1 539
Number of employees (X_3)	10-19 (3)	6 991 (49.3)	842 (29.5)	261	581
	20-49 (4)	4 786 (33.7)	714 (25.0)	138	576
	50-99 (5)	1 436 (10.1)	417 (14.6)	39	378
	$\geq$ 100 (6)	976 (6.9)	879 (30.8)	69	810
Province (X_4)	Araba (1)	2388 (16.8)	646 (22.7)	113	533
	Gipuzkoa (2)	4729 (33.3)	958 (33.6)	155	803
	Bizkaia (3)	7072 (49.8)	1248 (43.8)	239	1009

3.1.2 Pseudo-population generation and sampling process

In order to carry out different simulation studies for various challenges faced throughout this dissertation, an artificial population (denoted as the pseudo-population, hereinafter) was generated based on the available information on the true finite population and the sample corresponding to ESIE. The pseudo-population has been generated considering all the establishments in BC. In the cases in which we have worked with establishments with at least 10 employees, the subset of the whole pseudo-population has been considered. The data generation process, as well as the sampling process of this pseudo-population, are explained in the following lines.

In order to distinguish between the real and simulated data, let us denote as $\mathcal{S}$ the original (real) survey sample and as $\mathcal{U}$ the real finite population of size N ($\mathcal{S} \subset \mathcal{U}$). As explained above, a total of H strata have been defined (i.e., $\{1, \ldots, H\}$) combining information of three categorical variables, which will be denoted as $X_1^{(H)}$, $X_2^{(H)}$ and $X_3^{(H)}$.

Our goal is to generate a pseudo-population (U) based on the known real ESIE survey data, for which all the information of the covariates $X_1, \ldots, X_p$ and the response variables $Y_1, \ldots, Y_4$ will be available (note that in the real finite population Y_1, Y_2, Y_3 and Y_4 are not available). This new pseudo-population U will be the same size as the true ESIE population $\mathcal{U}$ (N). In order to ease the notation, the variable names of the pseudo-population are the same as in the real finite population. In contrast, the units of the real ESIE population will be denoted as $i' \in \mathcal{U}$ while the units that are artificially generated for the pseudo-population will be denoted as $i \in U$.

As explained above, several dichotomous response variables are available in the original survey. All possible combinations of these response variables have been examined. Let us go back to Figure 3.1 for a better understanding. If the response variable $Y_1 = 0$, then information about Y_2 is not recorded, and the same happens to Y_4 if $Y_3 = 0$. Thus, let us define two new variables as follows. $\forall i' \in \mathcal{S}$,

$$y_{1i'}^{(*)} = \begin{cases} 0, & \text{if} \quad y_{1i'} = 0, \\ 1, & \text{if} \quad y_{1i'} = 1,\ y_{2i'} = 1, \\ 2, & \text{if} \quad y_{1i'} = 1,\ y_{2i'} = 0, \end{cases} \quad \text{and} \quad y_{3i'}^{(*)} = \begin{cases} 0, & \text{if} \quad y_{3i'} = 0, \\ 1, & \text{if} \quad y_{3i'} = 1,\ y_{4i'} = 1, \\ 2, & \text{if} \quad y_{3i'} = 1,\ y_{4i'} = 0. \end{cases} \tag{3.6}$$

Let then $\boldsymbol{y}_{i'}^{(*)} = (y_{1,i'}^{(*)}, y_{3,i'}^{(*)})$ indicate the set of values of the response variables, corresponding to $\forall i' \in \mathcal{S}$. Then, $\boldsymbol{y}_{i'}^{(*)}$ is necessarily equal to one of the following

combinations:

$$\mathfrak{C} = \{\mathfrak{c}_1, \ldots, \mathfrak{c}_9\} = \{(0,0), (0,1), (0,2), (1,0), (1,1), (1,2), (2,0), (2,1), (2,2)\}. \quad (3.7)$$

That is, $\forall i' \in \mathcal{S},\ \exists! \mathfrak{c} \in \mathfrak{C} : \boldsymbol{y}_{i'}^{(*)} = \mathfrak{c}$.

For each stratum $h \in \{1, \ldots, H\}$ and for each possible combination of the responses (i.e., $\forall \mathfrak{c} \in \mathfrak{C}$) we generate $N_{h,\mathfrak{c}}$ units in the pseudo-population (U) with information about the response variable and strata. This procedure is summarized in Figure 3.2 and can be written mathematically as follows:

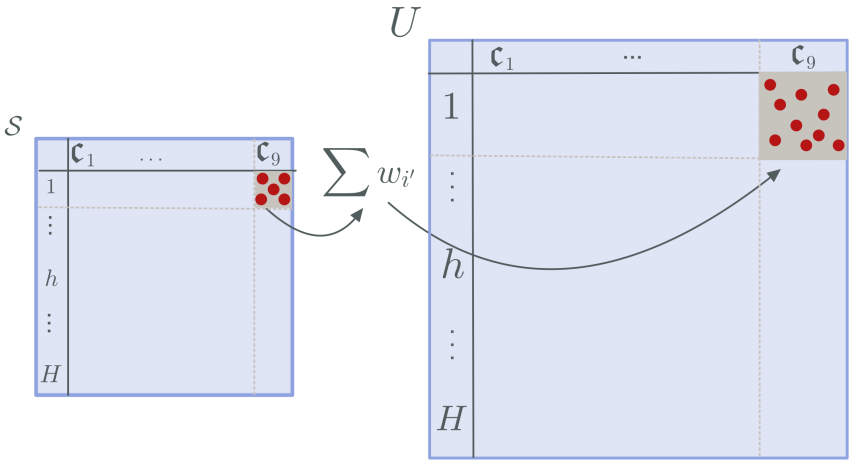


Figure 3.2: Graphical representation of the generation of the pseudo-population U with $N = \sum_{h=1}^{H} \sum_{\mathfrak{c} \in \mathfrak{C}} N_{h,\mathfrak{c}}$ elements.

$$N_{h,\mathfrak{c}} = \sum_{i' \in \mathcal{S}} w_{i'} \cdot 1_{\mathcal{U}_h}(i') \cdot I\left(\boldsymbol{y}_{i'}^{(*)} = \mathfrak{c}\right), \quad (3.8)$$

where,

$$1_{\mathcal{U}_h}(i') = \begin{cases} 1, & \text{if } i' \in \mathcal{U}_h, \\ 0, & \text{if } i' \notin \mathcal{U}_h, \end{cases} \quad (3.9)$$

and

$$I\left(\boldsymbol{y}_{i'}^{(*)} = \mathfrak{c}\right) = \begin{cases} 1 & \text{if} \quad \boldsymbol{y}_{i'}^{(*)} = \mathfrak{c}, \\ 0 & \text{if} \quad \boldsymbol{y}_{i'}^{(*)} \neq \mathfrak{c}. \end{cases} \tag{3.10}$$

In this way, $N_{h,\mathfrak{c}}$ is the number of units of the pseudo-population U in stratum h, which take the values of responses $\boldsymbol{y}_{i'}^{(*)} = (y_{1,i'}^{(*)}, y_{3,i'}^{(*)}) = \mathfrak{c}$. Once we repeat the process for $\forall h \in \{1, \ldots, H\}$ and $\forall \mathfrak{c} \in \mathfrak{C}$ a pseudo-population of N units is generated with the information of response variables and strata (hence, information for the design variables $X_1^{(H)}$, $X_2^{(H)}$ and $X_3^{(H)}$ will also be generated), where N is calculated as:

$$N = \sum_{h=1}^{H} \sum_{\mathfrak{c} \in \mathfrak{C}} N_{h,\mathfrak{c}} = \sum_{i' \in \mathcal{S}} w_{i'}. \tag{3.11}$$

Finally, we generate the rest of the covariates (note all of them are categorical) as follows. $\forall j \in \{1, \ldots, p\}$ for ease of notation we assume that X_j is a categorical variable with a total of G categories denoted as $\{1, \ldots, G\}$. $\forall i \in U$, $\exists! h \in \{1, \ldots, H\}$: $i \in U_h$, that is, $1_{U_h}(i) = 1$. Then, we generate the value $x_{ji} \in \{1, \ldots, G\}$ following a categorical distribution:

$$x_{ji} \sim Cat(\pi_{j1}^{(h)}, \ldots, \pi_{jG}^{(h)}), \quad \text{where } h \text{ satisfies} \quad 1_{U_h}(i) = 1, \tag{3.12}$$

and the probability corresponding to each category $g \in \{1, \ldots, G\}$, $\pi_{jg}^{(h)}$, is calculated as the fraction of the number of units in the true ESIE finite population in stratum h that takes the value g in the covariate X_j (let us denote this value as $N_{h,jg}$) and the total number of units in that stratum (N_h) (see Figure 3.3), which can be defined as follows mathematically:

$$\pi_{jg}^{(h)} = \frac{\sum_{i' \in \mathcal{U}} 1_{\mathcal{U}_h}(i') \cdot I(x_{ji'} = g)}{\sum_{i' \in \mathcal{U}} 1_{\mathcal{U}_h}(i')} = \frac{N_{h,jg}}{N_h}, \quad \forall g \in \{1, \ldots, G\}, \tag{3.13}$$

where $1_{\mathcal{U}_h}(i')$ is defined in eq. (3.9) and,

$$I(x_{ji'} = g) = \begin{cases} 1 & \text{if} \quad x_{ji'} = g, \\ 0 & \text{if} \quad x_{ji'} \neq g, \end{cases} \quad \forall i' \in \mathcal{U} \text{ and } \forall g \in \{1, \ldots, G\}. \tag{3.14}$$

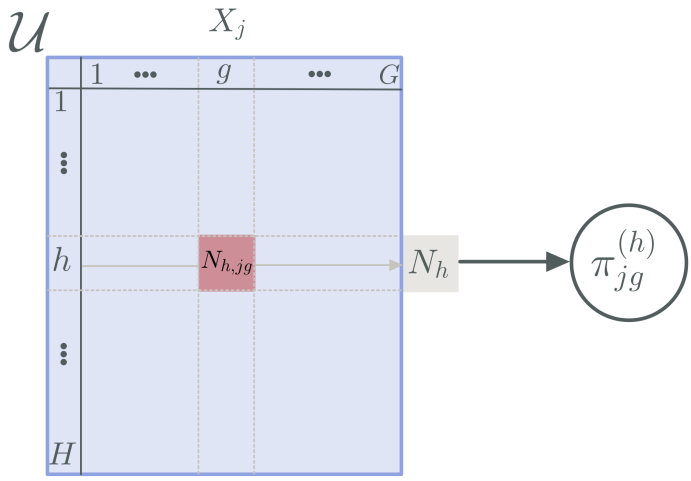


Figure 3.3: Graphical explanation of the calculation of the probabilities to be considered in the categorical distribution defined in eq. (3.12).

In this way, the pseudo-population based on the ESIE survey has been generated with the response variable Y, the vector of explanatory variables $\boldsymbol{X}$, and the strata. Note that these variables follow the same distribution as the real survey data shown in Tables 3.1 and 3.2 due to the way in which they have been generated.

The pseudo-population U has been sampled following a single-stage stratified sampling design (as described in Section 2.1.1), mimicking the original sampling procedure. Thus, a pre-specified number of units (n_h, $\forall h = 1, \ldots, H$), established by EUSTAT, were sampled from each pseudo-population stratum, forming in this way the sample S. We calculated the sampling weights for sampled units following eq. (2.7), i.e.,

$$w_i = \sum_{h=1}^{H} 1_{S_h}(i) \cdot \frac{N_h}{n_h}, \quad \forall i \in S. \tag{3.15}$$

It should be noted that the objective of generating the pseudo-population is not to be able to draw conclusions about the real population based on the simulated data. Rather, these data have simply been used in simulation studies where the main objective has been to analyze the effect of a realistic sampling design (defined

in a real survey) that allows us to understand a little better what can happen when we work with real complex survey data. Thus, we are aware that there may be reasonable differences between the simulated population and the real one, so we should not use the simulated one to draw conclusions related to reality, but it does allow us to draw conclusions regarding the effect of the sampling design.

3.2 Survey on the Population with Relation to Activity (PRA)

On the other hand, the Population with Relation to Activity (PRA) Survey is conducted among the inhabitants of BC aged 16 and over every three months, with the aim of estimating the percentage of the labor force of BC. In this dissertation, we consider information related to the last quarter of 2016 for illustration purposes. Specifically, the response variable Y that we consider in this study indicates whether each individual is *active* (1) or not (0). Following the definition of EUSTAT, we denote as *active* those that are part of the *working population*, that is, "those persons who work to produce goods and services, and who do not have a current job, are seeking employment and are available to start work".

From a total of 1 851 316 individuals 10 609 were sampled following one-stage stratified sampling. In this survey, strata are defined as the regions of BC, which are a total of 23. Specifically, strata sizes range from 2 768 to 438 595, being the median 44 335 and 22 024 – 72 834 the interquartile range. Thus, we can say that the strata defined in this survey are few (given the number of units in the population) and large. The sampling probabilities range from 0.004 to 0.049, with a median of 0.006 (the interquartile range is $0.006 - 0.010$).

The sample estimate of the probability of event (without considering sampling weights) is 0.540, while if we account for the sampling weights, the weighted estimate is 0.564. The unweighted sample estimates of the probability of event across the strata do not vary considerably either, given that it ranges from 0.484 to 0.632, with a median of 0.549 and interquartile range of $0.526 - 0.559$.

Among the most important covariates, we found age, educational level, nationality, and sex, which are described in Table 3.4.

To perform the simulation studies related to this survey, instead of generating a pseudo-population, EUSTAT provided us with one they have generated based on their own analysis. The main characteristics of the real population and the pseudo-

population are similar. In the pseudo-population, information about 1 830 443 individuals is available, and the population probability of event is 0.581. Strata sizes range from 2 541 to 438 887, with a median of 46 211 and an interquartile range of 20 818 - 73 227. A descriptive analysis of the variables is displayed in Table 3.5. Note that in the pseudo-population, in contrast to the true population, there are no missing values.

To sample the pseudo-population, a one-stage stratified sampling design was followed as in the original sample. The number of units to be sampled from each stratum (n_h, $\forall h \in \{1, \ldots, H\}$) was established by EUSTAT, and the sampling weights for the sampled units were calculated as usual following eq. (2.7):

$$w_i = \sum_{h=1}^{H} 1_{S_h}(i) \cdot \frac{N_h}{n_h}, \quad \forall i \in S. \tag{3.16}$$

Table 3.4: Descriptive analysis of the covariates considered in the models fitted to PRA survey data. In particular, the following information is provided: the name of the variable (in brackets, the codification of those variables we use throughout this dissertation), the category (in brackets, their codification), the number (and percentage) of the units that take each category in the finite population (N (%)) and in the sample (n (%)), and the number of units without ($n_{Y=0}$) and with ($n_{Y=1}$) the event of interest in the sample.

Variable	**Category**	N **(%)**	n **(%)**	$n_{Y=0}$	$n_{Y=1}$
Age (X_1)	16-24 (1)	162 146 (8.8)	940 (8.9)	702	238
	25-29 (2)	100 215 (5.4)	417 (3.9)	63	354
	30-34 (3)	126 619 (6.8)	524 (4.9)	48	476
	35-44 (4)	348 604 (18.8)	1 855 (17.5)	174	1 681
	45-54 (5)	345 072 (18.6)	2 058 (19.4)	258	1 800
	55-64 (6)	299 112 (16.2)	1 863 (17.6)	724	1 139
	$\geq$ 65 (7)	469 548 (25.4)	2 952 (27.8)	2 910	42
Education level (X_2)	No studies (1)	106 627 (5.8)	531 (5.0)	435	96
	Primary (2)	640 753 (34.6)	3 856 (36.3)	2 383	1 473
	Professionals (3)	322 612 (17.4)	1 829 (17.2)	493	1 336
	Secondary (4)	391 488 (21.1)	2 137 (20.1)	1 087	1 050
	Medium-superior (5)	142 276 (7.7)	803 (7.6)	196	607
	Higher (6)	247 264 (13.4)	1 319 (12.4)	224	1 095
	Not available (NA)	296 (0.0)	134 (1.3)	61	73
Sex (X_3)	Male (1)	893 064 (48.2)	5 057 (47.7)	2 068	2 989
	Female (2)	958 252 (51.8)	5 552 (52.3)	2 811	2 741
Nationality (X_4)	Spanish (1)	1 726 303 (93.2)	10 175 (95.9)	4 769	5 406
	Other (2)	125 013 (6.8)	434 (4.1)	110	324

Table 3.5: Descriptive analysis of the pseudo-population of PRA survey data: the name of the variable (in brackets, the codification of those variables we use throughout this dissertation), the category (in brackets, their codification), and the number (and percentage) of the units that take each category in the pseudo-population (N (%)), and the number of units without ($N_{Y=0}$) and with ($N_{Y=1}$) the event of interest in the pseudo-population.

Variable	**Category**	N **(%)**	$N_{Y=0}$	$N_{Y=1}$
Age (X_1)	16-24 (1)	159 471 (8.7)	115 491	43 980
	25-29 (2)	96 973 (5.3)	14 840	82 133
	30-34 (3)	122 968 (6.7)	10 195	112 773
	35-44 (4)	342 660 (18.7)	24 292	318 368
	45-54 (5)	341 728 (18.7)	38 354	303 374
	55-64 (6)	297 289 (16.2)	104 986	192 303
	$\geq$ 65 (7)	469 354 (25.6)	459 274	10 080
Education level (X_2)	No studies (1)	104 763 (5.7)	78 677	26 086
	Primary (2)	635 125 (34.7)	370 111	265 014
	Professionals (3)	320 218 (17.5)	75 634	244 584
	Secondary (4)	385 499 (21.1)	172 612	212 887
	Medium-superior (5)	140 981 (7.7)	30 040	110 941
	Higher (6)	243 857 (13.3)	40 358	203 499
Sex (X_3)	Male (1)	881 410 (48.2)	331 584	549 826
	Female (2)	949 033 (51.8)	435 848	513 185
Nationality (X_4)	Spanish (1)	1 714 330 (93.7)	731 740	982 590
	Other (2)	116 113 (6.3)	35 692	80 421
Active (Y)	Inactive (0)	767 432 (41.9)		
	Active (1)	1 063 011 (58.1)		

CHAPTER 4

Estimation of logistic regression parameters

The work presented in this chapter has been accepted for publication:

Iparragirre, A., Barrio, I., Aramendi, J., & Arostegui, I. (2024) Estimation of logistic regression parameters for complex survey data: simulation study based on real survey data. SORT - Statistics and Operations Research Transactions, *in press.*

This chapter mostly replicates the above-mentioned article. However, some changes have been made to keep the notation and ensure cohesion with the rest of the document.

Summary

In the context of complex survey data, whether sampling weights should or should not be considered in the estimation process of model parameters is a question that still continues to generate much discussion among researchers in different fields. We contribute to this debate by means of a simulation study based on real survey data in the framework of logistic regression models. Three methods have been considered for estimating the coefficients of the logistic regression model: a) the unweighted model, b) the weighted model, and c) the unweighted mixed model. A simulation study has been conducted in order to study their performance. The results suggest that the performance of the weighted logistic regression model is superior, showing the importance of using sampling weights in the estimation of the model parameters.

4.1 Introduction

As introduced in Chapter 1, whether or not to use the sampling weights when fitting prediction models is a question that has been widely discussed in the literature by a number of researchers (Brewer and Mellor 1973, Smith 1981). As stated previously, different perspectives can be adopted when fitting prediction models to survey data, which are usually denoted as model- and design-based approaches (Binder and Roberts 2009, Chambers and Skinner 2003). On the one hand, the researchers that adopt the design-based perspective warn that if the complex sampling design, and in particular, the sampling weights are not considered in the estimation process of model parameters, the variances tend to be underestimated and biased estimates may be obtained (Binder and Roberts 2009, Heeringa et al. 2017). Therefore, they claim that the sampling weights should be considered in the estimation process of model parameters.

On the other hand, from a model-based point of view, if the model is well specified, the coefficient estimates must be unbiased even though the sampling weights are not considered directly in the estimation process, and considering them may increase the standard deviations of the estimates, particularly for small sample sizes (Chambers and Skinner 2003, Korn and Graubard 1995, Reiter et al. 2005, Scott and Wild 1986). In this context, Rubin (1976), Scott (1977) and Sugden and Smith (1984) established conditions under which the sampling design may be ignored for

inference purposes. As explained by Skinner et al. (1989), a condition for a design to be ignorable is to be noninformative. A sampling design is denoted as informative if the response variable is related to the sampling weights, even after considering the covariates that are going to be part of the model (Pfeffermann and Sverchkov 2009). Different methods have been proposed from the model-based perspective in order to ensure that the design is ignorable and the models are well specified (Pfeffermann and Sverchkov 2009). Researchers that adopt this perspective propose, among other techniques, to incorporate into the model as covariates all the design variables that have been considered in the sampling process and the interactions between them (see, e.g., DeMets and Halperin (1977), Gelman (2007), Nathan and Holt (1980)).

Although it was already pointed out by Chambers and Skinner (2003), the discussion between the two perspectives is still alive. Some more recent works, such as Reiter et al. (2005), Masood et al. (2016) and Lumley and Scott (2017), show that this debate still generates doubts among researchers and makes it difficult to decide whether or not to use sampling weights in their analyses. Most researchers agree that it is not advisable to ignore sampling weights if the design is informative or the model is not well specified, but at the same time, they encourage analysts to ignore the sampling weights when they are not strictly necessary. The difficulty usually lies in identifying whether or not sampling weights are necessary to estimate model parameters based on our particular survey data or, put another way, whether or not the design is informative. As explained by Pfeffermann and Sverchkov (2009), informativeness depends not only on the sampling design but also on the model that is going to be fitted, the response variable of that model, and the covariates that will be included. Therefore, commonly it is not easy to know whether the sampling design of the survey data to be analyzed is informative or not to fit a particular model. In addition, it is not always possible to include all the design variables and the interactions between them in the model due to several reasons, such as the lack of information or the large number of design variables (Pfeffermann and Sverchkov 2009). Consequently, nowadays, it is still not easy in practice to decide whether sampling weights should or not be considered for estimating model parameters. For this reason, we believe that further studies are needed in this area, and in particular, we consider that it is necessary to provide insight considering simulation studies based on real survey data as a complement to the theoretical results and case studies that have been most discussed so far.

Throughout this work, we focus on the estimation of model parameters and, in particular, on the logistic regression framework for dichotomous response variables.

Although, in general, there are more studies concerning the linear regression model (see, e.g., DeMets and Halperin (1977), Hausman and Wise (1981), Holt et al. (1980), Nathan and Holt (1980)), a number of works have also been carried out in order to address this problem arising from complex samplings in the context of logistic regression models. In particular, Scott and Wild (1986; 2002) work with simulated data inspired by a case-control study. It should be noted that case-control studies consist on stratifying the data based on the dichotomous response variable and, therefore, are always based on informative sampling designs. But, what if we do not know whether our sampling design is informative or not to fit a certain model? As mentioned above, in practice, this is the situation that usually occurs when working with real complex survey data. Chambless and Boyle (1985), Lumley and Scott (2017) and Reiter et al. (2005) raise this issue in their analysis with real survey data and they compare several estimation methods adopting both, model- and design-based perspectives and they finally select the most appropriate model for their analysis. However, how can we know in practice whether these differences in estimates are large or not, and if so, which of the estimates is the most appropriate? In this chapter, we aim to go a step further and contribute to the work that has been done in the above-mentioned papers by analyzing the differences among different methods by means of a simulation study based on real survey data in order to work under a real-life scenario that allows us to compare the coefficient estimates to the theoretical ones. Hence, data were generated based on real surveys, and a priori, whether these data are informative or not to fit different models is not known to us in advance. Our goal is to analyze by means of a simulation study a situation that frequently occurs in practice and to evaluate the consequences or the effect of making the decision to consider or not the sampling weights to estimate the coefficients of the logistic regression model in each situation. Therefore, we compare the performance of several estimation methods that are commonly applied for estimating the coefficients of the logistic regression model (see, e.g., Lumley and Scott (2017)). In particular, we compare the coefficient estimates obtained by: a) the unweighted logistic regression model (defined in Section 2.2.2), b) the weighted logistic regression model (also introduced in Section 2.2.2), and c) the unweighted logistic regression mixed model with random intercept. Different scenarios were defined based on a) data obtained from two different real surveys (specifically, ESIE and PRA surveys, which are described in Sections 3.1 and 3.2, respectively, and were obtained based on a one-stage stratified sampling design); and b) the number of covariates/parameters in the model.

The rest of the chapter is organized as follows. In Section 4.2, the methods that were applied for estimating the model parameters are described. Information about the simulation procedure, scenarios that were drawn, and the results we obtained can be found in Section 4.3. In Section 4.4, we apply the described methods to real survey data for illustration purposes. Finally, the chapter concludes with a discussion in Section 4.5.

4.2 Methods

In this section, we describe the methods we have considered in order to estimate the logistic regression coefficients for complex survey data, so let Y indicate a dichotomous response variable. Sample S is drawn from the finite population U following a one-stage stratified sampling design.

Let us remind some methodological concepts defined in Chapter 2. The finite population logistic regression model is defined as follows (also shown in eq. (2.56)):

$$\text{logit}(p(\boldsymbol{x}_i)) = \ln\left[\frac{p(\boldsymbol{x}_i)}{1-p(\boldsymbol{x}_i)}\right] = \boldsymbol{x}_i\boldsymbol{\beta}^{\text{pop}}, \tag{4.1}$$

where $p(\boldsymbol{x}_i) = P(Y=1|\boldsymbol{x}_i)$ denotes the probability of event for the unit i given the values of covariates $\boldsymbol{x}_i$ $(\forall i \in U)$ and the coefficients $\boldsymbol{\beta}^{\text{pop}} = (\beta_0^{\text{pop}}, \beta_1^{\text{pop}}, \ldots, \beta_p^{\text{pop}})^T$ are obtained by maximizing the population likelihood:

$$L_{\text{pop}}(\boldsymbol{\beta}) = \prod_{i\in U} p(\boldsymbol{x}_i)^{y_i}(1-p(\boldsymbol{x}_i))^{1-y_i}. \tag{4.2}$$

However, it should be noted that responses y_i are usually known only for the sampled units, $i \in S$. For this reason, the model should be estimated based on the sample S. Therefore, a simple logistic regression model can be fitted to the complex survey sample S, which can be defined as follows:

$$\text{logit}(p(\boldsymbol{x}_i)) = \ln\left[\frac{p(\boldsymbol{x}_i)}{1-p(\boldsymbol{x}_i)}\right] = \boldsymbol{x}_i\boldsymbol{\beta}. \tag{4.3}$$

Different methods can be applied to estimate the vector of regression coefficients $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_p)^T$ based on S. The unweighted and weighted logistic regression models have already been defined in Chapter 2. Let us remind them briefly in the following lines:

M1. Unweighted logistic regression model (unw)

Let us denote as $\hat{\boldsymbol{\beta}}_{\text{unw}}$ the coefficients estimated by maximizing the likelihood function (eq. (4.4), previously defined in eq. (2.53)), throughout this chapter:

$$L(\boldsymbol{\beta}) = \prod_{i \in S} p(\boldsymbol{x}_i)^{y_i} \left(1 - p(\boldsymbol{x}_i)\right)^{1-y_i} . \tag{4.4}$$

M2. Weighted logistic regression model (w)

Let us denote as $\hat{\boldsymbol{\beta}}_w$ the coefficient estimates obtained based on maximizing the pseudo-likelihood function (eq. (4.5), previously defined in eq. (2.58)) and considers the sampling weights w_i:

$$PL(\boldsymbol{\beta}) = \prod_{i \in S} p(\boldsymbol{x}_i)^{y_i w_i} \left(1 - p(\boldsymbol{x}_i)\right)^{(1-y_i)w_i} . \tag{4.5}$$

M3. Unweighted logistic regression model with random intercept (mix)

In addition to the above-mentioned methods, another option is to fit a mixed model considering the complex sampling design as second-level units (see, e.g. Lumley and Scott (2017), Masood et al. (2016)). In this study, in particular, we consider a random intercept model in the same way as in Lumley and Scott (2017). Note that if sample S is partitioned into non-overlapping strata as explained in Chapter 2, $\forall i \in S$, $\exists! h \in \{1, \ldots, H\} : i \in S_h$. In order to ease the notation, let $i(h)$ indicate that $i \in S_h$, and $\boldsymbol{x}_{i(h)}$ and $y_{i(h)}$ be the values of the vector of covariates and response variable for $i \in S_h$, respectively. Then, we aim to fit the following model to our sample S:

$$\text{logit}(p_{i(h)}) = \ln\left(\frac{p_{i(h)}}{1 - p_{i(h)}}\right) = \boldsymbol{x}_{i(h)}\boldsymbol{\gamma} + u_h, \quad u_h \sim N(0, \sigma_u^2). \tag{4.6}$$

where $p_{i(h)} = P(Y = 1 | \boldsymbol{x}_{i(h)}, u_h) = \dfrac{e^{\boldsymbol{x}_{i(h)}\boldsymbol{\gamma} + u_h}}{1 + e^{\boldsymbol{x}_{i(h)}\boldsymbol{\gamma} + u_h}}$.

In this case, the likelihood function is defined as follows:

$$L_{\text{mix}}(\boldsymbol{\gamma}, \sigma_u^2) = \prod_{h=1}^{H} \int_{-\infty}^{+\infty} f(y_{i(h)} | \boldsymbol{x}_{i(h)}, u_h) f(u_h) \, \mathrm{d}u_h \, , \tag{4.7}$$

where

$$f(y_{i(h)} | \boldsymbol{x}_{i(h)}, u_h) = \prod_{i \in S_h} p_{i(h)}^{y_{i(h)}} (1 - p_{i(h)})^{1 - y_{i(h)}},$$

and

$$f(u_h) = \frac{1}{\sigma_u\sqrt{2\pi}} e^{-u_h^2/2\sigma_u^2}.$$

The parameters $\boldsymbol{\gamma}$ and σ_u^2 are commonly estimated by maximizing the likelihood function in eq. (4.7) numerically, usually by means of Laplace approximation (Lee and Nelder 2001). Let us denote as $\hat{\boldsymbol{\gamma}}$ and $\hat{\sigma}_u^2$ those estimates, respectively, hereinafter.

However, the comparison of the coefficients obtained from conditional random effect models and the corresponding marginal models is not straightforward (Lee and Nelder 2004). In the case of logistic random intercept models, marginal coefficients $\boldsymbol{\beta}$ can be obtained based on conditional parameters $\boldsymbol{\gamma}$ as follows:

$$\boldsymbol{\beta} = \frac{\boldsymbol{\gamma}}{\sqrt{1 + \tilde{c}^2\sigma_u^2}}, \tag{4.8}$$

where $\tilde{c} = (16\sqrt{3})/(15\pi)$ (Diggle et al. 2002). Let us denote as $\hat{\boldsymbol{\beta}}_{\text{mix}}$ the coefficient estimates obtained based on $\hat{\boldsymbol{\gamma}}$ and $\hat{\sigma}_u^2$.

The goal is to analyze the performance of the above-mentioned methods by comparing the estimates $\hat{\boldsymbol{\beta}}_{\text{unw}}$, $\hat{\boldsymbol{\beta}}_w$ and $\hat{\boldsymbol{\beta}}_{\text{mix}}$ (which are commonly considered for estimating the coefficients of the logistic regression models in the context of complex survey data, e.g., Lumley and Scott (2017)) to the true finite population coefficients $\boldsymbol{\beta}^{\text{pop}}$. Note that, as stated above, in practice, the responses y_i are usually known only for the sampled units, i.e., $\forall i \in S$. Nevertheless, when a finite population is simulated, the responses y_i are known in both, the finite population U and the sample S, and thus, the true finite population coefficients $\boldsymbol{\beta}^{\text{pop}}$ and the estimates $\hat{\boldsymbol{\beta}}_{\text{unw}}$, $\hat{\boldsymbol{\beta}}_w$ and $\hat{\boldsymbol{\beta}}_{\text{mix}}$ can be compared. In the following section, we explain the simulation study process in detail.

4.3 Simulation study

In this section, we describe the simulation study that we have conducted in order to analyze the behavior of the estimation methods described in Section 4.2 for estimating the coefficients of the logistic regression model based on complex survey data under different scenarios. As mentioned previously, our goal is to compare the coefficient estimates to the true finite population coefficients in real data-based scenarios.

In Section 4.3.1 the simulation process is described in detail and in Section 4.3.2 the results obtained in the simulation study are shown.

4.3.1 Scenarios and set up

In this section, we describe the different scenarios under which the simulation study has been conducted and the steps that have been followed. The simulation process is described below, step by step:

Step 1. Generate the pseudo-population U of N units from the set of random variables $(Y, \boldsymbol{X})$: $\{(y_i, \boldsymbol{x}_i)\}_{i=1}^{N}$.

Step 2. Compute $\boldsymbol{\beta}^{\text{pop}}$ by maximizing the population likelihood in eq. (4.2).

For $r = 1, \ldots, R$ repeat the following steps:

Step 3. Obtain a sample $S^r \subset U$ by one-stage stratified sampling and assign the corresponding sampling weights $w_i, \forall i \in S^r$.

Step 4. Fit the models to S^r by the likelihood functions in eqs. (4.4), (4.5) and (4.7) and obtain $\hat{\boldsymbol{\beta}}^r_{\text{unw}}$, $\hat{\boldsymbol{\beta}}^r_{w}$ and $\hat{\boldsymbol{\beta}}^r_{\text{mix}}$, respectively.

Finally, for the results obtained based on samples $r = 1, \ldots, R$ and for each method m, $\forall m \in \{\text{unw}, w, \text{mix}\}$, let us define the bias of the coefficient vector estimates as shown in eq. (4.9). Note that all the covariates considered in this simulation study are categorical and one coefficient was estimated for each category, except for the one considered as reference category. Thus, let p indicate the number of covariates included in the model and $p' > p$ the total number of model parameters (including the intercept). Then,

$$\text{bias}^r_j = \hat{\beta}^r_{j,m} - \beta^{\text{pop}}_j, \quad \forall j = 0, 1, \ldots, (p' - 1). \tag{4.9}$$

The average bias (AvBias) and the mean squared error (MSE) across $\forall r = 1, \ldots, R$ are defined in eqs. (4.10) and (4.11), respectively:

$$\text{AvBias}_j = \frac{1}{R}\sum_{r=1}^{R}\left(\text{bias}^r_j\right) = \frac{1}{R}\sum_{r=1}^{R}\left(\hat{\beta}^r_{j,m} - \beta^{\text{pop}}_j\right), \quad \forall j = 0, 1, \ldots, (p' - 1), \tag{4.10}$$

$$\text{MSE}_j = \frac{1}{R}\sum_{r=1}^{R}\left(\text{bias}^r_j\right)^2 = \frac{1}{R}\sum_{r=1}^{R}\left(\hat{\beta}^r_{j,m} - \beta^{\text{pop}}_j\right)^2, \quad \forall j = 0, 1, \ldots, (p' - 1). \tag{4.11}$$

Two scenarios have been defined based on the two real surveys described in Chapter 3, ESIE (Scenario 1, hereinafter) and PRA (Scenario 2, hereinafter). One finite pseudo-population was generated based on each of the surveys (as indicated in **Step 1.**). Those populations were sampled based on the complex sampling designs that were applied by EUSTAT in the corresponding real surveys (indicated in **Step 3.**). The pseudo-population was generated and the samples were obtained as described in Chapter 3. In particular, in Scenario 1 related to the ESIE survey data, all the establishments were considered. A total of $R = 500$ samples were obtained from each pseudo-population.

In addition, two different models were fitted to the finite population as well as to the samples for each of the surveys with different numbers of covariates (**Step 2**). In particular, in Scenario 1, models with $p = 1$ covariate (X_1, $p' = 7$ parameters including the intercept β_0) and $p = 3$ covariates (X_1, X_2 and X_3, $p' = 14$ parameters) were fitted. In the same way, in Scenario 2, the models were fitted with $p = 1$ covariate (X_1, $p' = 7$ parameters) and $p = 4$ covariates (X_1, X_2, X_3 and X_4, $p' = 14$ parameters). Those readers interested in the meaning of the covariates can go back to Chapter 3 (in particular, Tables 3.2 and 3.4) where the codification implemented in the descriptive analysis of both datasets is explained in detail.

All computations were performed in (64 bit) R 4.0.5 (R Core Team, 2021) and a workstation equipped with 32GB of RAM, an Intel i7-8700 processor (3.20 Ghz) and Windows 10 operating system. In particular, the unweighted logistic regression models (unw) were fitted by means of the `glm()` function from the `stats` package, the weighted logistic regression models (w) by means of the `svyglm()` from the `survey` package (Lumley 2020) and the unweighted mixed models with random intercept (mix) by the `glmer()` of the `lme4` package (Bates et al. 2015).

4.3.2 Results

In this section, we describe the results we obtained in both scenarios: Scenario 1 (which is based on the ESIE survey) and Scenario 2 (which is based on the PRA survey). As explained in Section 4.3.1, in each scenario, two models were fitted with different numbers of covariates. Our goal is to compare the estimates obtained based on the three coefficient estimation methods described in Section 4.2 (which are the unweighted estimates obtained based on the unweighted logistic regression (unw), the weighted estimates based on the weighted logistic regression (w) and the mixed estimates for the unweighted logistic regression with random intercept (mix)) to the

finite population coefficients ($\boldsymbol{\beta}^{\text{pop}}$), in terms of bias and MSE.

Due to the large number of results obtained, we begin by summarizing the main findings. When comparing the performance of the three methods in each scenario, we observe that the results differ depending on the scenario. In Scenario 1, the weighted estimates, in general, outperform the unweighted and mixed ones in terms of bias and MSE, while the weighted estimates had a greater variance than the others. On the other hand, in Scenario 2, there are no differences among the results obtained with the three methods. The results also show that the weighted method performs correctly in both scenarios and the results are quite similar in terms of bias (which is negligible in all scenarios) and MSE. However, the performance of the unweighted and mixed models in terms of bias (and consequently, also in terms of MSE) differ depending on the scenario, being much lower in Scenario 2 than in Scenario 1. We proceed below to analyze the graphical and numerical results related to each scenario.

Figure 4.1 depicts the box-plots of the bias of the unweighted (unw), weighted (w) and mixed (mix) estimates for the models with $p = 1$ (Figure 4.1 (a)) and $p = 3$ (Figure 4.1 (b)) covariates in Scenario 1. As can be observed, the weighted estimates are the ones that perform the best in terms of bias in both models, with either $p = 1$ or $p = 3$ covariates. This can also be observed in Table 4.1. This table describes the numerical results of the mean, standard deviation, average bias, and MSE of those estimates, as well as the true finite population coefficients in Scenario 1 for the models with $p = 1$ and $p = 3$ covariates, respectively. As can be seen, while the weighted estimates are quite similar to the population coefficients ($\boldsymbol{\beta}^{\text{pop}}$, which leads to low average biases for this method), the unweighted and mixed estimates differ considerably. In the estimates obtained for the model with $p = 1$, for example, for the coefficient $\beta_{1,6}$ for instance, the average bias obtained by the weighted model is -0.095, which is considerably lower than the one of the unweighted model (0.378) and the mixed model (-1.377). It can also be observed that the average bias decreases for all the methods (and most notably for unweighted and mixed ones) when $p = 3$ covariates are included in the model. In particular, the average bias of the coefficient estimates related to the category $\beta_{1,6}$ decreases to 0.050 for the unweighted estimates, to 0.007 for the weighted, and to -0.771 for the mixed ones in the model with $p = 3$ covariates.

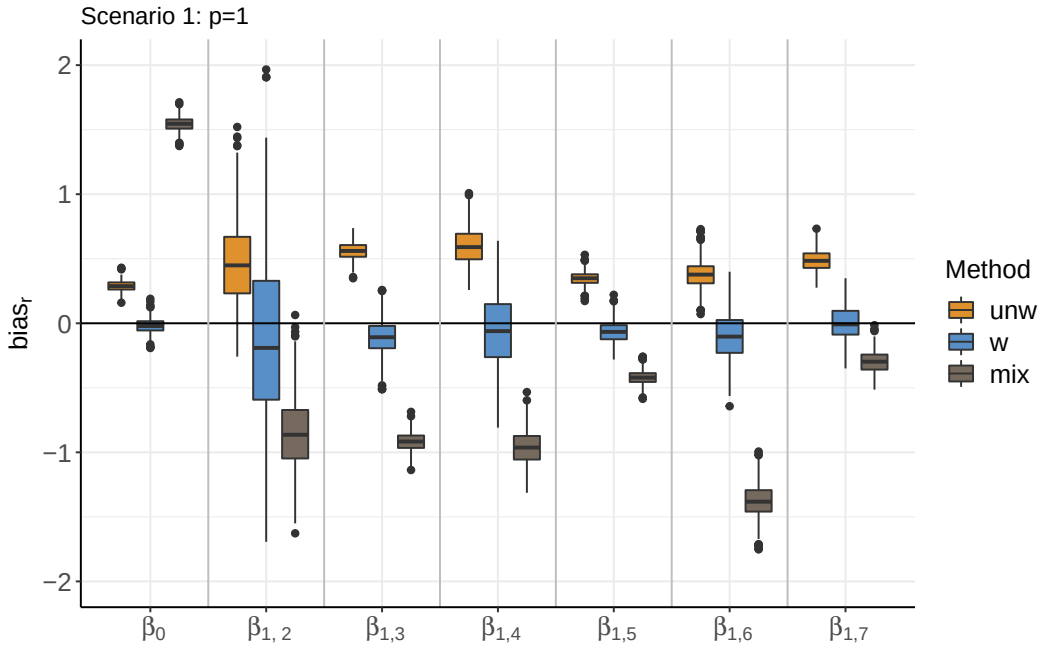
In Figure 4.1, it can also be seen that the variability of the weighted estimates is the greatest one in comparison to the rest. This is also shown in Table 4.1, where the standard deviations of these estimates can be up to twice as large as that of the

unweighted and mixed ones. For example, the standard deviations corresponding to the estimates of $\beta_{1,3}$ (for $p = 1$) are 0.063, 0.132 and 0.070 for unweighted, weighted and mixed models, respectively. The source of variability could also be related to data. It is especially remarkable the variability of the estimates of the coefficient $\beta_{1,2}$ for all the methods, in general, and most importantly for the weighted ones. It should be noted that there are very few units in category 2 of the covariate X_1 in Scenario 1. In particular, 450 units in the simulated population (0.2% of the total of units in the finite population) take this category on that covariate, and in the samples, this amount varies from 27 (0.3%) to 53 (0.7%) (similar to the distribution shown for the real finite population in Table 3.2 in Chapter 3). This may be affecting the estimates of the parameter $\beta_{1,2}$, specifically for the weighted model. The behavior of the estimates of $\beta_{1,4}$ could be explained in the same way, for which a greater variability is also observed, especially for the weighted ones (2008 units (1.0%) in the simulated finite population, from 178 (2.5%) to 232 (3.2%) in the samples). In addition, in Table 4.1, it should also be noted that for all the methods, the standard deviations are slightly greater for the model with $p = 3$ than for the one with $p = 1$ covariates.

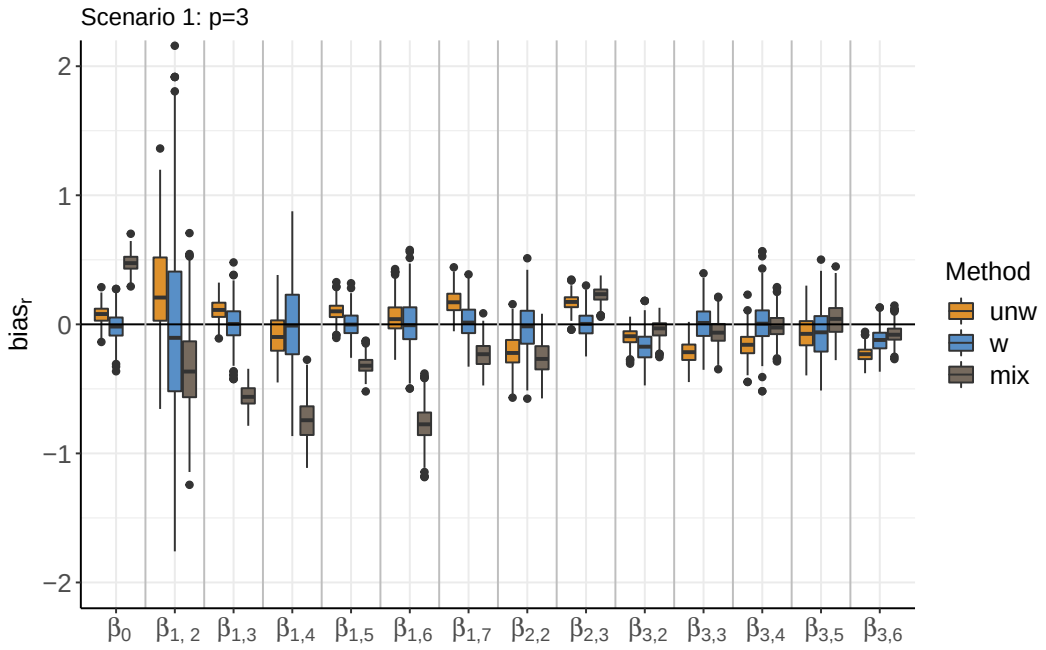
Finally, as shown in Table 4.1, the mixed estimates are, in most cases, the ones with the greatest MSE because of their large bias. For instance, the MSE of the coefficient corresponding to the category $\beta_{1,4}$ in the model with $p = 1$ is 0.942 for the mixed model, while for the weighted and unweighted models, the MSE are 0.085 and 0.378, respectively. Given that the bias decreases while adding covariates for the unweighted and mixed models, the MSE also decreases in the same way. For the same coefficient, when $p = 3$, the MSE related to the mixed estimates decreases to 0.576. The MSE of the weighted estimates is quite similar in both models, with $p = 1$ and $p = 3$ covariates. Comparing the MSE of weighted and unweighted estimates, it can be observed that the MSE of the unweighted estimates is greater when $p = 1$. However, in Scenario 1 with $p = 3$, there are no differences in terms of MSE between weighted and unweighted estimates due to the larger variability of the weighted estimates despite their smaller bias.

Figure 4.2 depicts the box-plots of the bias of the unweighted, weighted and mixed estimates for the models with $p = 1$ and $p = 4$ covariates in Scenario 2. In this case, as shown in Figure 4.2, the performance of the three methods is quite similar in terms of bias and variability. The differences are not considerable, neither among the different methods nor between the different models (fitted with $p = 1$ and $p = 4$ covariates). Table 4.2 describes the numerical results of the mean, standard

deviation, average bias, and MSE of those estimates and the true finite population coefficients for $p = 1$ and $p = 4$ in Scenario 2. The average bias is very low for all the methods and in both models, either with $p = 1$ or $p = 4$ covariates. The largest observed average bias is -0.067, which corresponds to the coefficient $\beta_{4,2}$ of the weighted model with $p = 4$. The variability of the weighted estimates is usually slightly greater than that of the rest of the methods. However, as noted above, those differences are very small. The greatest difference in terms of the standard deviation of the estimates and MSE is observed in the model with $p = 4$ for the coefficient estimates corresponding to $\beta_{2,5}$. The standard deviation of the estimates obtained by means of the weighted model is 0.185, while the ones corresponding to the unweighted and mixed models are 0.174. In the same way, the MSE of the weighted model for this coefficient is 0.035, while for the unweighted and mixed models is 0.030. It can be concluded that all the studied methods perform properly to estimate the finite population model coefficients in Scenario 2.

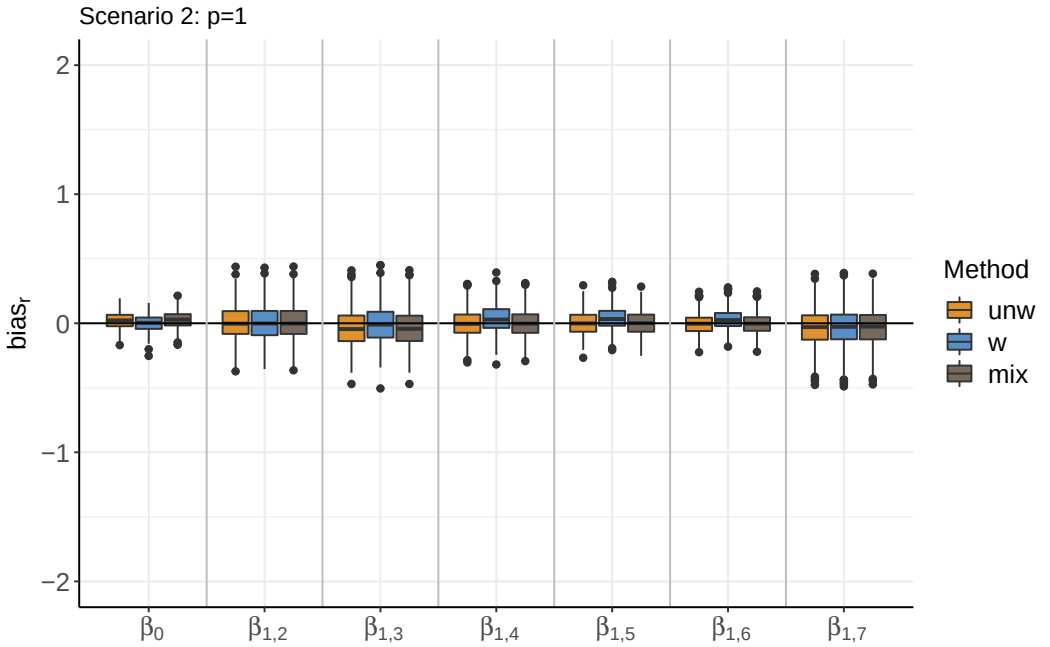


(a)

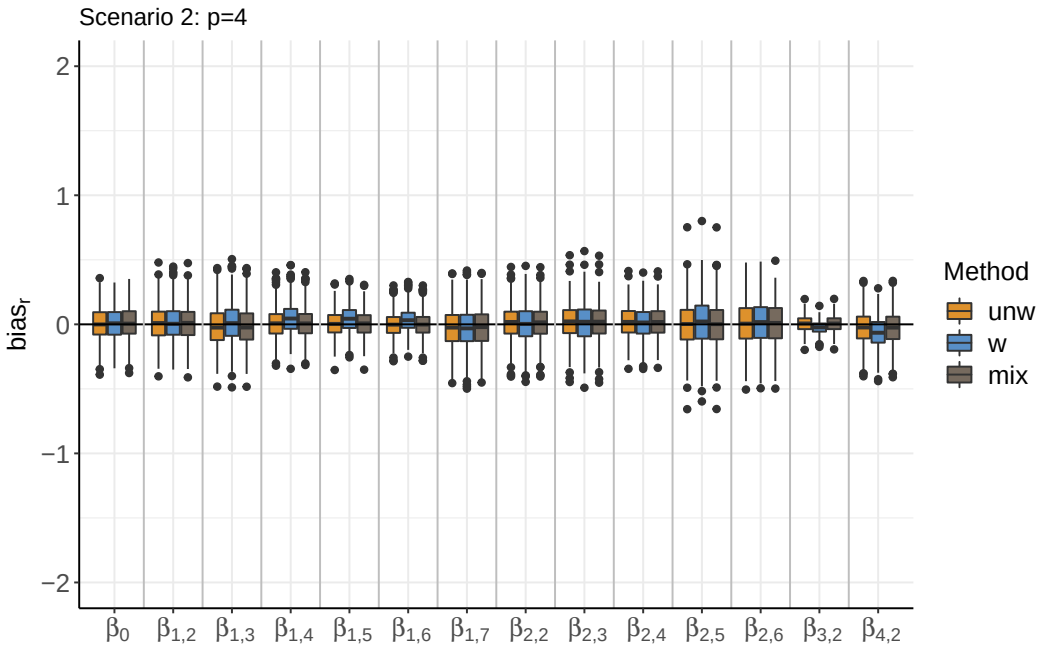


(b)

Figure 4.1: Box-plots of the bias of the estimates obtained by the methods unw, w, and mix for the coefficients in the models with (a) $p = 1$ ($p' = 7$) and (b) $p = 3$ ($p' = 14$) covariates in Scenario 1, $\forall r = 1, \ldots, R$.



(a)



(b)

Figure 4.2: Box-plots of the bias of the estimates obtained by the methods unw, w, and mix for the coefficients in the models with (a) $p = 1$ ($p' = 7$) and (b) $p = 4$ ($p' = 14$) covariates in Scenario 2, $\forall r = 1, \ldots, R$.

Table 4.1: True finite population model coefficients ($\boldsymbol{\beta}^{\text{pop}}$) and the average (mean), standard deviation (sd), average bias (AvBias) and MSE of the estimates obtained by the unw, w and mix methods for the models with $p = 1$ and $p = 3$ covariates ($p' = 7$ and $p' = 14$ parameters, respectively) in Scenario 1 for $R = 500$ samples.

	$\boldsymbol{\beta}^{\text{pop}}$	$\hat{\boldsymbol{\beta}}_{\text{unw}}$			$\hat{\boldsymbol{\beta}}_{\boldsymbol{w}}$			$\hat{\boldsymbol{\beta}}_{\text{mix}}$		
		Mean (sd)	**AvBias**	**MSE**	**Mean (sd)**	**AvBias**	**MSE**	**Mean (sd)**	**AvBias**	**MSE**
p = 1 (p′ = 7)										
β_0	-1.015	-0.727 (0.043)	0.288	0.085	-1.035 (0.058)	-0.021	0.004	0.529 (0.056)	1.544	2.387
$\beta_{1,2}$	1.184	1.658 (0.325)	0.474	0.330	1.050 (0.659)	-0.134	0.451	0.334 (0.290)	-0.850	0.807
$\beta_{1,3}$	1.360	1.920 (0.063)	0.560	0.318	1.252 (0.132)	-0.108	0.029	0.443 (0.070)	-0.916	0.845
$\beta_{1,4}$	1.342	1.942 (0.138)	0.600	0.378	1.283 (0.286)	-0.060	0.085	0.382 (0.141)	-0.960	0.942
$\beta_{1,5}$	0.537	0.885 (0.055)	0.348	0.124	0.471 (0.085)	-0.067	0.012	0.118 (0.053)	-0.420	0.179
$\beta_{1,6}$	1.908	2.286 (0.110)	0.378	0.155	1.813 (0.184)	-0.095	0.043	0.531 (0.129)	-1.377	1.912
$\beta_{1,7}$	0.470	0.956 (0.080)	0.487	0.243	0.471 (0.121)	0.001	0.015	0.174 (0.088)	-0.296	0.095
p = 3 (p′ = 14)										
β_0	-0.959	-0.883 (0.069)	0.076	0.010	-0.979 (0.111)	-0.020	0.013	-0.484 (0.066)	0.475	0.230
$\beta_{1,2}$	0.602	0.850 (0.356)	0.248	0.188	0.582 (0.668)	-0.020	0.445	0.257 (0.330)	-0.344	0.227
$\beta_{1,3}$	0.824	0.937 (0.078)	0.112	0.019	0.830 (0.146)	0.006	0.021	0.269 (0.082)	-0.556	0.316
$\beta_{1,4}$	0.926	0.840 (0.158)	-0.086	0.032	0.926 (0.310)	0.000	0.096	0.184 (0.161)	-0.742	0.576
$\beta_{1,5}$	0.382	0.482 (0.066)	0.101	0.014	0.383 (0.096)	0.002	0.009	0.065 (0.064)	-0.317	0.104
$\beta_{1,6}$	1.145	1.195 (0.124)	0.050	0.018	1.153 (0.197)	0.007	0.039	0.374 (0.138)	-0.771	0.614
$\beta_{1,7}$	0.355	0.526 (0.089)	0.172	0.037	0.373 (0.125)	0.018	0.016	0.119 (0.097)	-0.235	0.065
$\beta_{2,2}$	-0.630	-0.843 (0.136)	-0.212	0.064	-0.647 (0.182)	-0.016	0.033	-0.892 (0.121)	-0.262	0.083
$\beta_{2,3}$	0.036	0.207 (0.058)	0.170	0.032	0.040 (0.098)	0.004	0.010	0.266 (0.056)	0.230	0.056
$\beta_{3,2}$	0.042	-0.055 (0.065)	-0.097	0.014	-0.128 (0.115)	-0.171	0.042	0.005 (0.067)	-0.038	0.006
$\beta_{3,3}$	1.514	1.297 (0.086)	-0.217	0.054	1.520 (0.143)	0.006	0.020	1.448 (0.091)	-0.066	0.013
$\beta_{3,4}$	1.418	1.260 (0.096)	-0.158	0.034	1.428 (0.150)	0.010	0.023	1.402 (0.098)	-0.016	0.010
$\beta_{3,5}$	1.564	1.491 (0.134)	-0.073	0.023	1.497 (0.190)	-0.067	0.040	1.601 (0.133)	0.037	0.019
$\beta_{3,6}$	2.700	2.465 (0.053)	-0.234	0.058	2.577 (0.093)	-0.123	0.024	2.623 (0.069)	-0.077	0.011

Table 4.2: True finite population model coefficients ($\boldsymbol{\beta}^{\text{pop}}$) and the average (mean), standard deviation (sd), average bias (AvBias) and MSE of the estimates obtained by the unw, w and mix methods for the models with $p = 1$ and $p = 4$ covariates ($p' = 7$ and $p' = 14$ parameters, respectively in Scenario 2 for $R = 500$ samples.

	$\boldsymbol{\beta}^{\text{pop}}$	$\hat{\boldsymbol{\beta}}_{\text{unw}}$			$\hat{\boldsymbol{\beta}}_{\boldsymbol{w}}$			$\hat{\boldsymbol{\beta}}_{\text{mix}}$		
		Mean (sd)	**AvBias**	**MSE**	**Mean (sd)**	**AvBias**	**MSE**	**Mean (sd)**	**AvBias**	**MSE**
p = 1 (p′ = 7)										
β_0	-0.965	-0.945 (0.068)	0.020	0.005	-0.965 (0.065)	0.000	0.004	-0.941 (0.068)	0.025	0.005
$\beta_{1,2}$	2.676	2.681 (0.127)	0.005	0.016	2.680 (0.133)	0.003	0.018	2.682 (0.127)	0.006	0.016
$\beta_{1,3}$	3.369	3.337 (0.149)	-0.032	0.023	3.366 (0.152)	-0.003	0.023	3.338 (0.149)	-0.031	0.023
$\beta_{1,4}$	3.539	3.539 (0.110)	0.000	0.012	3.576 (0.109)	0.038	0.013	3.539 (0.109)	0.001	0.012
$\beta_{1,5}$	3.034	3.035 (0.093)	0.002	0.009	3.071 (0.091)	0.037	0.010	3.036 (0.093)	0.003	0.009
$\beta_{1,6}$	1.571	1.568 (0.082)	-0.003	0.007	1.600 (0.075)	0.029	0.006	1.569 (0.082)	-0.002	0.007
$\beta_{1,7}$	-2.854	-2.886 (0.139)	-0.033	0.020	-2.883 (0.146)	-0.029	0.022	-2.884 (0.139)	-0.030	0.020
p = 4 (p′ = 14)										
β_0	-1.529	-1.523 (0.125)	0.006	0.016	-1.524 (0.127)	0.005	0.016	-1.516 (0.125)	0.012	0.016
$\beta_{1,2}$	2.478	2.490 (0.134)	0.012	0.018	2.490 (0.140)	0.012	0.020	2.490 (0.134)	0.012	0.018
$\beta_{1,3}$	3.206	3.188 (0.153)	-0.017	0.024	3.221 (0.156)	0.015	0.024	3.189 (0.153)	-0.017	0.024
$\beta_{1,4}$	3.329	3.342 (0.121)	0.013	0.015	3.379 (0.122)	0.050	0.017	3.341 (0.121)	0.013	0.015
$\beta_{1,5}$	2.780	2.785 (0.103)	0.005	0.011	2.824 (0.103)	0.044	0.012	2.786 (0.103)	0.006	0.011
$\beta_{1,6}$	1.379	1.376 (0.097)	-0.003	0.009	1.413 (0.092)	0.034	0.010	1.378 (0.097)	-0.001	0.009
$\beta_{1,7}$	-2.943	-2.974 (0.149)	-0.031	0.023	-2.974 (0.157)	-0.031	0.026	-2.970 (0.150)	-0.028	0.023
$\beta_{2,2}$	0.702	0.715 (0.134)	0.013	0.018	0.704 (0.144)	0.002	0.021	0.714 (0.134)	0.012	0.018
$\beta_{2,3}$	1.391	1.407 (0.140)	0.016	0.020	1.404 (0.148)	0.013	0.022	1.406 (0.140)	0.015	0.020
$\beta_{2,4}$	0.811	0.827 (0.117)	0.016	0.014	0.819 (0.126)	0.008	0.016	0.827 (0.117)	0.016	0.014
$\beta_{2,5}$	1.620	1.618 (0.174)	-0.001	0.030	1.638 (0.185)	0.018	0.035	1.618 (0.174)	-0.001	0.030
$\beta_{2,6}$	1.664	1.669 (0.159)	0.005	0.025	1.678 (0.165)	0.014	0.027	1.670 (0.159)	0.006	0.025
$\beta_{3,2}$	-0.427	-0.422 (0.057)	0.005	0.003	-0.451 (0.049)	-0.024	0.003	-0.421 (0.057)	0.005	0.003
$\beta_{4,2}$	-0.481	-0.506 (0.129)	-0.025	0.017	-0.548 (0.123)	-0.067	0.019	-0.508 (0.129)	-0.027	0.017

4.4 Application to the real data sets

In this section, we apply the methods described in Section 4.2 to the real survey data described in Chapter 3. The goal is to compare the coefficient estimates obtained by means of the different methods among them. Note that in this case, the real finite population coefficients are not known for us, given that we are working with real survey data, and hence, the information of the response variable is not available for all the units in the finite population.

One model was fitted to each of the surveys. In particular, we fitted the model with three covariates ($p = 3$) to the ESIE survey and the model with four covariates ($p = 4$) to the PRA survey. Those covariates are the ones that were considered in the simulation study for both surveys and are also considered in the models that are applied in practice by EUSTAT. Note that a descriptive analysis of those covariates is available in Tables 3.2 and 3.4. To fit those models, the three methods described in Section 4.2 were applied: the unweighted logistic regression (unw), the weighted logistic regression (w), and the unweighted logistic regression with random intercept (mix). Table 4.3 and Table 4.4 depict the coefficient estimates and their standard errors obtained for models fitted to the ESIE and PRA surveys, respectively.

As shown in Table 4.3, the coefficient estimates, as well as their standard errors, obtained by means of the three above-mentioned methods differ considerably in the ESIE survey. It should be noted that these differences in the estimations and their standard errors could lead to considerable differences in the Wald statistic, defined as the fraction among those parameters. However, in this case, those differences did not affect the significance of the model parameters, and all of them are statistically significant (results not shown). The largest standard errors are, in most of the cases, the ones obtained by means of the weighted logistic regression. In addition, the standard errors related to the coefficient $\beta_{1,2}$ are larger than any other, which is in line with the large variability observed in the simulation study for this coefficient (in Scenario 1). Based on the results obtained in the simulation study, we may conclude that the weighted model would be the preferred one in this case.

In contrast, the coefficient estimates and their standard errors obtained for the PRA survey are very similar among them, as can be observed in Table 4.4. This is also in line with the results observed in the simulation study (in Scenario 2). As expected, the standard errors of the weighted estimates are usually slightly greater than the rest, although there are not great differences, in general.

Table 4.3: Coefficient estimates (Estimate) and their standard errors (SE) obtained by means of the methods unw, w, and mix for the ESIE survey with $p = 3$ covariates.

	ESIE survey					
	$\hat{\boldsymbol{\beta}}_{\mathbf{unw}}$		$\hat{\boldsymbol{\beta}}_{\boldsymbol{w}}$		$\hat{\boldsymbol{\beta}}_{\mathbf{mix}}$	
	Estimate	**SE**	**Estimate**	**SE**	**Estimate**	**SE**
β_0	-2.261	0.097	-2.482	0.133	-2.217	0.140
$\beta_{1,2}$	1.892	0.338	1.293	0.444	1.697	0.368
$\beta_{1,3}$	2.490	0.107	2.718	0.161	2.337	0.119
$\beta_{1,4}$	2.248	0.196	2.577	0.299	2.151	0.215
$\beta_{1,5}$	1.550	0.084	1.721	0.111	1.458	0.094
$\beta_{1,6}$	2.260	0.146	2.544	0.206	2.092	0.181
$\beta_{1,7}$	1.341	0.103	1.130	0.133	1.197	0.119
$\beta_{2,2}$	-0.774	0.148	-0.613	0.189	-0.883	0.329
$\beta_{2,3}$	0.453	0.073	0.358	0.107	0.538	0.123
$\beta_{3,2}$	0.669	0.069	0.632	0.097	0.750	0.077
$\beta_{3,3}$	0.996	0.096	0.965	0.132	1.124	0.134
$\beta_{3,4}$	1.479	0.114	1.452	0.152	1.698	0.149
$\beta_{3,5}$	2.230	0.182	2.205	0.241	2.461	0.209
$\beta_{3,6}$	2.454	0.143	2.532	0.151	2.787	0.195

Table 4.4: Coefficient estimates (Estimate) and their standard errors (SE) obtained by means of the methods unw, w, and mix for the PRA survey with $p = 4$ covariates.

	PRA survey					
	$\hat{\boldsymbol{\beta}}_{\mathbf{unw}}$		$\hat{\boldsymbol{\beta}}_{\boldsymbol{w}}$		$\hat{\boldsymbol{\beta}}_{\mathbf{mix}}$	
	Estimate	**SE**	**Estimate**	**SE**	**Estimate**	**SE**
β_0	-2.039	0.176	-2.040	0.171	-2.037	0.179
$\beta_{1,2}$	2.508	0.164	2.523	0.172	2.515	0.164
$\beta_{1,3}$	3.106	0.179	3.105	0.191	3.113	0.179
$\beta_{1,4}$	3.191	0.121	3.292	0.126	3.194	0.122
$\beta_{1,5}$	2.836	0.114	2.934	0.118	2.835	0.114
$\beta_{1,6}$	1.455	0.103	1.543	0.108	1.454	0.103
$\beta_{1,7}$	-3.170	0.184	-3.102	0.199	-3.182	0.184
$\beta_{2,2}$	1.005	0.174	0.899	0.177	1.016	0.174
$\beta_{2,3}$	1.689	0.178	1.587	0.182	1.700	0.179
$\beta_{2,4}$	1.167	0.171	1.056	0.170	1.170	0.172
$\beta_{2,5}$	2.123	0.207	1.970	0.227	2.128	0.208
$\beta_{2,6}$	2.357	0.192	2.177	0.201	2.360	0.193
$\beta_{3,2}$	-0.596	0.063	-0.546	0.067	-0.596	0.063
$\beta_{4,2}$	0.547	0.158	0.530	0.190	0.551	0.159

4.5 Discussion

In this work, we compared the performance of three different methods to estimate model coefficients in the logistic regression framework for complex survey data by means of a simulation study based on real survey data. In general, the results obtained are in line with the ones observed in related works, based on either logistic (Chambless and Boyle 1985, Lumley and Scott 2017, Reiter et al. 2005, Scott and Wild 1986; 2002) or linear regression framework (DeMets and Halperin 1977, Holt et al. 1980, Nathan and Holt 1980, Smith 1981). Nevertheless, there are also some differences between this work and the above-mentioned studies. We proceed to comment on these similarities and differences in the following lines.

One of the greatest differences between this study and the ones mentioned previously is that this work is a simulation study based on real survey data. The objective has been to work in a realistic scenario that allows us to compare the results we obtain to the true coefficients of the finite population models. Thus, data for the simulation study have been simulated based on two real surveys conducted by EUSTAT. In both surveys, the finite population was sampled by one-stage stratified sampling. However, the strata were defined in very different ways, as explained in Chapter 3. In the ESIE survey, the strata were defined by means of the combination of three categorical variables with many categories, resulting in a total of 515 non-empty small strata. On the other hand, in the PRA survey, strata were defined by means of the region to which each individual belongs, which leads to 23 different strata. In addition to the sampling design, the impact of the number of covariates included in the model and the number of parameters were also analyzed. It should be noted that in this simulation study the theoretical model from which the finite population is generated is not known for us. Thus, we compare the model estimates obtained based on the methods under study to the true coefficient values obtained by fitting the model to the finite population.

The main conclusions of this study are that the weighted logistic regression (w) performed properly in both scenarios, and the estimates we obtained were unbiased. In contrast, the behavior of the unweighted logistic regression (unw) and the unweighted logistic regression with random intercept (mix) depended on the scenario and on the number of covariates/parameters estimated in the model. In the scenario related to the ESIE survey, unlike in the scenario based on the PRA survey, biased estimates were obtained with unweighted and mixed models. These results are in line with Holt et al. (1980), Nathan and Holt (1980), Scott and Wild (1986) among

others, which also warn about the bias of the unweighted coefficient estimates in both, linear and logistic regression frameworks. Scott and Wild (1986) claim that the bias of the unweighted coefficient estimates is smaller when the model fitted to the sample is exactly the same as the true theoretical model from which the data is derived than when the model fitted is "reasonable but not perfect". As mentioned previously, the theoretical model from which the finite population is generated is not known for us. Nevertheless, in this study, we have also observed that the bias becomes smaller when more covariates are included in the model, which would be in line with the results obtained in the above-mentioned studies. However, this bias is still larger than the bias obtained by means of the weighted logistic regression. For this reason, the message we aim to transmit with this work is the recommendation of fitting weighted models. However, it should also be noted that in line with Reiter et al. (2005), we agree that comparing the estimates obtained with the unweighted and weighted models can help to detect if the model is well specified (and improve the model if the needed variables are available) since a large difference between the two estimates can suggest that the fitted model is misspecified.

The variability of the estimates obtained by the weighted logistic regression model is greater than that of the estimates obtained by means of the unweighted logistic regression model (with and/or without random intercept), which is in line with Chambless and Boyle (1985), Lumley and Scott (2017) and Scott and Wild (1986). These differences are not very large in most cases. However, we have observed that when there are few individuals in a particular category of a categorical variable, then the variability of the weighted estimates of the coefficient corresponding to that category can be much greater than the unweighted ones. We conclude that we should be careful when we have categorical variables with an imbalanced distribution of individuals in the categories. It should be noted that in the simulation study we have conducted, it was unfeasible to put all the design information as a fixed effect (as recommended for strata) because of the problems that would arise for both model estimation and interpretation. For this reason, we have opted to use the strata as a random effect. Through this study, we have been able to verify that the mixed model does not provide us with advantages compared to the other models.

We also applied the three methods under study to real survey data and the estimates we obtained are in line with the results observed in the simulation study. On the one hand, in the PRA survey, the estimates are quite similar among them, and there are not many differences between the standard errors of these estimates,

which leads us to conclude that all the studied methods work properly in this case. On the other hand, in the ESIE survey, there are many differences in the estimates of the parameters among different methods. Observing the similarities between the simulation study and the application to real data sets, and taking into account that those results are also in line with the results obtained in similar empirical studies, such as Chambless and Boyle (1985) and Lumley and Scott (2017), we can assume that the weighted logistic regression would be preferred when working with ESIE survey data.

We now proceed to comment on the limitations of this study. First of all, in this simulation study, we are unable to know which is the theoretical model from which the data is derived due to the fact that we aimed for the simulation study to be based on real survey data and hence, we have focused on comparing the estimates obtained based on the samples with the true coefficients of the model fitted to the finite population. It should be noted that often the objective in working with survey data is to draw conclusions related to that particular finite population, and therefore, this comparative study makes sense in that context. For those readers who are interested in comparisons with the theoretical infinite population model, we suggest checking Scott and Wild (2002). Secondly, as mentioned above, some authors recommend including the design variables and the interactions between them as covariates in the model. However, in this case, and in particular in the case of the ESIE survey, this option would not be feasible due to the large number of parameters (a total of 515) to be estimated within the model. Therefore, we have decided to fit the mixed model, replicating in this way the comparison made by Lumley and Scott (2017) on real data sets. In addition, some of the covariates included in the models are related to the stratification variables. Finally, we would also like to point out that the estimation of the weighted mixed model (that takes into account the sampling weights when estimating the model coefficients) is ongoing research (see, e.g., Lumley and Huang (2023a) and Lumley and Huang (2023b)), and it would be interesting to analyze its behavior by means of a simulation study based on real survey data in the future.

To sum up, the weighted logistic regression performs properly in all the scenarios we have drawn. In contrast, the behavior of the unweighted logistic regression (both with and without random intercept) depends on the scenario. Therefore, based on the results of the simulation study, we believe that not using sampling weights when necessary leads to worse results than using them when they are not needed. For this reason, we would recommend the use of the weighted logistic regression model

in the context of complex survey data.

CHAPTER 5

Variable selection with LASSO regression

The paper related to the work presented in this chapter has been published:

Iparragirre, A., Lumley, T., Barrio, I., & Arostegui, I. (2023). Variable selection with LASSO regression for complex survey data. Stat, *12(1), e578.*

This chapter mostly replicates the above-mentioned article. However, some changes have been made to keep the notation and ensure cohesion with the rest of the document.
The code to reproduce the simulation study conducted in this study as well as the R package `wlasso` *derived from this work can be found on GitHub*[9] *and is presented in Chapter 8 of this document.*

[9] https://github.com/aiparragirre/wlasso

Summary

Variable selection is an important step to end up with good prediction models. LASSO regression models are one of the most commonly used methods for this purpose, for which cross-validation is the most widely applied validation technique to choose the tuning parameter (λ). Validation techniques in a complex survey framework are closely related to "replicate weights". However, to our knowledge, they have never been used in a LASSO regression context. Applying LASSO regression models to complex survey data could be challenging. The goal of this chapter is two-fold. On the one hand, we analyze the performance of replicate weights methods to select the tuning parameter for fitting LASSO regression models to complex survey data. On the other hand, we propose new replicate weights methods for the same purpose. In particular, we propose a new design-based cross-validation method as a combination of the traditional cross-validation and replicate weights. The performance of all these methods to select the tuning parameter for LASSO regression models has been analyzed and compared to the traditional cross-validation technique by means of an extensive simulation study. The results suggest a considerable improvement when the new proposal design-based cross-validation is used instead of the traditional cross-validation.

5.1 Introduction

In this chapter, we focus on variable selection for complex survey data in linear and logistic regression frameworks. In particular, throughout this chapter, we work under the two-stage stratified cluster sampling design, in which, as described in Chapter 2.1.2, the clusters are the Primary Sampling Units (PSU) or, in other words, the elements sampled in the first stage of the sampling process. However, the methods we propose in this chapter can easily be extended to one-stage stratified samples.

Least Absolute Shrinkage and Selection Operator (LASSO) regression (Tibshirani 1996), is nowadays a widely used technique for variable selection, especially when a large amount of predictor variables are available, in order to obtain more parsimonious, and hence, more interpretable prediction models. Very briefly, one goal of LASSO regression models, is to set some model coefficients to zero, reducing in this way the dimension of the model by selecting a subset of the available

predictor variables. The selection of this subset depends in turn on the election of a tuning parameter (λ) for which techniques such as bootstrap or cross-validation can be applied, the latter being the most widely used technique, in practice. These techniques, commonly known as validation methods, are used in order to select the tuning parameter with which the error of the final model, evaluated in a sample different from the one used to develop the model, is minimized (see, e.g., Hastie et al. (2009), James et al. (2013)). Shortly, those techniques consist in defining different training sets (in which models are fitted considering several tuning parameters) and test sets (in which the error of the models is estimated). The tuning parameter that minimizes the error of the training models in the test sets is selected to fit the LASSO model to the whole sample.

However, fitting LASSO regression models to complex survey data could be problematic for two reasons. In the first place, the debate mentioned in previous chapters and analyzed in particular in Chapter 4 about the need for considering sampling weights when fitting prediction models, could be extended to LASSO regression models. In addition, with the traditional above-mentioned validation methods (bootstrap or cross-validation, among others), training and test sets are randomly defined without considering the sampling design in the process. This may be a problem when working with complex survey data, given that PSUs could be split into training and test sets which may lead the training sets to underestimate the variability produced due to the sampling process and underestimate population error. This problem is usually known as "data leakage" (Kaufman et al. 2011). Both of these problems (weights-related as well as design-related) have recently been discussed in the literature. McConville et al. (2017) proposed incorporating sampling weights into the LASSO linear regression estimation process and Kshirsagar et al. (2017) extended this proposal to logistic regression models. Nonetheless, both of them applied the traditional K-fold cross-validation, which consists in randomly splitting sampled units into K subsamples (or folds) and defining K training sets, excluding a different fold (test set) each time. Nevertheless, if we apply this method to complex survey data, we may come across two types of problems. In the first place, sampling weights of the units in neither the training sets nor the test sets properly represent the entire finite population. Besides, and more importantly, as mentioned above, sampling design is not reflected in the way the folds are defined. Wieczorek et al. (2022) warned about this problem and proposed mimicking the structure of the sample obtained from the finite population in each fold. For example, for stratified sampling designs, Wieczorek et al. (2022) proposed making each fold a stratified

sample of PSUs from each stratum, i.e., creating simple random sample folds separately within each stratum (being all the elements from a given PSU placed in the same fold) and then combine them across strata. In this way, the weights of the units in the training and test sets represent the finite population properly and the variability of the data is also represented. However, as pointed out by the authors, it should be noted that in this way the number of folds could be limited by means of the sampled PSUs in each stratum (cannot be defined more folds than the maximum number of sampled PSUs per stratum). In other words, we need at least K PSUs per stratum for the proper application of this method. Furthermore, if we have a different (and non-proportional to K) number of PSUs in each stratum, the sampling weights of the training and test sets would also incorrectly represent the finite population.

In complex survey frameworks, other approaches, different from the abovementioned validation techniques, are usually used to define partially independent subsets of the sample. Those approaches are known as "replicate weights" methods. These methods consist of modifying the sampling weights to define new subsamples that replicate the original sample, in the way that these subsamples by means of these new weights (i.e., the "replicate weights") correctly represent the finite population. The most well-known replicate weights methods which are implemented in the `survey` R package (Lumley (2010), Lumley (2020)), are Jackknife Repeated Replication (JKn), Balanced Repeated Replication (BRR) and Rescaling Bootstrap (Bootstrap). Note that Jackknife term is usually used in variance estimation framework but this term is commonly denoted as leave-one-cluster-out (LOCO) (Merkle et al. 2019) or leave-one-group-out (LOGO) cross-validation (Kuhn and Johnson 2019) when the goal is validation. However, in order to be consistent with the terminology used in the `survey` R package (Lumley 2020) we denote this method as Jackknife throughout this chapter.

Therefore, the aim of the work presented in this chapter is two-fold. On the one hand, we aim to analyze the performance of the above-mentioned replicate weights methods, instead of traditional validation techniques, to select the tuning parameter for fitting LASSO regression models. On the other hand, our goal is to propose new methods to this end based on the idea of replicating weights. In particular, due to the popularity of cross-validation in this context, we propose a new design-based cross-validation method based on replicate weights, which will be more flexible than the one proposed by Wieczorek et al. (2022). In addition to the cross-validation, we also propose two new techniques (which we denote as split-sample repeated replication

(split) and extrapolation (extrap)) to select the tuning parameter for LASSO models. In this study we aim to analyze a) the impact of considering complex designs when using validation techniques for the selection of the tuning parameter, and b) the impact of the sampling weights when fitting LASSO models. Therefore, we compare by means of a simulation study the performance of different proposals based on replicate weights, to a) the traditional K-fold cross-validation that defines the folds by ignoring the sampling design but considers sampling weights for fitting LASSO models (weighted simple random sample cross-validation, w-SRSCV), and b) the unweighted simple random sample K-fold cross-validation ignoring weights for fitting LASSO models (unw-SRSCV).

The rest of the chapter is organized as follows. In Section 5.2 the basic notation on linear and logistic regression models and LASSO regression are given, existing replication methods applied in this work for the selection of the tuning parameter are defined and new methods based on the idea of replicating weights are also proposed. The performance of all the methods is analyzed by means of a simulation study, which is described in Section 5.3. Finally, we close the chapter with the main conclusions in Section 5.4.

5.2 Methods

This section is divided into three different parts. In Section 5.2.1, the basic notation previously described in Section 2.2 and implemented in this chapter is briefly reminded. In Section 5.2.2, the traditional LASSO regression is described and the modifications we propose to incorporate the sampling design into the process are set. Finally, Section 5.2.3 describes replicate weights methods considered in this study, both the ones previously defined in the literature together with the new methods proposed by the authors.

5.2.1 Basic Notation

Let us recall the basic notation that is necessary to properly follow the contents of this chapter. For more details on the models mentioned in this section, the readers can go back to Section 2.2.

As described in eq. (2.39), for a continuous response variable Y, the linear regression model is defined as follows:

$$y_i = \boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2), \tag{5.1}$$

and the vector of regression coefficients $\boldsymbol{\beta}$ is estimated ($\hat{\boldsymbol{\beta}}$) based on sample S by minimizing the residual sum of square (RSS($\boldsymbol{\beta}$)) as in eq. (5.2) (previously defined in eq. (2.40)),

$$\text{RSS}(\boldsymbol{\beta}) = \sum_{i \in S} (y_i - \sum_{j=0}^{p} \beta_j x_{ij})^2. \tag{5.2}$$

In a similar way, if Y is a dichotomous response variable, the logistic regression model is defined in eq. (5.3) (previously defined in eq. (2.51))

$$\text{logit}(P(Y = 1|\boldsymbol{x}_i)) = \text{logit}(p(\boldsymbol{x}_i)) = \ln\left[\frac{p(\boldsymbol{x}_i)}{1 - p(\boldsymbol{x}_i)}\right] = \boldsymbol{x}_i\boldsymbol{\beta}, \tag{5.3}$$

where $p(\boldsymbol{x}_i) = \dfrac{e^{\boldsymbol{x}_i\boldsymbol{\beta}}}{1 + e^{\boldsymbol{x}_i\boldsymbol{\beta}}}$ and $\hat{\boldsymbol{\beta}}$ is obtained by maximizing the log-likelihood function $\ell(\boldsymbol{\beta})$ defined in eq. (5.4) (previously defined in eq. (2.54)) (or equivalently, minimizing $-\ell(\boldsymbol{\beta})$):

$$\ell(\boldsymbol{\beta}) = \sum_{i \in S} \left[y_i \ln(p(\boldsymbol{x}_i)) + (1 - y_i) \ln(1 - p(\boldsymbol{x}_i)) \right]. \tag{5.4}$$

However, as explained in Section 2.2, when working with complex survey data, the weighted residual sum of square (WRSS($\boldsymbol{\beta}$)) and the pseudo-log-likelihood ($p\ell(\boldsymbol{\beta})$) functions can be considered instead of eqs. (5.2) and (5.4), respectively (note that these functions have previously been defined in eqs. (2.48) and (2.59)):

$$\text{WRSS}(\boldsymbol{\beta}) = \sum_{i \in S} w_i (y_i - \sum_{j=0}^{p} \beta_j x_{ij})^2, \tag{5.5}$$

and

$$p\ell(\boldsymbol{\beta}) = \sum_{i \in S} w_i \left[y_i \ln(p(\boldsymbol{x}_i)) + (1 - y_i) \ln(1 - p(\boldsymbol{x}_i)) \right]. \tag{5.6}$$

After estimating the regression coefficients, a value for the response variable can be estimated given the values of covariates $\boldsymbol{x}_i$ for unit $i \in U$ as $\hat{y}_i = \boldsymbol{x}_i\hat{\boldsymbol{\beta}}$ in linear regression framework and as $\hat{p}(\boldsymbol{x}_i) = \dfrac{e^{\boldsymbol{x}_i\hat{\boldsymbol{\beta}}}}{1 + e^{\boldsymbol{x}_i\hat{\boldsymbol{\beta}}}}$ in logistic regression. In order to ease the notation, let us denote as $\hat{f}(\cdot)$ the fitted (either linear or logistic) model and as $\hat{f}(\boldsymbol{x}_i)$ the corresponding estimated response for unit i, hereinafter, i.e.,

$$\hat{f}(\boldsymbol{x}_i) = \begin{cases} \hat{y}_i, & \text{in linear regression framework,} \\ \hat{p}(\boldsymbol{x}_i), & \text{in logistic regression framework.} \end{cases} \tag{5.7}$$

5.2.2 LASSO regression for variable selection

When a large amount of predictor variables are available, the LASSO regression model is commonly used for variable selection. Briefly, this method forces some regression coefficients to zero, and thus more interpretable models are obtained. This variable selection method is briefly described below. For more information and details on this topic related to the performance and geometrical interpretation of LASSO regression models we recommend Hastie et al. (2009), James et al. (2013), Sanchez and Marzban (2020) and Tibshirani (1996).

For a given value of the tuning parameter λ, linear and logistic LASSO regression models are fitted by minimizing the following functions, respectively:

$$\min\left\{\text{RSS}(\boldsymbol{\beta})+\lambda\sum_{j=1}^{p}|\beta_j|\right\} \quad \text{and} \quad \min\left\{-\ell(\boldsymbol{\beta})+\lambda\sum_{j=1}^{p}|\beta_j|\right\}. \tag{5.8}$$

In practice, $K-$fold cross-validation is usually applied to select the optimum value for λ in order to minimize the error of the fitted model. Sampled units are randomly split into K subsamples of the same size. In each step, $\forall t = 1,\ldots,K$ the t^{th} subsample is set as the test set ($S_{\text{test}(t)}$), while the rest $K-1$ subsets form the training set ($S_{\text{tr}(t)}$). Then, a grid for λ values is defined (λ_l, $\forall l = 1,\ldots,L$), and for each of these values a model is fitted to each training set $S_{\text{tr}(t)}$, $\forall t = 1,\ldots,K$ following eq. (5.8) (let us denote this model (either linear or logistic) as $\hat{f}^l_{\text{tr}(t)}(\cdot)$) and applied to the test set (let $\hat{f}^l_{\text{tr}(t)}(\boldsymbol{x}_i)$ indicate the predicted value, $\forall i \in S_{\text{test}(t)}$). The estimation error for each unit is calculated by means of the loss function defined in eq. (5.9) depending on the framework, linear or logistic regression, as follows. $\forall i \in S_{\text{test}(t)}$,

$$\mathcal{L}(y_i,\hat{f}^l_{\text{tr}(t)}(\boldsymbol{x}_i)) = \begin{cases} (y_i-\hat{f}^l_{\text{tr}(t)}(\boldsymbol{x}_i))^2, & \text{linear,} \\ -y_i\ln\left(\hat{f}^l_{\text{tr}(t)}(\boldsymbol{x}_i)\right)-(1-y_i)\ln\left(1-\hat{f}^l_{\text{tr}(t)}(\boldsymbol{x}_i)\right), & \text{logistic.} \end{cases} \tag{5.9}$$

$\forall t = 1,\ldots,K$, the error in subset t of the model fitted considering λ_l, $\forall l = 1,\ldots,L$ in eq. (5.8) is estimated as follows:

$$\widehat{Err}^{\,l}_{(t)} = \frac{1}{n_{\text{test}(t)}}\sum_{i\in S_{\text{test}(t)}}\mathcal{L}(y_i,\hat{f}^l_{\text{tr}(t)}(\boldsymbol{x}_i)), \tag{5.10}$$

being $n_{\text{test}(t)}$ the size of $S_{\text{test}(t)}$, $\forall t = 1,\ldots,K$. This process is repeated K times,

by setting a different subset t as the test set each time. The cross-validated error corresponding to the tuning parameter λ_l, $\forall l = 1, \ldots, L$ is finally estimated as follows:

$$\widehat{Err}_{CV}(\lambda_l) = \frac{1}{K} \sum_{t=1}^{K} \widehat{Err}^{l}_{(t)}. \tag{5.11}$$

Among all the values considered as tuning parameters, the one that minimizes the cross-validated error is selected as the "optimal" penalty parameter,

$$\Lambda = \underset{\lambda_l \,:\, l \in \{1,\ldots,L\}}{\operatorname{argmin}} \{\widehat{Err}_{CV}(\lambda_l)\}, \tag{5.12}$$

and the model is fitted to the whole sample S, including Λ as the tuning parameter in eq. (5.8), i.e.,

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \text{RSS}(\boldsymbol{\beta}) + \Lambda \sum_{j=1}^{p} |\beta_j| \right\} \quad \text{and} \quad \hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ -\ell(\boldsymbol{\beta}) + \Lambda \sum_{j=1}^{p} |\beta_j| \right\}. \tag{5.13}$$

However, in the whole process explained above, sampling design and sampling weights are not considered (let us denote the method described above as the unweighted simple random sample cross-validation (unw-SRSCV), hereinafter). We believe that when working with complex survey data, sampling design should be considered in the whole process: 1) when fitting the model, 2) when defining training and test sets, and 3) when estimating the error. Below, we explain how we propose to address these three points as a whole.

In the first place, when fitting the LASSO regression models, sampling weights should be considered as in eq. (5.14) instead of eq. (5.8) for linear and logistic regression models, respectively (Kshirsagar et al. 2017, McConville et al. 2017):

$$\min \left\{ \text{WRSS}(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \quad \text{and} \quad \min \left\{ -p\ell(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} |\beta_j| \right\}. \tag{5.14}$$

Secondly, in Section 5.2.3 we describe different methods based on replicate weights that could be considered to take into account the sampling design when defining training and test sets. Finally, sampling weights should also be considered when estimating the error. In particular, if we focus on the above-mentioned cross-validation method, we could rewrite eq. (5.10) as follows in order to consider the weights when

estimating the error in subset t:

$$\widehat{Err}^{l}_{(t)} = \frac{1}{\sum_{i \in S_{\text{test}(t)}} w_i} \sum_{i \in S_{\text{test}(t)}} w_i \mathcal{L}(y_i, \hat{f}^{l}_{\text{tr}(t)}(\boldsymbol{x}_i)). \tag{5.15}$$

We denote as weighted simple random sample cross-validation (w-SRSCV) the method that considers eq. (5.14) to fit the model and eq. (5.15) to estimate the error, but defines the folds by randomly splitting sampled units into different subsets ignoring the sampling design as described previously.

5.2.3 Selecting LASSO model's tuning parameter with complex survey data

In this work, we propose to use replicate weights for the selection of the tuning parameter λ. In the following lines, we describe the six replicate weights methods we considered in this work to define training and test sets when selecting the tuning parameter for LASSO models (three of which are existing methods and the other three are new proposals of the authors). The goal of replicate weights methods is to modify the sampling weights to define new partially independent subsamples that replicate the original sample, in the way that the finite population is properly represented in each subsample by means of the modified weights known as "replicate weights". Some of the replicate weights methods that are described below are commonly applied for other purposes, such as variance estimation, when working with complex survey data. These methods are known as Jackknife Repeated Replication (JKn), Rescaling Bootstrap (Bootstrap), and Balanced Repeated Replication (BRR) (see, e.g., Heeringa et al. (2017) and Wolter (2007) for more information about these methods). However, as far as we know, they have never been used for selecting the tuning parameter for LASSO regression models. In this work, we propose to incorporate the abovementioned replicate weights methods in this context. In addition, we also propose three new methods based on the idea of replicating weights, which we denote as the design-based K-fold cross-validation (dCV), Split-sample Repeated Replication (split) and extrapolation (extrap) for the same purpose. Figures from 5.1 to 5.6 depict a graphical summary of all these methods (note that the figures are not self-explanatory and should be read in combination with the descriptions below for a correct understanding of each method).

As mentioned at the beginning of the chapter, in this study, a two-stage stratified cluster sampling has been considered, thus PSUs are the clusters sampled from

each stratum. Nevertheless, it should also be noted that all the methods described below can be extended to one-stage stratified samples in which different numbers of individuals (which would be the PSUs in that case) rather than clusters are sampled from each stratum.

For a better understanding of the methods' performance, let us recall the basis of two-stage stratified cluster sampling designs and the notation previously described in Section 2.1.2. As indicated in eqs. (2.14) and (2.15), the finite population U is partitioned into H strata, which at the same time are partitioned into A_h clusters, i.e.,

$$U = \bigcup_{h=1}^{H} \bigcup_{\alpha=1}^{A_h} U_{h,\alpha}. \tag{5.16}$$

The sampling process is carried out in two stages. In the first stage, from each stratum $h \in \{1, \ldots, H\}$ a previously specified number of clusters (which we denote as a_h) is randomly selected to be part of the sample. Recall that $\mathbb{A}_h$ in eq. (5.17) (previously defined in eq. (2.24)) is the set of cluster indexes from stratum h, that are selected in the first stage.

$$\mathbb{A}_h = \{\alpha \in \{1, \ldots, A_h\} \ : 1_h(\alpha) = 1\}, \tag{5.17}$$

where the indicator function $1_h(\alpha)$ in eq. (5.18) (previously defined in eq. (2.16)) indicates whether the cluster $U_{h,\alpha}$ has been selected in the first stage:

$$1_h(\alpha) = \begin{cases} 1 & \text{if the cluster } U_{h,\alpha} \text{ is selected in the first stage,} \\ 0 & \text{otherwise.} \end{cases} \tag{5.18}$$

Then, $\forall \dot{\alpha} \in \mathbb{A}_h$, the set of individuals in the cluster $\dot{\alpha}$ from stratum h, $U_{h,\dot{\alpha}}$, has been selected in the first stage. In the second stage of the sampling process, $\forall h \in \{1, \ldots, H\}$ and $\forall \dot{\alpha} \in \mathbb{A}_h$, out of the $N_{h,\dot{\alpha}}$ units in $U_{h,\dot{\alpha}}$ only a given number of units $n_{h,\dot{\alpha}}$ are finally sampled and end up in the sample subset $S_{h,\dot{\alpha}} \subset U_{h,\dot{\alpha}}$, the union of which form finally the sample S as defined in eq. (5.19) (previously defined in eq. (2.26)),

$$S = \bigcup_{h=1}^{H} \bigcup_{\dot{\alpha} \in \mathbb{A}_h} S_{h,\dot{\alpha}}. \tag{5.19}$$

We now proceed to describe, one by one, each of the replicate weights methods considered. We start by defining the three methods that are already proposed in the literature (JKn, Bootstrap and BRR), and, afterward, we describe in detail the

new methods proposed by the authors (dCV, split, and extrapolation). A different number of training and test sets are generated for each method that has been considered. In order to unify the notation for all the methods, let T_m indicate the total number of training and test sets for each method m considered. The training and test sets are indicated as $S^m_{\text{tr}(t)}$ and $S^m_{\text{test}(t)}$ for the method m, $\forall t = 1, \ldots, T_m$.

Jackknife Repeated Replication (JKn)

Based on this method, the number of training and test sets are exactly the total number of sampled PSUs (i.e., $T_{\text{JKn}} = \sum_{h=1}^{H} a_h = a$). A different PSU is set as the test set each time, while the rest form the training set. That is, $\forall h \in \{1, \ldots, H\}$ and $\forall \dot{\alpha} \in \mathbb{A}_h$, $\exists! \, t \in \{1, \ldots, T_{\text{JKn}}\} : S^{\text{JKn}}_{\text{test}(t)} = S_{h,\dot{\alpha}}$, being the corresponding training set formed by the rest $a - 1$ sampled PSUs excluding $S_{h,\dot{\alpha}}$ (i.e., $S^{\text{JKn}}_{\text{tr}(t)} = S - S_{h,\dot{\alpha}}$). See Figure 5.1 for a graphical explanation.

Replicate weights for the t^{th} training set are defined as follows. The sum of all the sampling weights of the individuals in the test set $S^{\text{JKn}}_{\text{test}(t)} = S_{h,\dot{\alpha}}$ is assumed by the individuals of the same stratum S_h that are part of the training set. The replicate weight of the rest of the units that end up in the training set but are not in stratum S_h, is the same as their original sampling weight. This is mathematically defined in eq. (5.20):

$$w^{*,\text{tr}(t)}_{i,\text{JKn}} = \begin{cases} 0, & \text{if } i \in S^{\text{JKn}}_{\text{test}(t)} = S_{h,\dot{\alpha}}, \\ w_i, & \text{if } i \in S^{\text{JKn}}_{\text{tr}(t)} \text{ and } i \notin S_h, \\ w_i \cdot \dfrac{a_h}{a_h - 1}, & \text{if } i \in S^{\text{JKn}}_{\text{tr}(t)} \text{ and } i \in S_h, \end{cases} \quad \forall i \in S. \tag{5.20}$$

Even though each test set is formed by the PSU $S_{h,\dot{\alpha}}$ excluded from the corresponding training set, the error is usually estimated considering the whole sample. Hence, the replicate weights of the units in the test set, are assumed to be equal to the original sampling weights, as defined in eq. (5.21):

$$w^{*,\text{test}(t)}_{i,\text{JKn}} = \begin{cases} 0, & \text{if } i \in S^{\text{JKn}}_{\text{tr}(t)}, \\ w_i, & \text{if } i \in S^{\text{JKn}}_{\text{test}(t)}. \end{cases} \quad \forall i \in S. \tag{5.21}$$

Rescaling Bootstrap (Bootstrap)

$T_{\text{Bootstrap}} = B$ bootstrap resamples are generated as proposed by Rao and Wu (1988). $\forall h \in \{1, \ldots, H\}$ $a_h - 1$ PSUs are selected with replacement, which form

the t^{th} training set, $\forall t = 1, \ldots, T_{\text{Bootstrap}}$ (see Figure 5.2). For each stratum $h \in \{1, \ldots, H\}$ and each sampled cluster from h, $\dot{\alpha} \in \mathbb{A}_h$ let us denote as $v_{h,\dot{\alpha}}^{(t)}$, the number of times that the PSU $S_{h,\dot{\alpha}}$ is selected to be part of the bootstrap resample t. Note that if the PSU $S_{h,\dot{\alpha}}$ is not selected to form the resample t, then $v_{h,\dot{\alpha}}^{(t)} = 0$. Then, $\forall t = 1, \ldots, T_{\text{Bootstrap}}$ the replicate weights for the t^{th} training set are defined as follows. $\forall i \in S$, $\exists!\, h \in \{1, \ldots, H\}$ and $\exists!\, \dot{\alpha} \in \mathbb{A}_h \,:\, i \in S_{h,\dot{\alpha}}$. Then,

$$w_{i,\text{Bootstrap}}^{*,\text{tr}(t)} = \sum_{h=1}^{H} \sum_{\dot{\alpha} \in \mathbb{A}_h} 1_{S_{h,\dot{\alpha}}}(i) \cdot w_i \cdot \frac{a_h}{a_h - 1} \cdot v_{h,\dot{\alpha}}^{(t)}, \quad \forall i \in S, \tag{5.22}$$

where $1_{S_{h,\dot{\alpha}}}(i)$ is the indicator function taking the value 1 in case $i \in S_{h,\dot{\alpha}}$ and 0 otherwise (see eq. (2.19)).

The test set corresponding to each training set is the original sample $S = S_{\text{test}(t)}^{\text{Bootstrap}}$, $\forall t = 1, \ldots, T_{\text{Bootstrap}}$. Hence, for this method, the replicate weights of the test set are the original sampling weights as defined in eq. (5.23), $\forall t = 1, \ldots, T_{\text{Bootstrap}}$:

$$w_{i,\text{Bootstrap}}^{*,\text{test}(t)} = w_i, \quad \forall i \in S_{\text{test}(t)}^{\text{Bootstrap}} = S. \tag{5.23}$$

Balanced Repeated Replication (BRR)

This method was originally designed to be applied in samples with 2 PSUs per stratum. $\forall h = 1, \ldots, H$ one of the PSUs from the stratum h is set to the training set while the other is set to the test set (depicted in Figure 5.3). There are 2^H different possible training and test sets to define in this way, which may usually be computationally unfeasible. Instead, T_{BRR} (where $T_{\text{BRR}} \leq H + 4$) different sets are usually defined by selecting the PSU splits in a particular way by means of the Hadamard matrix as proposed by McCarthy (1966). Nowadays, this method is extended to be also applied when an even number of PSUs per stratum are available (Lumley 2020, Wolter 2007). $\forall i \in S$, the replicate weights for the t^{th} training set are defined as follows, $\forall t = 1, \ldots, T_{\text{BRR}}$:

$$w_{i,\text{BRR}}^{*,\text{tr}(t)} = \begin{cases} 0, & \text{if } i \in S_{\text{test}(t)}^{\text{BRR}}, \\ 2w_i, & \text{if } i \in S_{\text{tr}(t)}^{\text{BRR}}, \end{cases} \quad \forall i \in S. \tag{5.24}$$

Replicate weights for the corresponding test sets ($w_{i,\text{BRR}}^{*,\text{test}(t)}$) are defined by exchanging the roles of test and training sets in eq. (5.24), as shown in eq. (5.25):

$$w_{i,\text{BRR}}^{*,\text{test}(t)} = \begin{cases} 0, & \text{if } i \in S_{\text{tr}(t)}^{\text{BRR}}, \\ 2w_i, & \text{if } i \in S_{\text{test}(t)}^{\text{BRR}}, \end{cases} \quad \forall i \in S. \tag{5.25}$$

Design-based K-fold cross-validation (dCV)

This method is summarized in Figure 5.4. The $a = \sum_{h=1}^{H} a_h$ sampled PSUs are randomly split into K subsets and $T_{\text{dCV}} = K$ training and test sets are defined. For $t = 1, \ldots, T_{\text{dCV}}$ the t^{th} subset is set as test set ($S_{\text{test}(t)}^{\text{dCV}}$), being the corresponding training set ($S_{\text{tr}(t)}^{\text{dCV}}$) formed by the rest $K-1$ subsets (excluding the t^{th} one). $\forall t = 1, \ldots, T_{\text{dCV}}$ let $a_{h,\text{test}(t)}^{*}$ indicate the number of PSUs from S_h that has ended up in $S_{\text{test}(t)}^{\text{dCV}}$, while $a_{h,\text{tr}(t)}^{*} = a_h - a_{h,\text{test}(t)}^{*}$ indicates the number of PSUs from S_h in $S_{\text{tr}(t)}^{\text{dCV}}$. For each sampled unit $i \in S$, we propose to define replicate weights as follows for the t^{th} training set:

$$w_{i,\text{dCV}}^{*,\text{tr}(t)} = \begin{cases} 0, & \text{if } i \in S_{\text{test}(t)}^{\text{dCV}}, \\ \sum_{h=1}^{H} 1_{S_h}(i) \cdot w_i \cdot \dfrac{a_h}{a_h - a_{h,\text{test}(t)}^{*}}, & \text{if } i \in S_{\text{tr}(t)}^{\text{dCV}}, \end{cases} \quad \forall i \in S. \tag{5.26}$$

That is, the sum of all the sampling weights of the units in the test set is assumed by the units from the same stratum in the training set. In the same way, replicate weights for the t^{th} test sets (which will be denoted as $w_{i,\text{dCV}}^{*,\text{test}(t)}$ hereinafter) can be defined in the same way exchanging the roles of the training and test sets with each other in eq. (5.26), that is,

$$w_{i,\text{dCV}}^{*,\text{test}(t)} = \begin{cases} 0, & \text{if } i \in S_{\text{tr}(t)}^{\text{dCV}}, \\ \sum_{h=1}^{H} 1_{S_h}(i) \cdot w_i \cdot \dfrac{a_h}{a_h - a_{h,\text{tr}(t)}^{*}}, & \text{if } i \in S_{\text{test}(t)}^{\text{dCV}}, \end{cases} \quad \forall i \in S. \tag{5.27}$$

Note that our goal is to get at least one PSU from each stratum in every training set (not in every fold) in order to have all the strata represented in the training models. In contrast, we are not concerned about not having any PSU from a particular stratum in a particular test set. In other words,

$$1 \leq a_{h,\text{tr}(t)}^{*} \leq a_h, \quad \forall t = 1, \ldots, T_{\text{dCV}}, \tag{5.28}$$

and note that if $a_{h,\text{tr}(t)}^{*} = a_h$, then necessarily, $a_{h,\text{test}(t)}^{*} = 0$. The condition set in

eq. (5.28) will be satisfied as long as no stratum ends up with all its PSUs in the same fold t, $\forall t = 1, \ldots, T_{\text{dCV}}$ (given that in that case, when the t^{th} fold is left as the test set $S^{\text{dCV}}_{\text{test}(t)}$, $a^*_{h,\text{tr}(t)} = 0$). After PSUs are randomly assigned to folds, the dCV method checks whether the condition in eq. (5.28) is satisfied or not. In case any stratum has all its PSUs in the same fold, then this method reassigns folds until the condition in eq. (5.28) is satisfied.

Therefore, at least two PSUs per stratum are needed for the correct application of this method (i.e., $\forall h = 1, \ldots, H,\ a_h \geq 2$), each of them classified in a different fold. This is an advantage over the method proposed by Wieczorek et al. (2022), which requires at least K PSUs per stratum ($\forall h = 1, \ldots, H,\ a_h \geq K$).

Split-sample Repeated Replication (split)

A given percentage of PSUs is randomly set into the training set and the rest into the test set (see Figure 5.5). This process could be repeated T_{split} times with a different split each time, defining in this way T_{split} training and test sets. Replicate weights for units in either training or test sets, can be defined in two different ways, $\forall t = 1, \ldots, T_{\text{split}}$:

- *split-cv:* As previously described in eq. (5.26) for the dCV, in this method, the sum of the weights of the units that end up in the test set is assumed by the units that end up in the training set in the same stratum. Thus, replicate weights of the t^{th} training set are defined as in eq. (5.29):

$$w^{*,\text{tr}(t)}_{i,\text{split-cv}} = \begin{cases} 0, & \text{if } i \in S^{\text{split-cv}}_{\text{test}(t)}, \\ \sum_{h=1}^{H} 1_{S_h}(i) \cdot w_i \cdot \dfrac{a_h}{a_h - a^*_{h,\text{test}(t)}}, & \text{if } i \in S^{\text{split-cv}}_{\text{tr}(t)}, \end{cases} \quad \forall i \in S, \tag{5.29}$$

  where $a^*_{h,\text{test}(t)}$ indicates the number of clusters from stratum h in $S^{\text{split-cv}}_{\text{test}(t)}$. Similarly, as described in eq. (5.27) for the dCV method, the replicate weights of the t^{th} test set are defined as in eq. (5.30), in which the sum of the sampling weights of the units in the training set are assumed by the units in the test set in the same stratum:

$$w^{*,\text{test}(t)}_{i,\text{split-cv}} = \begin{cases} 0, & \text{if } i \in S^{\text{split-cv}}_{\text{tr}(t)}, \\ \sum_{h=1}^{H} 1_{S_h}(i) \cdot w_i \cdot \dfrac{a_h}{a_h - a^*_{h,\text{tr}(t)}}, & \text{if } i \in S^{\text{split-cv}}_{\text{test}(t)}, \end{cases} \quad \forall i \in S, \tag{5.30}$$

  where $a^*_{h,\text{tr}(t)}$ indicates the number of clusters from stratum h in $S^{\text{split-cv}}_{\text{tr}(t)}$.

- *split-boot:* This method consists of replicating by replacement the PSUs of both the training and test sets until having $a_h - 1$ in each stratum and calculating the weights as in eq. (5.22) for the Bootstrap, that is,

$$w^{*,\text{tr}(t)}_{i,\text{split-boot}} = \sum_{h=1}^{H} \sum_{\dot{\alpha} \in \mathbb{A}_h} 1_{S_{h,\dot{\alpha}}}(i) \cdot w_i \cdot \frac{a_h}{a_h - 1} \cdot v^{(t)}_{h,\dot{\alpha}}, \quad \forall i \in S, \tag{5.31}$$

$$w^{*,\text{test}(t)}_{i,\text{split-boot}} = \sum_{h=1}^{H} \sum_{\dot{\alpha} \in \mathbb{A}_h} 1_{S_{h,\dot{\alpha}}}(i) \cdot w_i \cdot \frac{a_h}{a_h - 1} \cdot \tilde{v}^{(t)}_{h,\dot{\alpha}}, \quad \forall i \in S, \tag{5.32}$$

where $v^{(t)}_{h,\dot{\alpha}}$ indicates the number of times that the PSU $S_{h,\dot{\alpha}}$ is selected to be part of the t^{th} training set $S^{\text{split-boot}}_{\text{train}(t)}$, $\forall \dot{\alpha} \in \mathbb{A}_h$, $\forall h \in \{1, \ldots, H\}$ (note that if $S_{h,\dot{\alpha}}$ is set to the test set, then, $v^{(t)}_{h,\dot{\alpha}} = 0$). Similarly, $\tilde{v}^{(t)}_{h,\dot{\alpha}}$ indicates the number of times that the PSU $S_{h,\dot{\alpha}}$ is selected to be part of the t^{th} test set $S^{\text{split-boot}}_{\text{test}(t)}$ (if $S_{h,\dot{\alpha}}$ is set to the training set, then, $\tilde{v}^{(t)}_{h,\dot{\alpha}} = 0$).

In the same way as in the dCV method, and due to the same reasons, we force the algorithm to have at least one PSU from each stratum in every training set, in both split-cv and split-boot methods.

Extrapolation (extrap)

A given percentage of strata are set as training set and the rest as test set (Figure 5.6). The process is repeated T_{extrap} times with a different split each time, defining T_{extrap} different training and test sets. In this case, replicate weights are equal to sampling weights for units in the training set and 0 for units in the test set when fitting the models, $\forall i \in S$. $\forall t = 1, \ldots, T_{\text{extrap}}$:

$$w^{*,\text{tr}(t)}_{i,\text{extrap}} = \begin{cases} 0, & \text{if } i \in S^{\text{extrap}}_{\text{test}(t)}, \\ w_i, & \text{if } i \in S^{\text{extrap}}_{\text{tr}(t)}, \end{cases} \quad \forall i \in S. \tag{5.33}$$

Replicate weights $w^{*,\text{test}(t)}_{i,\text{extrap}}$ are described in the same way, exchanging the roles of training and test sets in eq. (5.33) as described in eq. (5.34):

$$w^{*,\text{test}(t)}_{i,\text{extrap}} = \begin{cases} 0, & \text{if } i \in S^{\text{extrap}}_{\text{tr}(t)}, \\ w_i, & \text{if } i \in S^{\text{extrap}}_{\text{test}(t)}, \end{cases} \quad \forall i \in S. \tag{5.34}$$

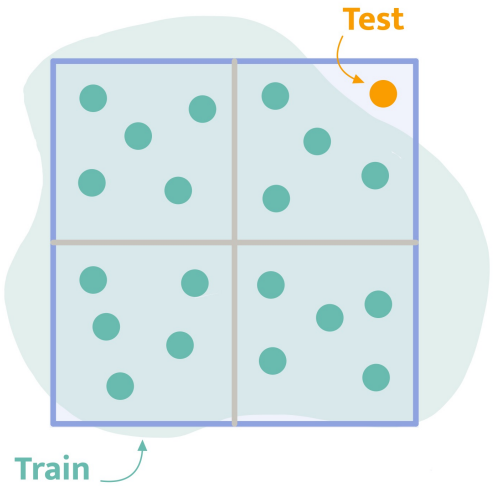


Figure 5.1: Graphical summary of Jackknife Repeated Replication (JKn). Note that each dot represents a PSU and gray lines define the strata.

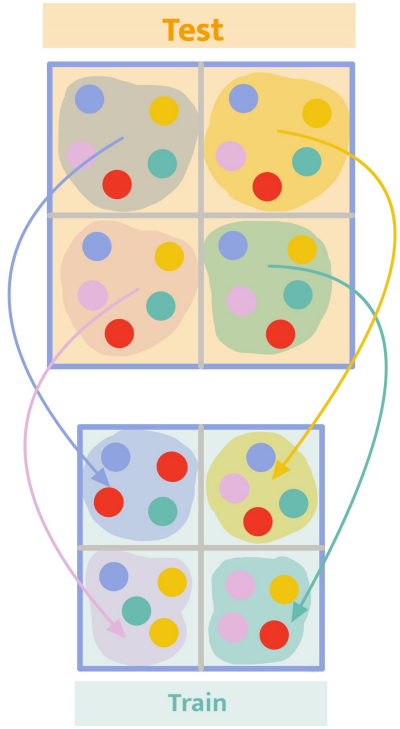


Figure 5.2: Graphical summary of Rescaling Bootstrap (Bootstrap). Note that each dot represents a PSU and gray lines define the strata.

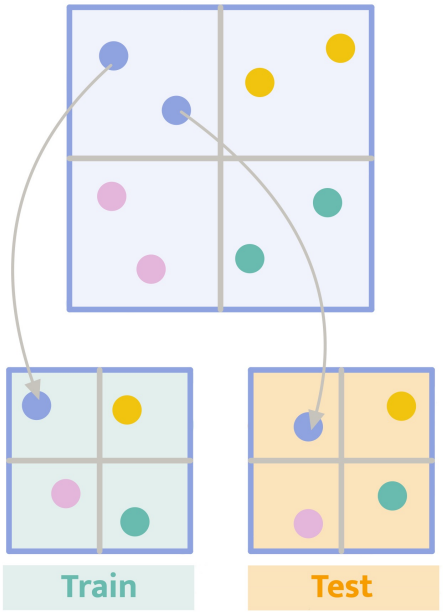


Figure 5.3: Graphical summary of Balanced Repeated Replication (BRR). Note that each dot represents a PSU and gray lines define the strata.

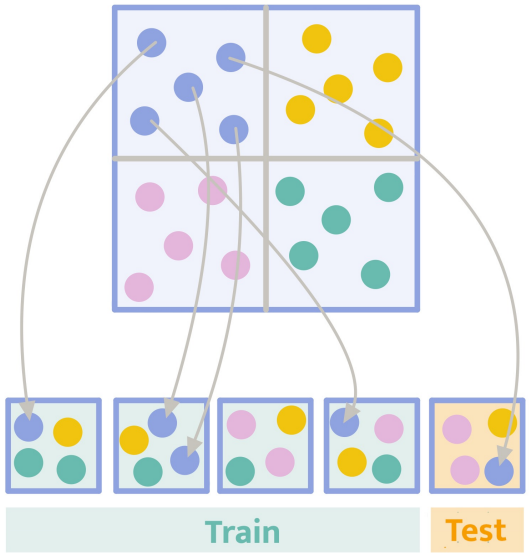


Figure 5.4: Graphical summary of Design-based K-fold cross-validation (dCV). Note that each dot represents a PSU and gray lines define the strata.

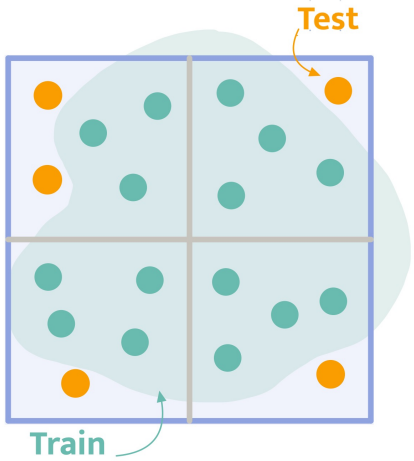


Figure 5.5: Graphical summary of Split-sample Repeated Replication (split). Note that each dot represents a PSU and gray lines define the strata.

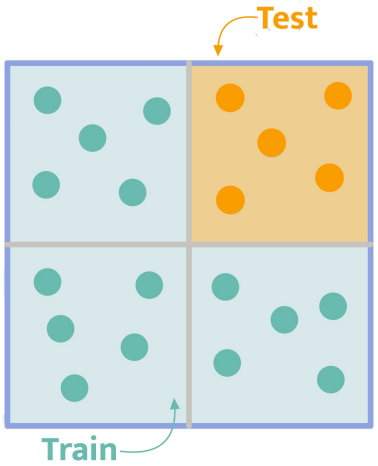


Figure 5.6: Graphical summary of Extrapolation (extra). Note that each dot represents a PSU and gray lines define the strata.

5.3 Simulation Study

This section describes the simulation study conducted in order to analyze the performance of different methods when selecting the tuning parameter for fitting LASSO regression models. Our goal is to compare the performance of the replication methods proposed in Section 5.2.3 (i.e., JKn, Bootstrap, BRR, dCV, split-cv, split-boot and extrap), to the methods described in Section 5.2.2 (unw-SRSCV and w-SRSCV). The goal is to compare the differences between the tuning parameters selected with different methods, the number of covariates that would be selected if the models were fitted considering that tuning parameter, and the error we would obtain with that model. We compare those results with the "true" results we would obtain if the finite population were known in practice.

The rest of the section is organized as follows: Section 5.3.1 describes the process of data simulation and scenarios, Section 5.3.2 describes the simulation set-up, and Section 5.3.3 depicts and summarizes the main results.

5.3.1 Data generation and sampling design

In the following lines data simulation process is described. Let us define as $N = 100\,000$ the finite population size and as $p = 75$ the number of variables denoted as $X_1, \ldots, X_{50}, Z_1, \ldots, Z_{25}$. In this simulation study, we consider the variables $Z_1, \ldots, Z_{25}$ to be latent variables, that are used to define the response variable, but are not available in the samples to fit the models. In this way, we aim to define more realistic scenarios, in which the perfect models cannot be fitted. Instead, $Z_1, \ldots, Z_{25}$ are used to define the sampling design.

For a given value of p^*, where $p^* \leq p$, let $\boldsymbol{\mu}_{p^*}$ indicate the null vector of dimension $1 \times p^*$ and $\Sigma_{p^* \times p^*}$ a matrix of dimension $p^* \times p^*$ of values of $\eta = 0.15$ off-diagonal and values of 1 on the diagonal, i.e.,

$$\boldsymbol{\mu}_{(p^*)} = (0, \ldots, 0)^T \quad \text{and} \quad \Sigma_{p^* \times p^*} = (1 - \eta) \cdot I_{p^* \times p^*} + \eta \cdot J_{p^* \times p^*}, \tag{5.35}$$

being $I_{p^* \times p^*}$ the identity matrix and $J_{p^* \times p^*}$ the matrix of 1s. In addition, the vector of regression coefficients $\boldsymbol{\beta} = (\beta_0, \boldsymbol{\beta^X}, \boldsymbol{\beta^Z})^T$ is defined as follows:

$$\boldsymbol{\beta^X} = (\underbrace{-2, \ldots, -2}_{(11)}, \underbrace{0, \ldots, 0}_{(9)}, \underbrace{-2, \ldots, -2}_{(10)}, \underbrace{0, \ldots, 0}_{(9)}, \underbrace{-2, \ldots, -2}_{(11)})^T, \quad \boldsymbol{\beta^Z} = (\underbrace{2, \ldots, 2}_{(25)})^T, \tag{5.36}$$

and the value of β_0 changes depending on the nature of the response variable (continuous or dichotomous), and hence, will be defined later on.

We describe below the steps followed to generate the finite populations of this simulation study. Two scenarios are defined generating two different populations. In Scenario 1 (S1), the $p = 75$ covariates are unit-level variables (i.e., there are $d = 0$ cluster-level variables), while in Scenario 2 (S2), $d = 5$ variables are defined as cluster-level variables, while the rest of $p - d = 70$ variables are unit-level. In each population, two response variables have been generated: (a) a continuous response variable (linear regression), and (b) a dichotomous response variable (logistic regression).

1. For $d = 0$ (S1) and $d = 5$ (S2), two finite populations are generated by making N realizations of:

$$(X_{d+1}, \ldots, X_{50}, Z_1, \ldots, Z_{25}) \sim N(\boldsymbol{\mu}_{(p-d)}, \Sigma_{(p-d)\times(p-d)}). \tag{5.37}$$

2. Let us denote as $\{\boldsymbol{z}_i = (z_{i,1}, \ldots, z_{i,25})\}_{i=1}^{N}$ the set of N realizations of $Z_1, \ldots, Z_{25}$. Data is sort based on $\boldsymbol{z}_i\boldsymbol{\beta}^{\boldsymbol{Z}}$, $\forall i = 1, \ldots, N$. Strata are defined by partitioning the population data set on sets of the same size ($H = 5$) and clusters by partitioning each stratum on sets of the same size ($A_h = 20, \forall h = 1, \ldots, H$, being A_h the number of clusters generated in stratum h in the population). In this way, a total of 100 clusters of size $N_{h,\alpha} = 1000$ are generated, $\forall h = 1, \ldots, H$ and $\forall \alpha = 1, \ldots A_h$.

3. If $d \neq 0$, generate d cluster-level variables by making $A = \sum_{h=1}^{H} A_h$ realizations of:

$$(X_1, \ldots, X_d) \sim N(\boldsymbol{\mu}_{(d)}, \Sigma_{(d)\times(d)}). \tag{5.38}$$

   Note that for two different units in the same cluster, their corresponding cluster-level covariates should take the same values, i.e., $\forall i, j$ in the same cluster, $(x_{i,1}, \ldots, x_{i,d}) = (x_{j,1}, \ldots, x_{j,d})$. Therefore, we repeat each realization $N_{h,\alpha}$ times. We now have defined the values corresponding to $X_1, \ldots, X_{50}$ variables for all the units in the finite population: $\{\boldsymbol{x}_i = (x_{i,1}, \ldots, x_{i,50})\}_{i=1}^{N}$.

4. Generate the values for the response variables as follows:

   (a) Linear regression framework: The values for the continuous response

variable are randomly generated as follows,

$$y_i = \boldsymbol{x}_i\boldsymbol{\beta}^{\boldsymbol{X}} + \boldsymbol{z}_i\boldsymbol{\beta}^{\boldsymbol{Z}} + \epsilon_i, \tag{5.39}$$

where ϵ_i is a realization of $\epsilon_i \sim N(0, 10^2)$, $\forall i = 1, \ldots, N$ ($\beta_0 = 0$ in this case).

(b) Logistic regression framework: First, we generate the probabilities of event $\forall i = 1, \ldots, N$ in the following way,

$$\text{logit}(p(\boldsymbol{x}_i, \boldsymbol{z}_i)) = \beta_0 + \boldsymbol{x}_i\boldsymbol{\beta}^{\boldsymbol{X}} + \boldsymbol{z}_i\boldsymbol{\beta}^{\boldsymbol{Z}} \Longrightarrow p(\boldsymbol{x}_i, \boldsymbol{z}_i) = \frac{e^{\beta_0 + \boldsymbol{x}_i\boldsymbol{\beta}^{\boldsymbol{X}} + \boldsymbol{z}_i\boldsymbol{\beta}^{\boldsymbol{Z}}}}{1 + e^{\beta_0 + \boldsymbol{x}_i\boldsymbol{\beta}^{\boldsymbol{X}} + \boldsymbol{z}_i\boldsymbol{\beta}^{\boldsymbol{Z}}}}, \tag{5.40}$$

where we set $\beta_0 = -10$. Finally, the value for the response variable y_i is randomly generated by following Bernoulli's distribution, i.e.,

$$y_i \sim Bernoulli(p(\boldsymbol{x}_i, \boldsymbol{z}_i)), \ \forall i = 1, \ldots, N. \tag{5.41}$$

In this way, given the value we set for β_0, the probability of event in the finite population is around 25%.

Then, the finite population U is defined as the set of values corresponding to the response variable y_i and the covariates $\boldsymbol{x}_i$, $\forall i = 1, \ldots, N$ (excluding the latent variables $\boldsymbol{z}_i$, given that they have already been used for defining the sampling design and will not be included in the LASSO models as previously explained) as well as strata and cluster indicators corresponding to each of them.

5. Sampling design is defined as a two-stage stratified cluster sampling.

First, $a_h = 4$, $\forall h \in \{1, \ldots, H\}$ clusters or PSUs are sampled per stratum (out of $A_h = 20$ clusters in the population stratum U_h).

Afterward, a different number of units (denoted as $n_{h,\dot{\alpha}}$) is sampled from each selected PSU (or cluster) $\dot{\alpha} \in \mathbb{A}_h$ of stratum $h \in \{1, \ldots, H\}$. In particular, $\forall h \in \{1, \ldots, H\}$ and $\forall \dot{\alpha} \in \mathbb{A}_h$ the following number of units $n_{h,\dot{\alpha}}$ have been

sampled in each scenario:

$$\textbf{S1(G):}\ n_{1,\dot{\alpha}} = 500,\ n_{2,\dot{\alpha}} = 50,\ n_{3,\dot{\alpha}} = 25,\ n_{4,\dot{\alpha}} = 10,\ n_{5,\dot{\alpha}} = 5,$$
$$\textbf{S1(B):}\ n_{1,\dot{\alpha}} = 5,\ n_{2,\dot{\alpha}} = 10,\ n_{3,\dot{\alpha}} = 25,\ n_{4,\dot{\alpha}} = 50,\ n_{5,\dot{\alpha}} = 500,$$

$$\textbf{S2(G):}\ n_{1,\dot{\alpha}} = 250,\ n_{2,\dot{\alpha}} = 100,\ n_{3,\dot{\alpha}} = 50,\ n_{4,\dot{\alpha}} = 25,\ n_{5,\dot{\alpha}} = 5,$$
$$\textbf{S2(B):}\ n_{1,\dot{\alpha}} = 5,\ n_{2,\dot{\alpha}} = 25,\ n_{3,\dot{\alpha}} = 50,\ n_{4,\dot{\alpha}} = 100,\ n_{5,\dot{\alpha}} = 250.$$

Note that the names of the scenarios refer to the distribution of the response variable: "G" for Gaussian distribution with reference to the framework of the linear regression and "B" for the Bernoulli distribution indicating the logistic regression framework. In this way, a total of four different scenarios were defined as a combination of S1 and S2 ($d = 0$ and $d = 5$ cluster-level variables), and the response variable considered (continuous ("G") or dichotomous ("B")).

6. As previously defined in eq. (2.28), the sampling weights are then calculated as indicated in eq. (5.42):

$$w_i = \sum_{h=1}^{H} \sum_{\dot{\alpha} \in \mathbb{A}_h} \frac{N_{h,\dot{\alpha}}}{n_{h,\dot{\alpha}}} \cdot \frac{A_h}{a_h} \cdot 1_{S_{h,\dot{\alpha}}}(i), \quad \forall i \in S. \tag{5.42}$$

5.3.2 Set-up

As explained above, in this simulation study we aim to compare the performance of the replication methods described in 5.2.3, to the w-SRSCV, and to the unw-SRSCV (both of them described in Section 5.2.2). In order to ease the notation, we could define replicate weights for training and test sets of the w-SRSCV as the original weights, i.e.,

$$w^{*,\mathrm{tr}(t)}_{i,\text{w-SRSCV}} = w^{*,\mathrm{test}(t)}_{i,\text{w-SRSCV}} = w_i,\ \forall i \in S, \tag{5.43}$$

for the t^{th} training and test sets, $\forall t = 1, \ldots, T_{\text{w-SRSCV}} = K$. In the same way, the unw-SRSCV would be equivalent to setting all the replicate weights for training and test sets to one:

$$w^{*,\mathrm{tr}(t)}_{i,\text{unw-SRSCV}} = w^{*,\mathrm{test}(t)}_{i,\text{unw-SRSCV}} = 1,\ \forall i \in S, \tag{5.44}$$

for the t^{th} training and test sets, $\forall t = 1, \ldots, T_{\text{w-SRSCV}} = K$. Considering this notation, the lines below describe the process of the simulation study for all the methods, including w-SRSCV and unw-SRSCV.

In order to compare the performance and analyze the validity of different meth-

ods for selecting the optimal tuning parameter for LASSO regression models, the steps described below are followed. Given the length and complexity of the simulation process, we separate it into three different parts and provide the Figures 5.7, 5.8 and 5.11 for a more visual explanation.

First, we need to obtain the true optimal tuning parameter that minimizes the error of the model fitted to the whole sample S in the finite population and calculate the number of variables that are kept in the model when considering this tuning parameter. This parameter and the corresponding number of covariates are the ones that we would like to obtain in practice based on our analysis. Thus, we start the simulation study as follows (see a graphical summary in Figure 5.7).

For $r = 1, \ldots, R$:

Step 1. Obtain the sample S^r.

Step 2. Define the penalty grid based on S^r using the default approach in `glmnet` (Friedman et al. 2010): $\lambda_1^r, \ldots, \lambda_L^r$. It should be noted that the number of elements in the grid, L, also depends on r. Nevertheless, in order not to complicate the notation too much, we avoid using more under or superscripts to indicate it.

Step 3. For each value λ_l^r, $\forall l = 1, \ldots, L$:

Step 3.1 Fit the LASSO model to S^r ($\hat{f}^{r,l}(\cdot)$) considering the vector of covariates $\boldsymbol{x}_i$, sampling weights w_i, $\forall i \in S^r$ and λ_l^r following eq. (5.14).

Step 3.2 Apply the model $\hat{f}^{r,l}(\cdot)$ to the finite population and estimate the response: $\hat{f}^{r,l}(\boldsymbol{x}_i)$, $\forall i = 1, \ldots, N$.

Step 3.3 Calculate the error of the model $\hat{f}^{r,l}(\cdot)$ in the finite population (i.e., true population error of the model fitted to S^r):

$$\widehat{Err}^r_{\text{true}}(\lambda_l^r) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(y_i, \hat{f}^{r,l}(\boldsymbol{x}_i)), \tag{5.45}$$

where $\mathcal{L}(y_i, \hat{f}^{r,l}(\boldsymbol{x}_i))$ is calculated as in eq. (5.9).

Step 3.4 Define the "true" optimal tuning parameter as the one that minimizes the true population error (not available in practice):

$$\Lambda^r_{\text{true}} = \underset{\lambda_l^r \,:\, l \in \{1,\ldots,L\}}{\operatorname{argmin}} \{\widehat{Err}^r_{\text{true}}(\lambda_l^r)\}. \tag{5.46}$$

Then, we calculate the tuning parameter that we would actually obtain by applying each of the methods considered in this work. We continue the simulation study as follows (depicted in Figure 5.8):

Step 4. For each method m, where $m \in \{$JKn, dCV, Bootstrap, BRR, split-cv, split-boot, extrap, w-SRSCV, unw-SRSCV$\}$:

Step 4.1 Define training and test sets following Section 5.2.3 ($S^{r,m}_{\text{tr}(t)}$ and $S^{r,m}_{\text{test}(t)}$, $\forall t = 1, \ldots, T_m$, respectively) and calculate the corresponding replicate weights for the sampled units: $w^{*,r,\text{tr}(t)}_{i,m}$ and $w^{*,r,\text{test}(t)}_{i,m}$, $\forall i \in S^r$.

Step 4.2 For $t = 1, \ldots, T_m$ and $l = 1, \ldots, L$:

Step 4.2.1 Fit the model to $S^{r,m}_{\text{tr}(t)}$ considering λ^r_l and the corresponding replicate weights $w^{*,r,\text{tr}(t)}_{i,m}$ following eq. (5.14): $\hat{f}^{r,l,m}_{\text{tr}(t)}(\cdot)$.

Step 4.2.2 Apply $\hat{f}^{r,l,m}_{\text{tr}(t)}(\cdot)$ to $\forall i \in S^{r,m}_{\text{test}(t)}$ and estimate the response: $\hat{f}^{r,l,m}_{\text{tr}(t)}(\boldsymbol{x}_i)$. Calculate the error of the training model in the test set (this is the error that can be estimated in practice):

$$\widehat{Err}^{r,m,t}_{\text{test}}(\lambda^r_l) = \frac{1}{\sum_{i \in S^{r,m}_{\text{test}(t)}} w^{*,r,\text{test}(t)}_{i,m}} \sum_{i \in S^{r,m}_{\text{test}(t)}} w^{*,r,\text{test}(t)}_{i,m} \mathcal{L}(y_i, \hat{f}^{r,l,m}_{\text{tr}(t)}(\boldsymbol{x}_i)). \tag{5.47}$$

Step 4.3 Define the average error of the training models in the test sets:

$$\widehat{Err}^{r,m}_{\text{test}}(\lambda^r_l) = \frac{1}{T_m} \sum_{t=1}^{T_m} \widehat{Err}^{r,m,t}_{\text{test}}(\lambda^r_l). \tag{5.48}$$

It should be noted that the process followed for the JKn method is a bit different from the rest. For the JKn method, a unique error is calculated considering all the test sets jointly. Thus, the error in eq. (5.47) is not calculated for this method, and in contrast, the test error of the training models in the test sets is estimated as shown in eq. (5.49) considering the sample S^r as a whole:

$$\widehat{Err}^{r,\text{JKn}}_{\text{test}}(\lambda^r_l) = \frac{\sum_{t=1}^{T_{\text{JKn}}} \sum_{i \in S^{r,\text{JKn}}_{\text{test}(t)}} w^{*,r,\text{test}(t)}_{i,\text{JKn}} \cdot \mathcal{L}(y_i, \hat{f}^{r,l,\text{JKn}}_{\text{tr}(t)}(\boldsymbol{x}_i))}{\sum_{t=1}^{T_{\text{JKn}}} \sum_{i \in S^{r,\text{JKn}}_{\text{test}(t)}} w^{*,r,\text{test}(t)}_{i,\text{JKn}}}, \tag{5.49}$$

where $w^{*,r,\text{test}(t)}_{i,\text{JKn}}$ have been defined in eq. (5.21).

Step 4.4 Define the optimal tuning parameter for method m as follows:

$$\Lambda_{\text{test}}^{r,m} = \underset{\lambda_l^r \,:\, l\in\{1,\ldots,L\}}{\text{argmin}} \{\widehat{Err}_{\text{test}}^{r,m}(\lambda_l^r)\}. \tag{5.50}$$

Finally, we analyze the performance of each method m by comparing the parameters obtained based on it to the true population parameters:

Step 5. Fit the models to S^r by considering Λ_{true}^r and $\Lambda_{\text{test}}^{r,m}$ in eq. (5.14), and denote them as $\hat{f}_{\text{true}}^{r,\star}(\cdot)$ and $\hat{f}_{\text{test}}^{r,\star,m}(\cdot)$, being the corresponding model coefficients denoted as $\hat{\boldsymbol{\beta}}_{\text{true}}^{r,\star}$ and $\hat{\boldsymbol{\beta}}_{\text{test}}^{r,\star,m}$, respectively. Then:

Step 5.1 Define the difference between the true optimal tuning parameter and the one obtained based on method m as follows:

$$\text{diff}^{r,m} = \log(\Lambda_{\text{test}}^{r,m}) - \log(\Lambda_{\text{true}}^r). \tag{5.51}$$

Step 5.2 Define δ_{true}^r and $\delta_{\text{test}}^{r,m}$ as the number of regression coefficients different to 0, when fitting LASSO models considering the tuning parameters Λ_{true}^r and $\Lambda_{\text{test}}^{r,m}$, respectively. That is,

$$\delta_{\text{true}}^r = \sum_{j=1}^{p} I(\hat{\beta}_{\text{true},j}^{r,\star} \neq 0) \quad \text{and} \quad \delta_{\text{test}}^{r,m} = \sum_{j=1}^{p} I(\hat{\beta}_{\text{test},j}^{r,\star,m} \neq 0). \tag{5.52}$$

The former indicates the number of variables that would be selected based on LASSO if the finite population were available (i.e., the "true" number of variables selected based on LASSO), while the latter indicates the number of variables that would be selected based on each method m.

Step 5.3 For each selected tuning parameter Λ_{true}^r and $\Lambda_{\text{test}}^{r,m}$, the true population error (obtained by applying the models $\hat{f}_{\text{true}}^{r,\star}(\cdot)$ and $\hat{f}_{\text{test}}^{r,\star,m}(\cdot)$ to the finite population) is calculated following eq. (5.45):

$$\widehat{Err}_{\text{true}}^{r}(\Lambda_{\text{true}}^r) = \frac{1}{N}\sum_{i\in U} \mathcal{L}(y_i, \hat{f}_{\text{true}}^{r,\star}(\boldsymbol{x}_i)), \tag{5.53}$$

$$\widehat{Err}_{\text{true}}^{r}(\Lambda_{\text{test}}^{r,m}) = \frac{1}{N}\sum_{i\in U} \mathcal{L}(y_i, \hat{f}_{\text{test}}^{r,\star,m}(\boldsymbol{x}_i)). \tag{5.54}$$

In this simulation study, a total of $R = 500$ samples were obtained. Cross-validation methods were applied for $K = 10$ number of folds, $B = 200$ Bootstrap

resamples were considered, and a total of 20 train and test sets were defined for split and extrap methods. For split methods, 70% of clusters are used for defining training sets, while for extrap training sets are defined by means of 3 out of 5 strata. All computations were performed in (64 bit) R 4.2.0 (R Core Team 2022) and a workstation equipped with 32GB of RAM, an Intel i7-8700 processor (3.20 Ghz), and a Windows 10 operating system. In particular, LASSO models were fitted by means of `glmnet` R package (Friedman et al. 2010) and for applying JKn, BRR and Bootstrap methods `survey` package (Lumley 2020) was used.

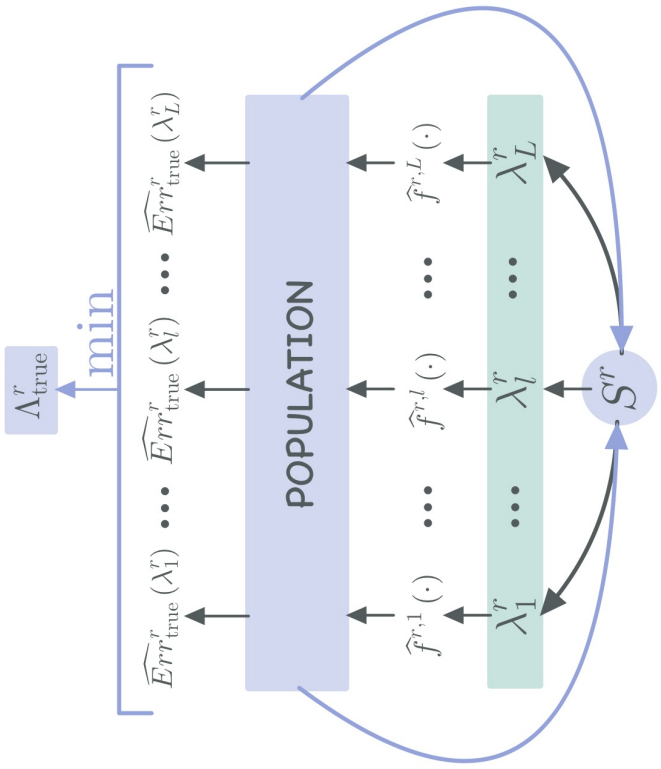


Figure 5.7: Graphical summary of Step 3 of the simulation study set-up.

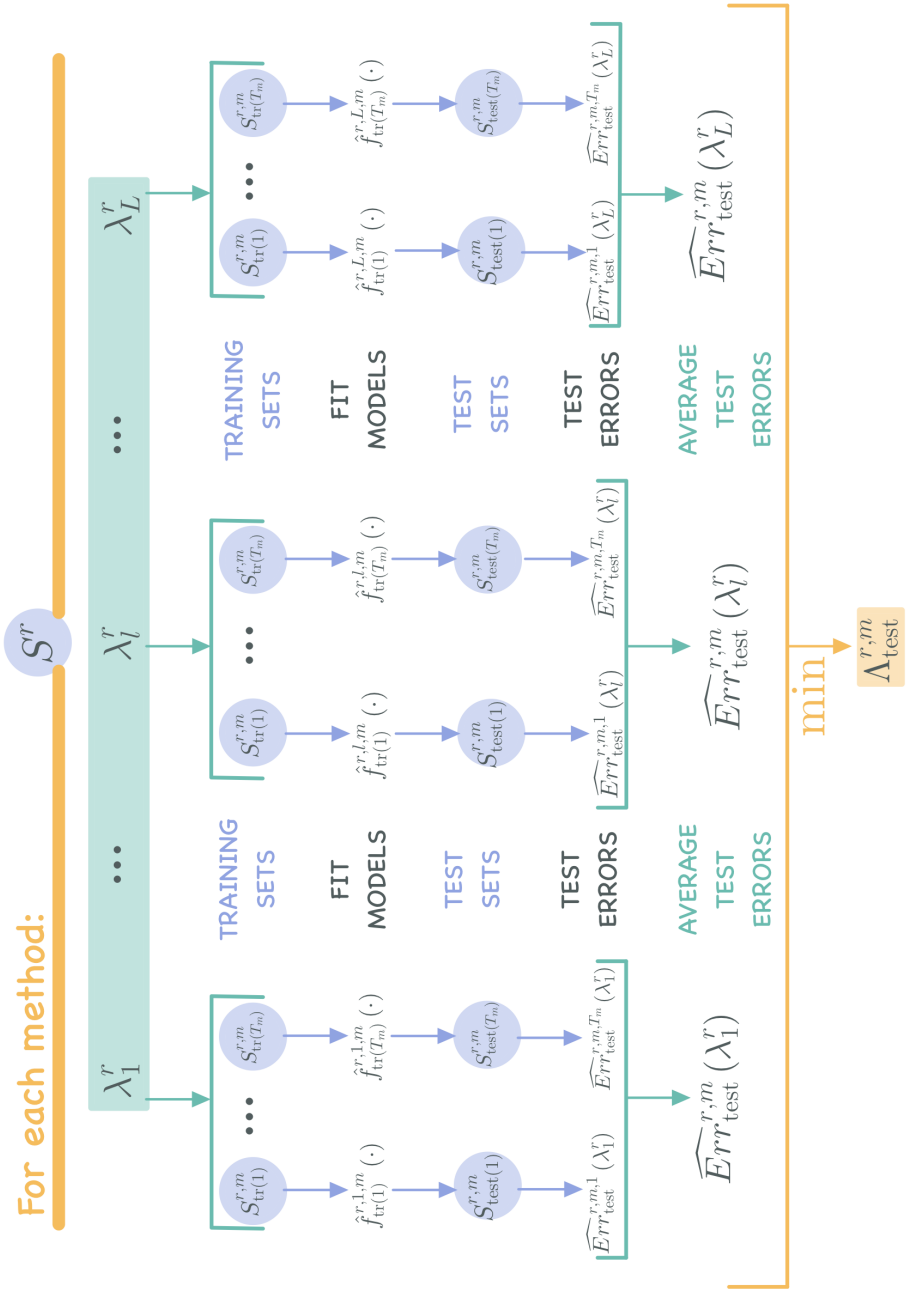


Figure 5.8: Graphical summary of Step 4 of the simulation study set-up.

5.3.3 Results

In this section, we summarize the main results of the simulation study, related to the performance of the analyzed methods when selecting the optimal tuning parameter and the number of covariates that end up in the final models.

Specifically, we compare the optimal tuning parameters obtained based on each method, the number of covariates kept in the model by those parameters, and the error of those models to the true population parameters described in Section 5.3.2. In particular, Figure 5.9 depicts the differences between the logarithms of the true optimum tuning parameter and the ones obtained based on each method (see eq. (5.51)), while Figure 5.10 shows the number of variables that would be selected based on those tuning parameters. Numerical results for linear (S1(G) and S2(G)) and logistic (S1(B) and S2(B)) models with cluster-level variables (S2(G) and S2(B)) or without them (S1(G) and S1(B)) are described in Tables 5.1 and 5.2. Due to the large number of results, we proceed to summarize the main findings.

The unw-SRSCV performs poorly in all scenarios, it selects unnecessarily complex models with a large number of variables. In particular, in more than 50% of the samples in scenario S1(G), all the 50 covariates are kept in the final model based on this method. This method is also the one with the highest error in all scenarios except in S1(G), where the extrap method showed the worst results in this aspect. Such a bad performance indicates the need to consider sampling weights when fitting LASSO models to complex survey data. In contrast, BRR, split-cv, split-boot and, in particular, extrap methods select large tuning parameters, that lead to models with very few numbers of covariates, increasing the population error estimated based on them.

The performance of the rest of the methods depends on the scenario. No great differences have been observed comparing the results obtained from scenarios related to linear (S1(G) and S2(G)) and logistic (S1(B) and S2(B)) regression models. However, the results obtained in scenarios with cluster-level variables (S2(G) and S2(B)) or without them (S1(G) and S1(B)) differ considerably for some methods. In S1(G) and S1(B) the selected tuning parameters based on the JKn and the dCV are unbiased with respect to the true population parameter, which leads to keeping a similar number of variables in the final models. In S2(G) and S2(B), these methods select slightly greater tuning parameters, which leads them to select in general models with less number of variables. Nevertheless, there are no great differences in terms of error, compared to the error that would be obtained if the true tuning pa-

rameter were selected. Therefore, it can be concluded that the performance of these methods is correct in all scenarios. In S1(G) and S1(B), there are no differences between w-SRSCV and dCV methods, and as mentioned above, all of them perform quite properly. In contrast, in S2(G) and S2(B), the tuning parameter selected by means of w-SRSCV is lower than the true one, which leads to select unnecessarily complex models with a large number of parameters, without a gain in terms of the error of the model, in comparison to the dCV method. The Bootstrap method is the one that performs the best in terms of the error in all the scenarios. However, its performance in terms of the number of variables of the selected models depends on the scenario: even though it shows a good performance in selecting a similar number of covariates to the true model in S2(G) and S2(B), it tends to select a too large number of covariates in the models corresponding to S1(G) and S1(B).

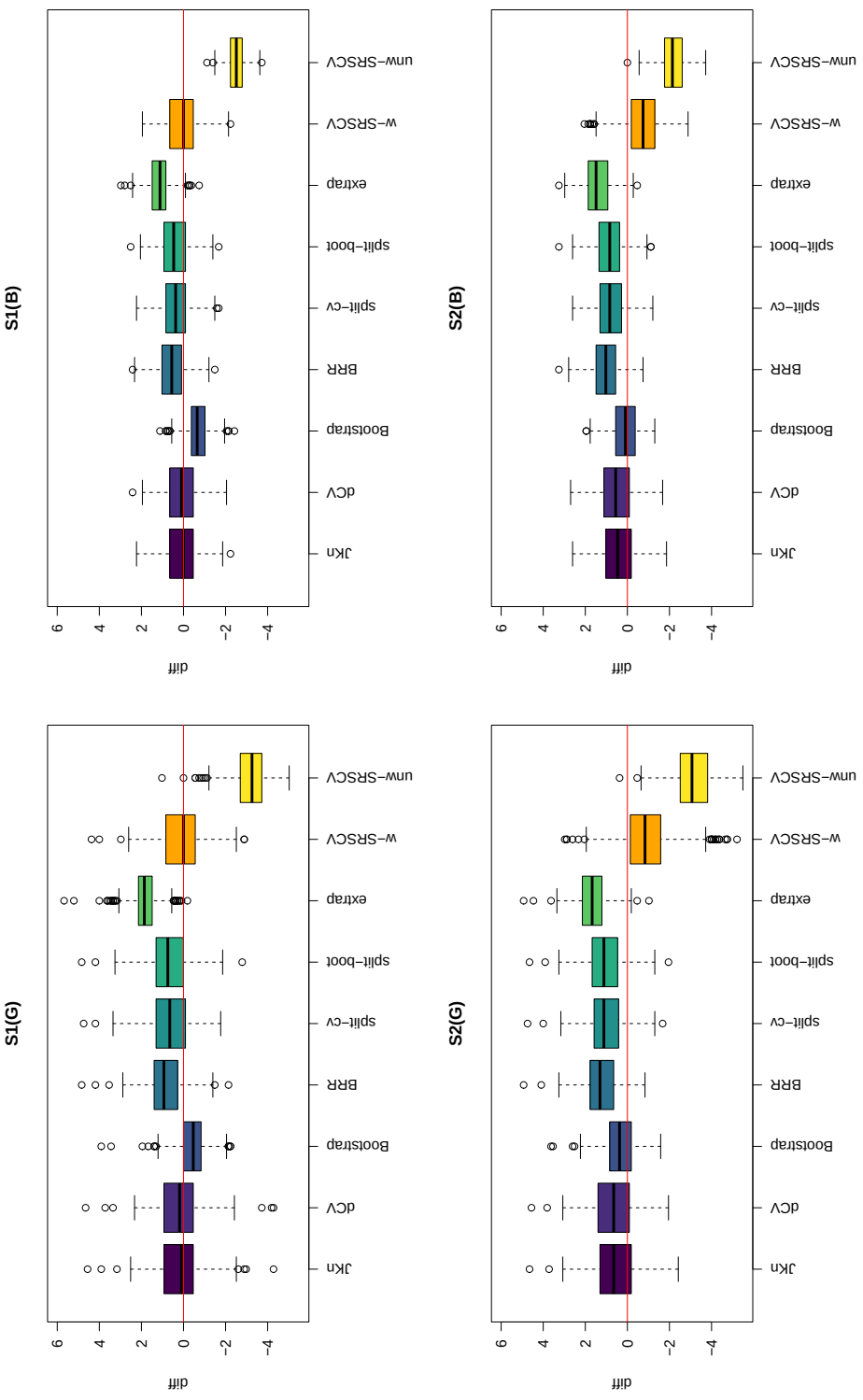


Figure 5.9: Box-plots of the differences between the logarithm of the true optimal tuning parameter (Λ^r_{true}) and the one obtained based on each analyzed method ($\Lambda^{r,m}_{\text{test}}$) across $R = 500$ samples in all the scenarios.

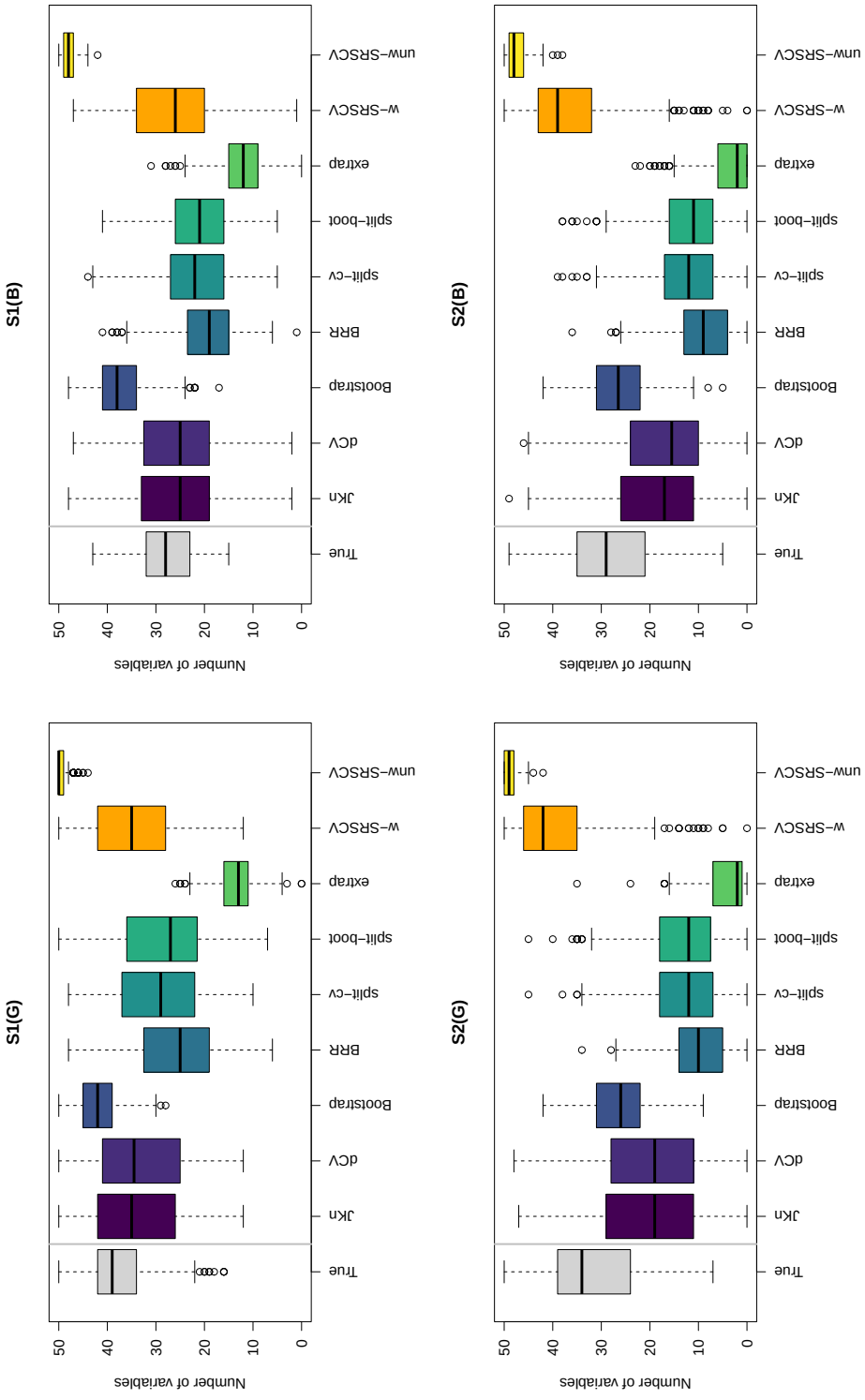


Figure 5.10: Box-plots of the number of variables considered in models fitted across $R = 500$ samples in all the scenarios considering the optimal tuning parameters selected based on each method m (i.e., $\delta^{r,m}_{\text{test}}$). Gray box-plot (denoted as "True") indicates the number of variables of the models fitted to the same samples considering the true optimal tuning parameters that minimize the population error (i.e., δ^{r}_{true}).

Table 5.1: Summary of the numerical results obtained in the simulation study in scenarios S1(G) and S2(G). For each scenario, the minimum (min) and maximum (max) values, the average (mean) and standard deviation (sd), the median and interquartile range (Q1-Q3) are displayed for the logarithm of the optimal tuning parameters (Λ^r_{true} and $\Lambda^{r,m}_{\text{test}}$) and the corresponding population errors ($\widehat{Err}^r_{\text{true}}(\Lambda^r_{\text{true}})$ and $\widehat{Err}^r_{\text{true}}(\Lambda^{r,m}_{\text{test}})$) obtained across $R = 500$ samples for each analyzed method m.

	S1(G)				**S2(G)**			
			log(λ)				**log(λ)**	
	min	**max**	**mean (sd)**	**median (Q1-Q3)**	**min**	**max**	**mean (sd)**	**median (Q1-Q3)**
True	-4.78	0.92	-0.92 (0.60)	-0.90 (-1.23, -0.58)	-3.91	0.93	-0.52 (0.67)	-0.58 (-0.97, -0.03)
JKn	-4.49	0.79	-0.77 (0.76)	-0.73 (-1.25, -0.21)	-1.82	1.37	0.06 (0.58)	0.10 (-0.33, 0.48)
dCV	-4.80	0.79	-0.71 (0.75)	-0.70 (-1.17, -0.16)	-2.24	1.40	0.11 (0.58)	0.15 (-0.29, 0.53)
Bootstrap	-2.35	-0.63	-1.35 (0.30)	-1.34 (-1.55, -1.16)	-1.12	0.70	-0.17 (0.22)	-0.19 (-0.31, -0.04)
BRR	-2.35	0.93	-0.08 (0.48)	-0.02 (-0.41, 0.30)	-0.35	1.54	0.70 (0.29)	0.68 (0.51, 0.88)
split-cv	-1.97	0.93	-0.33 (0.55)	-0.29 (-0.72, 0.11)	-1.14	1.37	0.50 (0.37)	0.51 (0.26, 0.73)
split-boot	-3.27	1.11	-0.28 (0.58)	-0.24 (-0.68, 0.18)	-1.14	1.44	0.51 (0.36)	0.53 (0.30, 0.73)
extrap	-0.11	1.63	0.90 (0.19)	0.91 (0.80, 1.02)	-0.54	1.70	1.11 (0.22)	1.13 (0.97, 1.25)
w-SRSCV	-3.31	0.56	-0.81 (0.67)	-0.76 (-1.24, -0.32)	-4.96	1.20	-1.40 (0.92)	-1.36 (-1.74, -0.82)
unw-SRSCV	-4.87	-2.20	-4.08 (0.64)	-4.27 (-4.63, -3.63)	-5.08	-2.17	-3.64 (0.72)	-3.52 (-4.14, -3.08)
			Error				**Error**	
	min	**max**	**mean (sd)**	**median (Q1-Q3)**	**min**	**max**	**mean (sd)**	**median (Q1-Q3)**
True	240.55	266.16	252.34 (4.73)	252.51 (248.92, 255.39)	269.98	303.91	285.76 (5.74)	285.85 (281.65, 289.52)
JKn	240.73	283.70	256.21 (6.84)	255.53 (251.49, 260.04)	275.10	490.81	295.20 (15.27)	292.56 (288.28, 298.73)
dCV	240.88	285.33	256.27 (6.77)	255.64 (251.62, 260.10)	275.57	490.81	295.31 (14.62)	292.83 (288.30, 298.08)
Bootstrap	241.04	284.29	254.55 (6.36)	254.07 (250.04, 258.05)	277.98	504.60	291.60 (14.53)	289.03 (286.00, 293.34)
BRR	242.95	304.80	258.84 (7.56)	258.18 (254.04, 262.82)	284.93	405.67	298.89 (8.77)	297.22 (292.78, 303.65)
split-cv	240.73	291.18	257.30 (7.17)	256.54 (252.53, 261.02)	281.55	442.20	297.09 (10.63)	294.89 (291.09, 301.17)
split-boot	241.74	303.09	257.86 (7.40)	256.93 (253.18, 261.62)	281.72	442.20	297.13 (10.80)	295.00 (291.46, 300.93)
extrap	253.49	329.70	276.76 (12.32)	275.49 (267.76, 283.74)	288.68	442.20	306.51 (9.02)	308.40 (301.13, 311.66)
w-SRSCV	240.99	290.83	255.80 (6.30)	255.28 (251.53, 259.43)	271.68	597.24	298.29 (25.92)	291.34 (285.29, 303.47)
unw-SRSCV	242.04	296.92	259.99 (8.36)	259.05 (253.90, 265.43)	269.99	667.99	307.57 (33.50)	299.21 (289.83, 315.53)

Table 5.2: Summary of the numerical results obtained in the simulation study in scenarios S1(B) and S2(B). For each scenario, the minimum (min) and maximum (max) values, the average (mean) and standard deviation (sd), the median and interquartile range (Q1-Q3) are displayed for the logarithm of the optimal tuning parameters (Λ^r_{true} and $\Lambda^{r,m}_{\text{test}}$) and the corresponding population errors ($\widehat{Err}^r_{\text{true}}(\Lambda^r_{\text{true}})$ and $\widehat{Err}^r_{\text{true}}(\Lambda^{r,m}_{\text{test}})$) obtained across $R = 500$ samples for each analyzed method m.

	S1(B)				**S2(B)**			
			log(λ)				**log(λ)**	
	min	**max**	**mean (sd)**	**median (Q1-Q3)**	**min**	**max**	**mean (sd)**	**median (Q1-Q3)**
True	-5.55	-2.79	-4.05 (0.38)	-4.07 (-4.30, -3.81)	-5.96	-2.73	-4.04 (0.53)	-4.06 (-4.40, -3.66)
JKn	-5.59	-2.73	-3.98 (0.53)	-3.93 (-4.37, -3.56)	-5.35	-2.49	-3.59 (0.51)	-3.53 (-3.93, -3.23)
dCV	-5.39	-2.82	-3.95 (0.51)	-3.88 (-4.35, -3.53)	-5.21	-2.49	-3.52 (0.49)	-3.46 (-3.82, -3.17)
Bootstrap	-5.59	-3.93	-4.74 (0.27)	-4.74 (-4.92, -4.55)	-4.84	-3.34	-3.94 (0.20)	-3.94 (-4.07, -3.80)
BRR	-4.71	-2.45	-3.51 (0.34)	-3.47 (-3.70, -3.28)	-4.05	-2.36	-3.04 (0.27)	-3.06 (-3.21, -2.87)
split-cv	-5.03	-2.81	-3.68 (0.42)	-3.62 (-3.95, -3.36)	-4.42	-2.36	-3.24 (0.35)	-3.22 (-3.46, -2.99)
split-boot	-5.12	-2.64	-3.64 (0.41)	-3.59 (-3.88, -3.34)	-4.37	-2.43	-3.20 (0.34)	-3.20 (-3.40, -2.98)
extrap	-3.82	-2.12	-2.91 (0.22)	-2.89 (-3.04, -2.76)	-3.48	-2.14	-2.65 (0.20)	-2.63 (-2.76, -2.53)
w-SRSCV	-5.65	-2.38	-4.04 (0.52)	-4.02 (-4.40, -3.62)	-6.33	-2.75	-4.71 (0.61)	-4.72 (-5.12, -4.41)
unw-SRSCV	-7.00	-5.91	-6.58 (0.22)	-6.59 (-6.74, -6.43)	-6.99	-5.22	-6.21 (0.27)	-6.21 (-6.40, -6.03)
			Error				**Error**	
	min	**max**	**mean (sd)**	**median (Q1-Q3)**	**min**	**max**	**mean (sd)**	**median (Q1-Q3)**
True	0.46	0.50	0.48 (0.01)	0.48 (0.48, 0.49)	0.53	0.58	0.55 (0.01)	0.55 (0.55, 0.56)
JKn	0.47	0.60	0.49 (0.01)	0.49 (0.48, 0.50)	0.54	0.92	0.57 (0.02)	0.56 (0.56, 0.57)
dCV	0.47	0.58	0.49 (0.01)	0.49 (0.48, 0.50)	0.54	0.89	0.57 (0.02)	0.56 (0.56, 0.57)
Bootstrap	0.47	0.61	0.49 (0.02)	0.49 (0.48, 0.50)	0.54	0.76	0.56 (0.02)	0.56 (0.55, 0.56)
BRR	0.47	0.56	0.49 (0.01)	0.49 (0.48, 0.50)	0.55	0.59	0.57 (0.01)	0.57 (0.56, 0.58)
split-cv	0.47	0.56	0.49 (0.01)	0.49 (0.48, 0.50)	0.55	0.66	0.57 (0.01)	0.57 (0.56, 0.57)
split-boot	0.47	0.57	0.49 (0.01)	0.49 (0.48, 0.50)	0.54	0.63	0.57 (0.01)	0.57 (0.56, 0.57)
extrap	0.48	0.59	0.51 (0.02)	0.51 (0.50, 0.52)	0.55	0.59	0.58 (0.01)	0.58 (0.57, 0.59)
w-SRSCV	0.47	0.60	0.49 (0.01)	0.49 (0.48, 0.50)	0.53	1.21	0.58 (0.05)	0.56 (0.55, 0.58)
unw-SRSCV	0.48	0.68	0.53 (0.02)	0.53 (0.52,0.55)	0.54	1.31	0.60 (0.06)	0.59 (0.57, 0.62)

5.3.4 Analyzing the differences between dCV and w-SRSCV

In the results shown in the previous section, we have seen that the dCV and w-SRSCV methods behave differently under some scenarios. In this section, we will further analyze the behavior of these two methods. The difference between the two methods is based on the way in which the training and test subsets are created. For this reason, we analyze the adequacy of the subsets obtained by the dCV and w-SRSCV methods to represent the population and the sampling design. For this purpose, we add the following steps to the simulation set-up described in Section 5.3.2 incorporating some modifications to **Step 4.**, as summarized in Figure 5.11 and described below:

Step 4.$(*)$ For each method m, where $m \in \{\text{dCV, w-SRSCV}\}$ and $r = 1, \ldots, 500$:

Step 4.1$(*)$ Define training and test sets ($S^{r,m}_{\text{tr}(t)}$ and $S^{r,m}_{\text{test}(t)}$, $\forall t = 1, \ldots, T_m$, respectively) and calculate the corresponding replicate weights for the sampled units: $w^{*,r,\text{tr}(t)}_{i,m}$ and $w^{*,r,\text{test}(t)}_{i,m}$, $\forall i \in S^r$.

Step 4.2$(*)$ For $t = 1, \ldots, T_m$ and $l = 1, \ldots, L$:

Step 4.2.1$(*)$ Fit the model to $S^{r,m}_{\text{tr}(t)}$ considering λ^r_l and the corresponding replicate weights $w^{*,r,\text{tr}(t)}_{i,m}$ following eq. (5.14): $\hat{f}^{r,l,m}_{\text{tr}(t)}(\cdot)$.

Step 4.2.2$(*)$ Apply $\hat{f}^{r,l,m}_{\text{tr}(t)}(\cdot)$ to $\forall i \in U$ and calculate the error of the model in the population (this indicates the true performance of the training model in the finite population):

$$\widehat{Err}^{r,m,t}_{\text{tr.pop}}(\lambda^r_l) = \frac{1}{N} \sum_{i \in U} \mathcal{L}(y_i, \hat{f}^{r,l,m}_{\text{tr}(t)}(\boldsymbol{x}_i)). \tag{5.55}$$

Step 4.3$(*)$ Define the average error of the training models in the finite population:

$$\widehat{Err}^{r,m}_{\text{tr.pop}}(\lambda^r_l) = \frac{1}{T_m} \sum_{t=1}^{T_m} \widehat{Err}^{r,m,t}_{\text{tr.pop}}(\lambda^r_l). \tag{5.56}$$

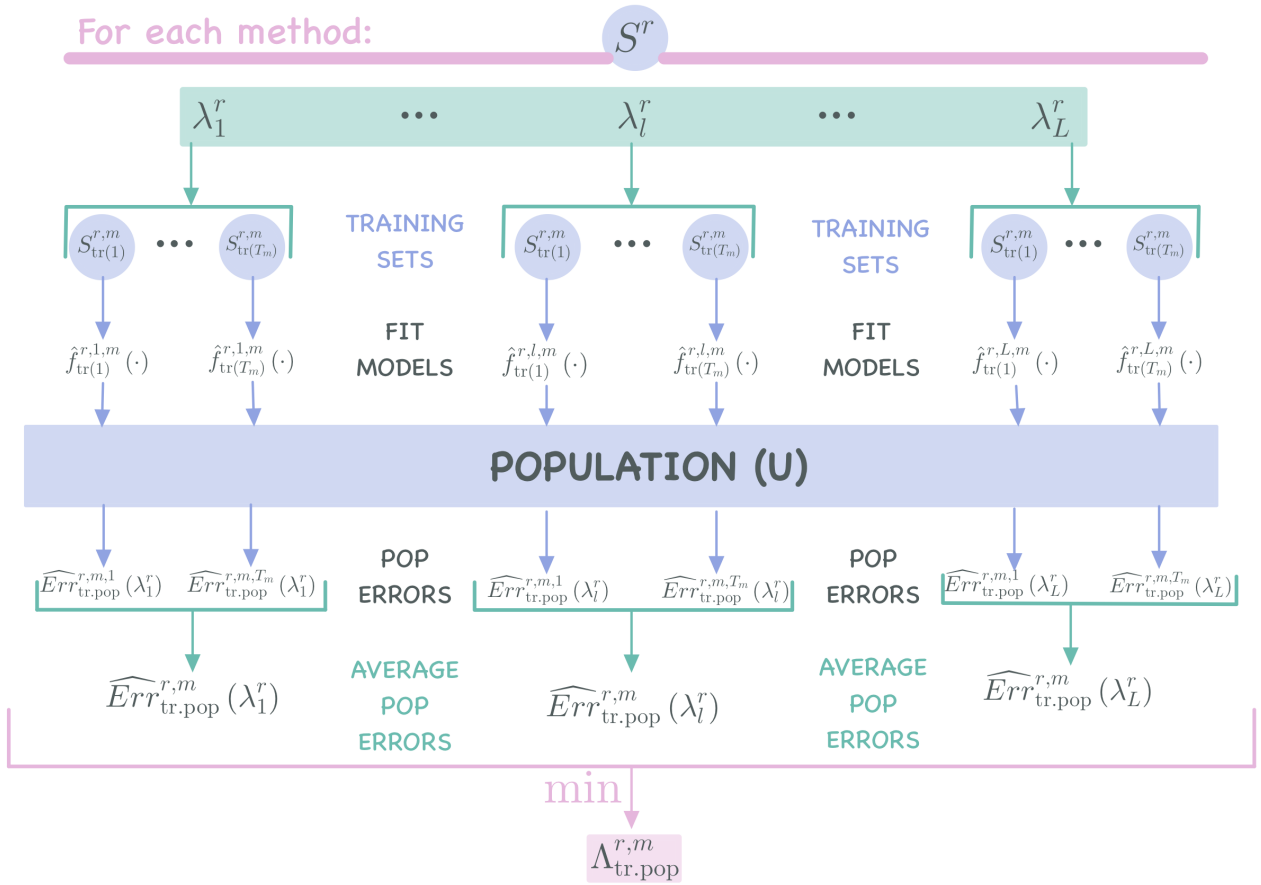


Figure 5.11: Graphical summary of the Step 4(*) of the simulation study set-up performed to analyze in more detail the behavior of dCV and w-SRSCV methods.

On the one hand, Figures 5.12 and 5.13 compare $\widehat{Err}^{r,m}_{\text{tr.pop}}(\lambda^r_l)$ defined in eq. (5.56) (i.e., the true error of the training models estimated in the finite population) to $\widehat{Err}^{r}_{\text{true}}(\lambda^r_l)$ defined in eq. (5.45) (that is, the finite population error of the model fitted to the whole sample S^r), $\forall r = 1, \ldots, R$ and $\forall l = 1, \ldots, L$ for the dCV and w-SRSCV methods, respectively, in all the scenarios depicted in Section 5.3.1. It can be observed that the finite population error of the training models $\widehat{Err}^{r,m}_{\text{tr.pop}}(\lambda^r_l)$ is quite similar to the error of the model fitted to the whole sample $\widehat{Err}^{r}_{\text{true}}(\lambda^r_l)$, for both, dCV (see Figure 5.12) and w-SRSCV (Figure 5.13) methods. From these results, we can conclude that the training models fitted based on both methods represent properly the models fitted to the whole samples and the differences observed in Section 5.3.3 between dCV and w-SRSCV methods do not occur due to poor performance of the training models.

On the other hand, Figures 5.14 and 5.15 compare $\widehat{Err}^{r,m}_{\text{test}}(\lambda^r_l)$ defined in eq. (5.48) (i.e., the error of the training models estimated in the test sets) to $\widehat{Err}^{r}_{\text{true}}(\lambda^r_l)$ defined in eq. (5.45), $\forall r = 1, \ldots, R$ and $\forall l = 1, \ldots, L$ for the dCV and w-SRSCV methods, respectively, in all the scenarios depicted in Section 5.3.1. It can be observed that the results vary depending on the scenario. In Scenarios S1(G) and S1(B), the performance of both methods is quite similar, and their test error $\widehat{Err}^{r,m}_{\text{test}}(\lambda^r_l)$ approximates the true population error $\widehat{Err}^{r}_{\text{true}}(\lambda^r_l)$ quite properly, being sometimes above and other times below the true population error (in line with the results shown in Section 5.3.3 for Scenarios S1(G) and S1(B)). In contrast, in Scenarios S2(G) and S2(B), the performance of the methods and the way in which they approximate the true population error $\widehat{Err}^{r}_{\text{true}}(\lambda^r_l)$ differ, and hence, they offer different results in scenarios S2(G) and S2(B) as shown in Section 5.3.3. The error estimated by means of the dCV method is commonly slightly greater than the true population error. In contrast, the w-SRSCV usually underestimates the true population error. Note that in the w-SRSCV method, the same clusters are part of both, training and test sets. Then all the clusters are considered for fitting the models as well as for estimating the error. Hence, the way of creating training and test sets based on w-SRSCV does not represent properly the real relationship between the sample and population, given that only a small proportion of clusters in the population are used to fit the models to the samples. The sampling design is better represented in the dCV method, in which some of the clusters are used to fit the models and others to estimate the error.

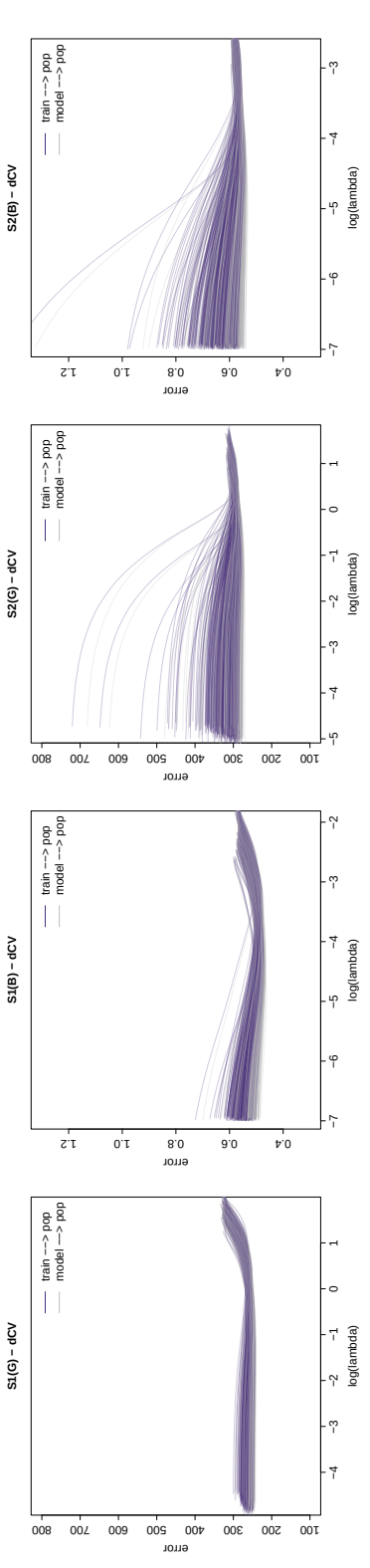


Figure 5.12: Estimated error of the training models created based on dCV method in the finite population ($\widehat{Err}^{r,m}_{\text{tr.pop}}(\lambda_l^r)$, in purple) and the true population error ($\widehat{Err}^{r}_{\text{true}}(\lambda_l^r)$, in gray) in Scenarios S1(G), S1(B), S2(G) and S2(G).

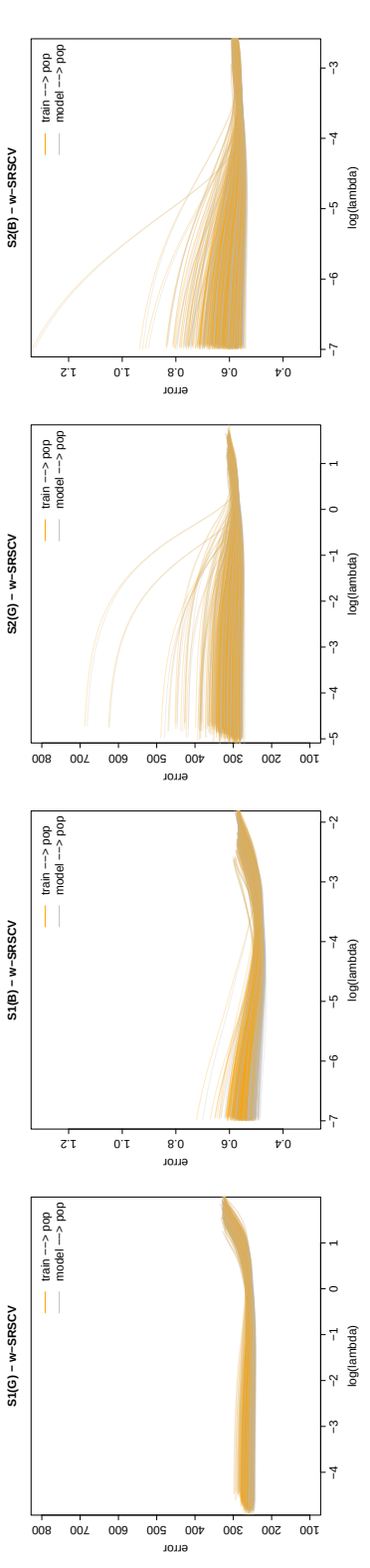


Figure 5.13: Estimated error of the training models created based on w-SRSCV method in the finite population ($\widehat{Err}^{r,m}_{\text{tr.pop}}(\lambda_l^r)$, in orange) and the true population error ($\widehat{Err}^{r}_{\text{true}}(\lambda_l^r)$, in gray) in Scenarios S1(G), S1(B), S2(G) and S2(G).

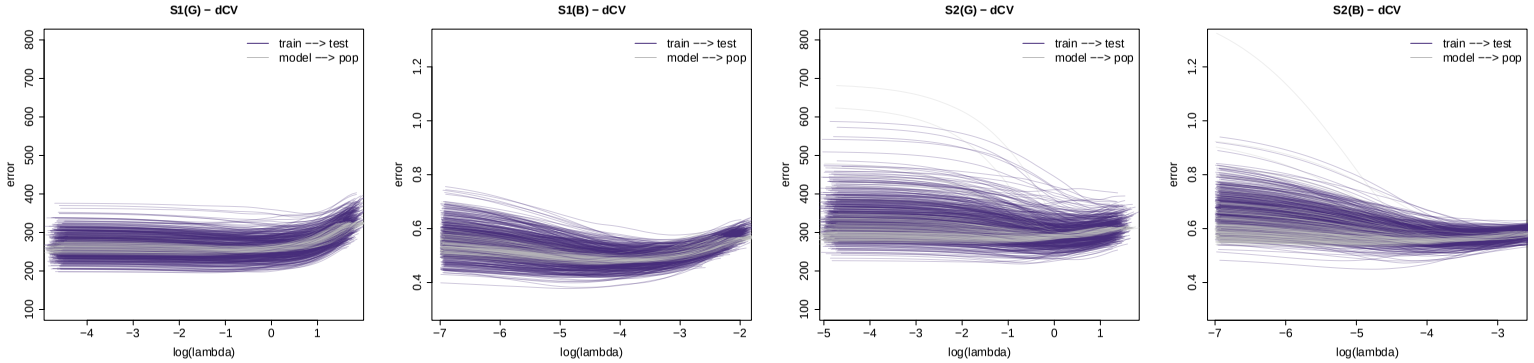


Figure 5.14: Estimated error of the training models in the test sets created based on dCV ($\widehat{Err}^{r,m}_{\text{test}}(\lambda^r_l)$, in purple) and the true population error ($\widehat{Err}^{r}_{\text{true}}(\lambda^r_l)$, in gray) in Scenarios S1(G), S1(B), S2(G) and S2(G).

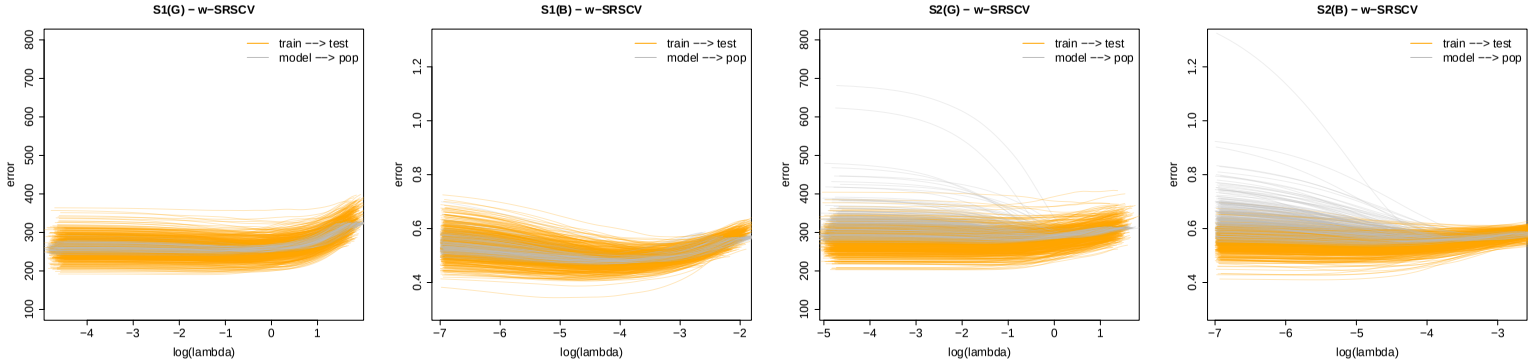


Figure 5.15: Estimated error of the training models in the test sets created based on w-SRSCV ($\widehat{Err}^{r,m}_{\text{test}}(\lambda^r_l)$, in purple) and the true population error ($\widehat{Err}^{r}_{\text{true}}(\lambda^r_l)$, in gray) in Scenarios S1(G), S1(B), S2(G) and S2(G).

5.4 Discussion

In this study, we worked on the variable selection process by means of LASSO regression models for complex survey data. As discussed throughout the chapter, two issues need to be analyzed before implementing LASSO regression to complex surveys. In the first place, the need to incorporate sampling weights into the estimation process of LASSO models should be checked. In addition, the validity of the traditional cross-validation techniques commonly applied to simple random samples for selecting the tuning parameters for LASSO models should also be analyzed when working with complex survey data. In this chapter, the performance of methods based on replicate weights that are well-known for other purposes in complex survey data framework but, to our knowledge, have never been used for LASSO, have been compared to the traditional cross-validation techniques to select the tuning parameter λ. In addition, new methods based on the idea of replicating weights have been proposed, among others, the dCV. This method could be seen as an extension of the Survey CV method proposed by Wieczorek et al. (2022), which in combination with replicate weights, allows us to be more flexible when defining different folds, and thus, it is valid for more types of designs, for example when a different number of PSUs per stratum is available, or a few numbers of PSUs per stratum are sampled.

The performance of all those methods for selecting the tuning parameter for LASSO models has been compared by means of an extensive simulation study. The sampling design considered in this study is a two-stage stratified cluster sampling in which a different number of units are sampled from each cluster. Let us highlight some of the most interesting conclusions of the simulation study in the following lines.

In the first place, the bad performance of the unw-SRSCV, which leads to very complex regression models selecting almost all variables, shows the need to incorporate sampling weights into the estimation process of LASSO regression models. It should be noted that in this work we have not considered the option to fit "perfect" prediction models (i.e., prediction models for which the sampling design is uninformative given the covariates included in the model). In line with previous works (see, e.g., Pfeffermann (1993), Scott and Wild (1986), Sugden and Smith (1984)), if we try to fit perfect models sampling weights are not needed in the estimation process of linear and logistic regression models. These conclusions can also be extended to LASSO models, and hence, this method would perform properly in that situation. However, it is important to point out, that when working with LASSO regression

models, as we are using a sparse shrinkage estimator, the sampling design must be uninformative given, not all the covariates, but the ones that actually end up in the final model, which is even more complicated and beyond the researcher's control. In addition, it should also be noted that when we work with real data, we will hardly ever be able to fit "perfect" regression models. Therefore, we would not recommend the use of this method in practice, in order to avoid fitting too complex regression models with biased estimates of regression coefficients.

The second point that should be mentioned is the similarities and differences between the performance of the w-SRSCV (which does not consider the sampling design when defining folds) and the new proposal dCV. It is striking that for the same sampling design (two-stage stratified cluster sampling), such different results were obtained across different scenarios. This fact could be explained as follows. In the scenario where cluster-level variables were incorporated, most of the variability induced by the sampling design could be explained by means of the sampling weights. In contrast, the inclusion of cluster-level variables leads to an increase in the effect of the sampling design that cannot be explained by means of the sampling weights themselves, thus offering greater differences between one method and the other. When no cluster-level variable is considered in the model, both methods perform properly and lead to reasonable and parsimonious regression models. In contrast, when including cluster-level variables into the process, models selected based on those methods differ considerably, being the ones selected by the w-SRSCV more complex than necessary. This is in line with the results obtained by Lumley and Scott (2015), in which the effect of the sampling design has shown an important role in the model selection, in particular, on the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). Briefly, this work shows that for samples with greater design effect, more parsimonious models are selected, given that the design effect penalizes more strongly the incorporation of the covariates into the model. Coming back to our study, we have observed that the greater the cluster effect, the greater the differences between the tuning parameters selected for fitting the models and the number of variables selected based on those methods. The w-SRSCV tends to select a larger number of variables than the dCV. Therefore, we recommend the use of the dCV rather than the w-SRSCV, in order to select more parsimonious models when fitting LASSO regression models to complex survey data. In addition, given the similarities of this work and Lumley and Scott (2015), we believe that the trace of the variance-covariance matrix (also used to define dAIC and dBIC parameters) could be used to analyze the cluster effect and thus diagnose

whether there may be differences between the dCV and w-SRSCV. However, the magnitude of the relationship between the trace of the variance-covariance matrix and the differences between w-SRSCV and dCV on the variable selection by means of LASSO will be further analyzed in future work.

Another consequence of the abovementioned cluster effect, which is reflected in the differences between the dCV and w-SRSCV, is the so-called "data leakage" (Kaufman et al. 2011). When the cluster effect is significant, splitting the clusters between training and test sets, (i.e., setting some units of a cluster into the training set and others into the test set as w-SRSCV does), has two consequences. On the one hand, we fit the models with more information than we should, given that all the clusters are considered in the process. The fact that all the clusters are considered when fitting the training models means that the sampling variability will be underestimated across them. On the other hand, very similar information to the one used when fitting the models is used to evaluate the error in the test sets given that, actually, the training and test sets are not independent sets since units of the same cluster are in both of them. Thus, the true population error is also underestimated. When applying the training models to the test sets, it can be observed that the w-SRSCV underestimates the true population error in contrast to the dCV, particularly in the scenarios with cluster-level variables. We have also observed that the variability of the training models is greater when the dCV is applied compared to the w-SRSCV (results not shown).

Note that the methods proposed and applied throughout this work can be extended in a very simple way to data obtained from a one-stage stratified sampling design. However, the behavior of the methods in such a situation has not been analyzed in this simulation study. The authors expect that the results may be similar to S1, where cluster-level variables have not been incorporated and have shown to have a low cluster-effect, but this should be studied in future work to be confirmed. Neither other types of sampling, such as sampling probability proportional to sample size nor post-stratification have been considered, so the conclusions obtained are limited to the schemes we have analyzed. In order to reduce the number of results shown, we set the number of folds in the cross-validation methods as $K = 10$, given that it is the one most commonly used in the literature (see, e.g., Witten et al. (2016)). Note also that other tries have been made by changing the number of folds in the methods based on cross-validation to $K = 5$, but no significant differences have been observed (results not shown). Also, cross-validation techniques allow repeating the process of splitting the sample several times, which is usually known as

cross-validation with replication. Those replicates have not been considered in the results shown in this chapter for the same reason. In this simulation study, we did not consider the method proposed by Wieczorek et al. (2022) as another alternative. First, it should be noted that the goal of our proposal is not to improve the performance of the Survey CV in the situations in which it works, but to gain flexibility and to be applicable in more situations. It should be noted that in the scenarios we analyzed, the Survey CV can only be applied with $K = a_h = 4$ folds, which is not usually considered in practice, and hence, we believe that the comparisons we would make would not be very interesting in that situation, and the design of more scenarios would distract the readers from the main goal of the study. We also believe that under scenarios in which both methods, Survey CV as well as dCV can be considered, they will probably perform similarly. However, a specific simulation study would be necessary to compare the performance of both methods and analyze their pros and cons in each situation. We should also comment that we have decided to discuss the number of variables that end up in the final LASSO models rather than to contrast whether those covariates are actually the ones that form the theoretical model. The main reason to take this decision is that given that the covariates are correlated and taking into account that we are working with simulated data in which the covariates do not have any particular meaning, we found it quite difficult to quantify whether the final models selected based on different methods are close or far from the theoretical one. However, we find it interesting to analyze this point in more detail as further work by means of a simulation study based on real data. Finally, the methods applied and proposed in this work could be used for other purposes beyond LASSO to define partially independent subsets of the sample. For example, it is straightforward the application of the analyzed methods to fit other types of models, such as ridge regression which is appropriate to deal with multicollinearity problems (Hoerl and Kennard 1970, Kidwell and Brown 1982) or elastic nets which is a combination of LASSO and ridge regression (Zou and Hastie 2005). However, the performance of the methods should be previously checked in that context.

In summary, the methods that have performed the best in all the scenarios are the dCV and the JKn, which have shown a similar performance in the scenarios that have been considered. It should also be noted the good performance of the Bootstrap, particularly in the scenarios with cluster-level variables, being the best in terms of error but considerably less parsimonious than JKn and the dCV. In addition, in terms of computational efficiency, the fastest has been the dCV, beating JKn and

Bootstrap, being twice as fast as JKn on average and between 15 and 20 times faster than Bootstrap in the scenarios that have been considered. In particular, in the scenarios that have been analyzed, the dCV needs on average between 0.26-0.45 seconds in both linear scenarios (S2(G)-S1(G)) and 1.09-1.70 seconds in logistic scenarios (S2(B)-S1(B)), JKn needs between 0.48-0.72 and 2.18-3.43 seconds on average and the Bootstrap between 5.28-7.43 and 16.68-23.8 seconds on average, respectively, in the same scenarios. In summary, this study shows that the K-fold cross-validation technique, which is commonly applied to select tuning parameters for fitting LASSO regression models to simple random samples, can be extended to the dCV when working with complex survey data and it will provide parsimonious regression models. For this reason, we recommend the use of this method for fitting LASSO regression models to complex survey data.

CHAPTER 6

Estimation of the Receiver Operating Characteristic (ROC) curve and the area under the curve (AUC)

The paper related to the work presented in this chapter has been published:

Iparragirre, A., Barrio, I., & Arostegui, I. (2023). Estimation of the ROC curve and the area under it with complex survey data. Stat, *12(1), e635.*

This chapter mostly replicates the above-mentioned article. However, some changes have been made to keep the notation and ensure cohesion with the rest of the document. In addition, Sections 6.2.3 and 6.3.4 incorporate new contents that were not included in the above-mentioned paper.

The R package `wROC` *derived from this work can be found on GitHub*[10] *and is presented in Chapter 8 of this document.*

[10] https://github.com/aiparragirre/wROC

Summary

Before implementing logistic regression models in daily practice, it is necessary to ensure they have an adequate predictive performance and, in particular, a good discrimination ability. In the context of logistic regression models, discrimination ability is usually estimated by means of the receiver operating characteristic (ROC) curve and the area under it (AUC). Traditional estimators of these parameters are thought to be applied to simple random samples, they do not consider complex sampling designs and, hence, are not appropriate for complex survey data. The goal of this work is to propose new weighted estimators for the ROC curve and AUC based on sampling weights. The behavior of the proposed estimators is evaluated and compared to the traditional unweighted ones by means of a simulation study. Finally, weighted and unweighted ROC curve and AUC estimators are applied to real survey data in order to compare the estimates in a real scenario. The results suggest the use of the weighted estimators proposed in this work in order to obtain unbiased estimates for the ROC curve and AUC of logistic regression models fitted to complex survey data.

6.1 Introduction

Given the impact of prediction models in many fields in daily practice, it is necessary to ensure that these models are valid and applicable in practice. In particular, when the goal is prediction, ensuring good model performance is essential. In this chapter, we focus on logistic regression models for dichotomous response variables. Model performance of logistic regression models is usually analyzed by means of calibration and discrimination ability (Steyerberg 2008). Calibration measures the agreement between outcomes and predictions (see, e.g., the goodness-of-fit test proposed by Hosmer and Lemesbow (1980)). In this study, we bring discrimination ability into focus, which measures the ability of the models to distinguish between units with the event of interest and without it. This is usually measured by means of the receiver operating characteristic (ROC) curve, which is defined as the curve formed by specificity and sensitivity parameters (i.e., probability of properly classifying individuals without and with the event of interest, respectively) across all the possible cut-off points (Green and Swets 1966, Pepe 2003, Swets and Pickett 1982). The area under the ROC curve (AUC), is one of the most widely used summary measures to

analyze the discrimination ability of logistic regression models (Pepe 2003). Bamber (1975) showed the equivalence between the area under the ROC curve and the Mann-Whitney U-statistic, offering in this way an interesting interpretation of the AUC as the probability that an individual with the event of interest is given by the model a higher probability of event than an individual without the event of interest.

In the area of complex design surveys, both calibration and discrimination of models have been of interest, and new proposals have been made in both directions in the last 20 years. On the one hand, Archer et al. (2007) proposed a goodness-of-fit test that considers complex sampling designs to analyze the calibration of the models fitted to complex survey data. Similarly, Lumley (2017) proposed a design-consistent estimator for the Cox-Snell and Nagelkerke R^2 (Cox and Snell 1991, Nagelkerke 1991). On the other hand, in the context of the discrimination ability, Yao et al. (2015) proposed a modification of the Mann-Whitney U-statistic in order to consider the sampling design to estimate the AUC of the models, incorporating pairwise sampling weights, which are defined as the inverse joint inclusion probability of a pair of observations (i^*, i^{**}) (Horvitz and Thompson 1952, Särndal et al. 2003), i.e., $w_{i^*i^{**}} = 1/\pi_{i^*i^{**}}$ where, $\pi_{i^*i^{**}} = P[(i^* \in S) \cap (i^{**} \in S)]$, $\forall i^*, i^{**} \in S$.

As mentioned previously, in this work, we aim to focus on the evaluation of the discrimination ability of logistic regression models. Even though Yao et al. (2015) proposed a weighted estimator for the AUC, to our knowledge, there is a lack of proposals for estimating the ROC curve considering complex sampling designs. Therefore, the main goal of this work is to propose a weighted estimator for the ROC curve. In particular, we propose weighted specificity and sensitivity estimators to define a new weighted estimator for the ROC curve. In addition, we calculate the area under the curve in order to estimate the AUC following Bamber (1975) and Tsuruta and Bax (2006), and finally, we show that this AUC estimator defined as the area under the weighted estimate of the ROC curve is equal to the weighted Mann-Whitney U-statistic considering marginal sampling weights w_i, $\forall i \in S$, rather than pairwise sampling weights $w_{i^*i^{**}}$, $\forall i^*, i^{**} \in S$ as proposed by Yao et al. (2015). The estimation of the AUC is then a simple weighted expression that can easily be calculated in practice, given that the marginal sampling weights are usually explicitly available when working with complex survey data, in contrast to the pairwise sampling weights, which usually need to be calculated by means of some computational package. The performance of this proposal is analyzed by means of a simulation study, in which the weighted and unweighted estimates of the ROC curve and AUC are compared to the true population ones. In addition, the proposed methods are

applied to real survey data, and the weighted estimates of the ROC curve and AUC are compared to the unweighted ones.

The rest of the chapter is organized as follows. In Section 6.2, we first set the basic notation needed to ease the reading of this chapter. Then, we define the proposed weighted estimator of the ROC curve and we calculate the area under it. We continue by proving the equivalence between the area under the weighted estimate of the ROC curve and the weighted Mann-Whitney U-statistic, considering marginal sampling weights. Finally, we define the AUC estimator proposal based on the pairwise sampling weights (Yao et al. 2015). In Section 6.3, the simulation study conducted in order to analyze the performance of the proposed estimators is defined and the results obtained are depicted and summarized. In Section 6.4, the proposed estimators are applied to real survey data. Finally, the chapter concludes with a discussion in Section 6.5.

6.2 Methods

The goal of this section is to describe our proposal to estimate the ROC curve of logistic regression models fitted to complex survey data considering marginal sampling weights. We calculate the area under the weighted estimate of the ROC curve in order to estimate the AUC and we show the equivalence between this area and a modification of the Mann-Whitney U-statistic considering marginal sampling weights, which leads us to conclude that this estimator can be used in order to obtain unbiased estimates of the AUC.

The rest of the section is organized as follows. In Section 6.2.1 we denote the basic notation related to the logistic regression model, ROC curve, and AUC. In Section 6.2.2, we define our proposal to consider sampling weights to estimate the ROC curve and the area under the curve (AUC) and we show the equivalence between the area under the weighted estimate of the ROC curve and the Mann-Whitney U-statistic considering marginal sampling weights. Finally, in Section 6.2.3 we define the proposal of Yao et al. (2015) based on the pairwise sampling weights for estimating the AUC.

6.2.1 Background and basic notation

Let us remind the basic notation of logistic regression models, focusing on their discrimination ability and defining, in particular, the receiver Operating Characteristic

(ROC) curve and the area under the ROC curve (AUC).

Let $p(\boldsymbol{x}_i) = P(Y = 1|\boldsymbol{X} = \boldsymbol{x}_i)$ indicate the conditional probability of event for an individual i given the values of its vector of covariates $\boldsymbol{x}_i$. As defined in eq. (2.52), the specific form of the logistic regression model in terms of the probability of event is:

$$p(\boldsymbol{x}_i) = \frac{e^{\boldsymbol{x}_i\boldsymbol{\beta}}}{1 + e^{\boldsymbol{x}_i\boldsymbol{\beta}}}. \tag{6.1}$$

Based on the probability $p(\boldsymbol{x}_i)$ and a cut-off point c, each individual can be classified as event (if $p(\boldsymbol{x}_i) \geq c$) or non-event ($p(\boldsymbol{x}_i) < c$). However, this classification may be correct or incorrect depending on the selected cut-off point c. The correct classifications, based on a particular cut-off point c, are usually quantified by specificity ($Sp(c)$) and sensitivity ($Se(c)$) parameters, which are defined as the probabilities of correctly classifying the non-events and events, respectively, i.e.,

$$Sp(c) = P\left[p(\boldsymbol{x}_i) < c|Y = 0\right] \quad \text{and} \quad Se(c) = P\left[p(\boldsymbol{x}_i) \geq c|Y = 1\right]. \tag{6.2}$$

The ROC curve is defined as the set of pairs $1 - Sp(c)$ and $Se(c)$ across all the possible cut-off points c (Green and Swets 1966, Swets and Pickett 1982), i.e.,

$$ROC(\cdot) = \{(1 - Sp(c),\ Se(c)),\ c \in (-\infty,\ \infty)\}. \tag{6.3}$$

The discrimination ability of a logistic regression model is usually evaluated by means of the AUC, which is defined as the area under the ROC curve defined in eq. (6.3). The AUC ranges from 0.5 (an uninformative model) to 1 (a perfect model in terms of discrimination) (Steyerberg 2008).

Before going through the estimation of the above-mentioned parameters in the context of complex survey data, let us explain the process for simple random samples. Let S indicate a sample of n observations of the vector of random variables $(Y, \boldsymbol{X})$, i.e., $\{(y_i, \boldsymbol{x}_i)\}_{i=1}^{n}$. Let $\widehat{\boldsymbol{\beta}}$ indicate the vector of estimated regression coefficients, estimated (for the moment) by means of the likelihood function in eq. (6.4) (first defined in eq. (2.53)) and let $\hat{p}_i = \hat{p}(\boldsymbol{x}_i)$ be the corresponding estimated probabilities of event, $\forall i \in S$ (McCullagh and Nelder 1989):

$$L(\boldsymbol{\beta}) = \prod_{i \in S} p(\boldsymbol{x}_i)^{y_i}(1 - p(\boldsymbol{x}_i))^{1-y_i}. \tag{6.4}$$

Let S_0 and S_1 be the subsamples of sizes n_0 and n_1 formed by the units without the event of interest and with the event of interest, respectively (note that $S_0 \cap S_1 = \varnothing$

and $S_0 \cup S_1 = S$). In order to distinguish between the units with and without the event of interest, let us use $i_0 \in S_0$ as the indicator for the units without the event of interest and $i_1 \in S_1$ for the units with the event of interest, hereinafter. In practice, specificity and sensitivity parameters for a particular cut-off point c are estimated as proportions of correctly classified sampled non-events and events, respectively (see, e.g., Pepe (2003)), i.e.:

$$\widehat{Sp}(c) = \frac{1}{n_0} \sum_{i_0 \in S_0} I(\hat{p}_{i_0} < c) \quad \text{and} \quad \widehat{Se}(c) = \frac{1}{n_1} \sum_{i_1 \in S_1} I(\hat{p}_{i_1} \geq c), \tag{6.5}$$

where $I(\cdot)$ denotes the indicator function, which takes the value 1 if the expression between brackets is satisfied, and 0 otherwise. Then, the estimated ROC curve is defined by means of each estimated pair of sensitivity and specificity parameters, for each possible cut-off point (Pepe 2003) as shown in eq. (6.6):

$$\widehat{ROC}(\cdot) = \left\{ (1 - \widehat{Sp}(c),\ \widehat{Se}(c)),\ c \in (-\infty,\ \infty) \right\}. \tag{6.6}$$

Bamber (1975) showed that the area under the ROC curve defined in eq. (6.6) can be estimated ($\widehat{AUC}$) as described in eq. (6.7), by means of the Mann-Whitney U-statistic:

$$\widehat{AUC} = \frac{1}{n_0 \cdot n_1} \sum_{i_0 \in S_0} \sum_{i_1 \in S_1} \left[I(\hat{p}_{i_0} < \hat{p}_{i_1}) + 0.5 I(\hat{p}_{i_0} = \hat{p}_{i_1}) \right]. \tag{6.7}$$

However, in the context of complex survey data, we aim to obtain information related to the finite population of interest U. If the whole finite population were known, we would be able to fit a logistic regression model by maximizing the population likelihood defined in eq. (2.57), i.e.,

$$L_{\text{pop}}(\boldsymbol{\beta}) = \prod_{i \in U} p(\boldsymbol{x}_i)^{y_i} (1 - p(\boldsymbol{x}_i))^{1 - y_i}, \tag{6.8}$$

and computing in this way the model coefficients (i.e., $\boldsymbol{\beta}^{\text{pop}}$) and the corresponding probabilities of event for all the units in the finite population p_i^{pop}, $\forall i \in U$ where,

$$p_i^{\text{pop}} = p^{\text{pop}}(\boldsymbol{x}_i) = \frac{e^{\boldsymbol{x}_i \boldsymbol{\beta}^{\text{pop}}}}{1 + e^{\boldsymbol{x}_i \boldsymbol{\beta}^{\text{pop}}}}. \tag{6.9}$$

Let N_0 and N_1 indicate the sizes of the subsets formed by the non-events (U_0) and events (U_1) of the finite population U. Then, the ROC curve and AUC of the

population model could be easily calculated following the definitions given above as follows, respectively:

$$ROC_{\text{pop}}(\cdot) = \{(1 - Sp_{\text{pop}}(c),\ Se_{\text{pop}}(c)),\ c \in (-\infty, \infty)\}\,, \tag{6.10}$$

where,

$$Sp_{\text{pop}}(c) = \frac{1}{N_0} \sum_{i_0 \in U_0} I(p_{i_0}^{\text{pop}} < c) \quad \text{and} \quad Se_{\text{pop}}(c) = \frac{1}{N_1} \sum_{i_1 \in U_1} I(p_{i_1}^{\text{pop}} \geq c), \tag{6.11}$$

and,

$$AUC_{\text{pop}} = \frac{1}{N_0 \cdot N_1} \sum_{i_0 \in U_0} \sum_{i_1 \in U_1} \left[I(p_{i_0}^{\text{pop}} < p_{i_1}^{\text{pop}}) + 0.5 I(p_{i_0}^{\text{pop}} = p_{i_1}^{\text{pop}}) \right]. \tag{6.12}$$

However, information on covariates and/or the response variable is not available for the finite population U, but only for a sample S obtained sampling U following some complex design. Thus, the model, as well as the ROC curve and AUC, need to be estimated based uniquely on S. As discussed throughout this dissertation, the regression coefficients and the corresponding probabilities of events are estimated by maximizing the pseudo-likelihood function (Binder 1983) shown in eq. (6.13) (defined in eq. (2.58)) in the context of complex surveys:

$$PL(\boldsymbol{\beta}) = \prod_{i \in S} p(\boldsymbol{x}_i)^{y_i w_i} \left(1 - p(\boldsymbol{x}_i)\right)^{(1-y_i) w_i}. \tag{6.13}$$

Taking into account the results obtained in Chapter 4, all the model parameters for the samples will be estimated based on the pseudo-likelihood function in eq. (6.13) (the likelihood function in eq. (6.4) will not be considered again in the whole chapter). Hence, for ease of notation, let us indicate the estimated regression coefficients as $\hat{\boldsymbol{\beta}}$ (instead of $\hat{\boldsymbol{\beta}}_w$, which was used in Chapter 4). In addition, for the same reason, $\hat{p}_i = \hat{p}(\boldsymbol{x}_i), \forall i \in S$ indicate the predicted probabilities for the sampled units obtained based on the pseudo-likelihood function in eq. (6.13), hereinafter.

At this point, it should be noted that the model fitted to the population and the model fitted to the sample are not exactly the same models, given that regression coefficient estimates $\boldsymbol{\beta}^{\text{pop}}$ (population model) and $\hat{\boldsymbol{\beta}}$ (sample model) may differ. In practice, the regression coefficients $\boldsymbol{\beta}^{\text{pop}}$ are not known and the model that will be applied is the one defined by $\hat{\boldsymbol{\beta}}$ (i.e., the sample model). Thus, we are particularly interested in analyzing whether the sample model (rather than the population model)

has a good discrimination ability or not. That is, the question we aim to address is the following: how good is the fitted sample model to discriminate between units with and without the event of interest? To answer this question, we first wonder about what the "true" discrimination ability of the sample model is and how it is defined. Let us think in the following way. The real discrimination ability of the sample model can be seen as the ability of that model to discriminate between units with and without the event of interest considering all the possible units, i.e., considering the whole finite population U. Hence, in case the finite population U were available in practice, the way to obtain the ROC curve and AUC of the sample model would be to extend it to the finite population and estimate those parameters by following the definitions given in eqs. (6.15) and (6.17) below. Specifically, the probabilities of event could be estimated for all the units in the finite population considering $\hat{\boldsymbol{\beta}}$ obtained by maximizing the pseudo-likelihood function in eq. (6.13):

$$\hat{p}_i = \hat{p}(\boldsymbol{x}_i) = \frac{e^{\boldsymbol{x}_i\hat{\boldsymbol{\beta}}}}{1 + e^{\boldsymbol{x}_i\hat{\boldsymbol{\beta}}}}, \quad \forall i \in U. \tag{6.14}$$

Then, the population ROC curve of the sample model is estimated as follows:

$$\widehat{ROC}_{\text{true}}(\cdot) = \left\{(1 - \widehat{Sp}_{\text{true}}(c),\ \widehat{Se}_{\text{true}}(c)),\ c \in (-\infty, \infty)\right\}, \tag{6.15}$$

where,

$$\widehat{Sp}_{\text{true}}(c) = \frac{1}{N_0}\sum_{i_0 \in U_0} I(\hat{p}_{i_0} < c) \quad \text{and} \quad \widehat{Se}_{\text{true}}(c) = \frac{1}{N_1}\sum_{i_1 \in U_1} I(\hat{p}_{i_1} \geq c). \tag{6.16}$$

Similarly, the sample model's AUC in the finite population could then be defined as follows:

$$\widehat{AUC}_{\text{true}} = \frac{1}{N_0 \cdot N_1}\sum_{i_0 \in U_0}\sum_{i_1 \in U_1}\left[I(\hat{p}_{i_0} < \hat{p}_{i_1}) + 0.5I(\hat{p}_{i_0} = \hat{p}_{i_1})\right]. \tag{6.17}$$

At this point, we think it is important to remark on the differences between the ROC curves defined in eqs. (6.10) and (6.15), and the AUCs in eqs. (6.12) and (6.17). The ROC curve and AUC defined in eqs. (6.10) and (6.12) (i.e., ROC_{pop} and AUC_{pop}) are the parameters that indicate the real (theoretical) discrimination ability of the model fitted to the whole finite population (which is not available in practice). That is, they indicate the real discrimination ability of the best model we could fit with the considered covariates, given that the model is fitted with all the

possible units, and hence, it is reasonable to assume that its performance will be better than the performance of the model fitted to the sample S. In contrast, the ROC curve and AUC defined in eqs. (6.15) and (6.17) (i.e., $\widehat{ROC}_{\text{true}}$ and $\widehat{AUC}_{\text{true}}$), indicate the real discrimination ability of the model fitted to the sample S (i.e., its ability to discriminate between units with and without the event of interest in the whole finite population). These real discrimination ability parameters $\widehat{ROC}_{\text{true}}$ and $\widehat{AUC}_{\text{true}}$ are neither available in practice, given that to obtain them, we would need the information of the whole finite population. Hence, in order to analyze the discrimination ability of the model fitted to the sample, we will be particularly interested in properly estimating the $\widehat{ROC}_{\text{true}}$ and $\widehat{AUC}_{\text{true}}$ parameters.

We believe that in the context of complex survey data, if the ROC curve and the AUC of the fitted model are estimated based on eqs. (6.6) and (6.7), which were designed to be applied in simple random samples and do not consider the sampling weights, then biased estimates can be obtained. For this reason, we propose a new estimator for the ROC curve and the AUC, which considers the sampling weights to estimate the ROC curve and AUC of the model fitted to S. This proposal is described in Section 6.2.2 below.

6.2.2 Proposal

In this section, we first propose an estimator to estimate the ROC curve for logistic regression models fitted with complex survey data and the AUC as the area under the curve. Then, we show the equivalence between the proposed AUC estimator and the Mann-Whitney U-Statistic incorporating marginal sampling weights. Finally, we define the AUC estimator that considers pairwise sampling weights instead of the marginal ones (Yao et al. 2015).

Estimation of the ROC curve and the area under it

We propose to estimate the ROC curve considering the sampling weights, as follows:

$$\widehat{ROC}_w(\cdot) = \left\{(1 - \widehat{Sp}_w(c),\ \widehat{Se}_w(c)),\ c \in (-\infty, \infty)\right\}, \tag{6.18}$$

for which specificity and sensitivity parameters are estimated by means of the sampling weights:

$$\widehat{Sp}_w(c) = \frac{\sum_{i_0 \in S_0} w_{i_0} \cdot I(\hat{p}_{i_0} < c)}{\sum_{i_0 \in S_0} w_{i_0}} \quad \text{and} \quad \widehat{Se}_w(c) = \frac{\sum_{i_1 \in S_1} w_{i_1} \cdot I(\hat{p}_{i_1} \geq c)}{\sum_{i_1 \in S_1} w_{i_1}}. \tag{6.19}$$

Therefore, we propose to calculate the area under $\widehat{ROC}_w(\cdot)$ in order to estimate the AUC (Tsuruta and Bax 2006). Let us denote as $\mathcal{A}$ the area under the curve. We now proceed to describe how the area under the ROC curve defined in eq. (6.18) can be calculated. Note that in practice, we always work with finite sample sizes and, hence, the number of different estimated probabilities is finite. Let us denote as Q the total number of different estimated probabilities, i.e., $\hat{p}^{(Q)} < \ldots < \hat{p}^{(1)}$ (where $Q \leq n$, being $Q = n$ if and only if all the estimated probabilities for each sampled unit are different). Note that for every cut-off point chosen between two ordered probabilities, the same values for the specificity and sensitivity parameters will be obtained, and therefore, the same pair $(1 - \widehat{Sp}_w(c), \widehat{Se}_w(c))$ will be obtained. Then, the ROC curve will be completely defined with $Q + 1$ different cut-off points. Specifically, the smallest possible cut-off point is $c_Q < \hat{p}^{(Q)}$, which will classify all the sampled units as events and therefore, the estimate of the sensitivity will be 1 and the specificity will be 0 (see eq. (6.19)), i.e., the cut-off point c_Q will draw the following point in the ROC curve:

$$(1 - \widehat{Sp}_w(c_Q),\ \widehat{Se}_w(c_Q)) = (1, 1). \tag{6.20}$$

In the same way, the point drawn in the ROC curve for $c_0 > \hat{p}^{(1)}$ will be the following one:

$$(1 - \widehat{Sp}_w(c_0),\ \widehat{Se}_w(c_0)) = (0, 0). \tag{6.21}$$

Let us denote and sort the rest of the $Q - 1$ cut-off points as follows:

$$c_Q < c_{Q-1} < c_{Q-2} < \ldots < c_2 < c_1 < c_0. \tag{6.22}$$

For ease of notation, $\forall q = 1, \ldots, Q - 1$, each cut-off point c_q can be defined as the average value of the probabilities $\hat{p}^{(q+1)}$ and $\hat{p}^{(q)}$, i.e.,

$$c_q = \frac{\hat{p}^{(q+1)} + \hat{p}^{(q)}}{2}, \quad \forall q = 1, \ldots, Q - 1. \tag{6.23}$$

Note that in this way, all the defined cut-off points will be different from the estimated probabilities, and since between any two different ordered predicted probabilities a cut-off point has been defined, only one different predicted probability lies in the interval $[c_q,\ c_{q-1})$, $\forall q = 1, \ldots, Q$. Each cut-off point c_q will draw a point of the ROC curve, $(1 - \widehat{Sp}_w(c_q),\ \widehat{Se}_w(c_q))$. In this way, the estimated ROC curve will be a polygonal line defined by Q segments. Each of these segments will define

an area with the abscissa axis. Let us denote as $\mathcal{A}_q$, $\forall q \in \{1, \ldots, Q\}$ each of these areas. A graphical explanation can be seen in Figure 6.1.

We now proceed to calculate analytically the area under the ROC curve defined in eq. (6.18). In particular, as the area $\mathcal{A}_1$ is a triangle of base $[1 - \widehat{Sp}_w(c_1)]$ and height $\widehat{Se}_w(c_1)$, it can be calculated as follows:

$$\mathcal{A}_1 = \frac{[1 - \widehat{Sp}_w(c_1)] \cdot \widehat{Se}_w(c_1)}{2}. \tag{6.24}$$

For $q = 2, \ldots, Q$, the areas $\mathcal{A}_q$ are right-angled trapezoids, the area of which can be easily calculated as the sum of the triangle $\mathcal{A}_q^1$ of base $\widehat{Sp}_w(c_{q-1}) - \widehat{Sp}_w(c_q)$ and height $\widehat{Se}_w(c_q) - \widehat{Se}_w(c_{q-1})$ and rectangle $\mathcal{A}_q^2$ of the same base and height $\widehat{Se}_w(c_{q-1})$:

$$\begin{aligned}
\mathcal{A}_q &= \mathcal{A}_q^1 + \mathcal{A}_q^2 \\
&= \frac{[\widehat{Sp}_w(c_{q-1}) - \widehat{Sp}_w(c_q)] \cdot [\widehat{Se}_w(c_q) - \widehat{Se}_w(c_{q-1})]}{2} \\
&\quad + [\widehat{Sp}_w(c_{q-1}) - \widehat{Sp}_w(c_q)] \cdot \widehat{Se}_w(c_{q-1}) \\
&= \frac{[\widehat{Sp}_w(c_{q-1}) - \widehat{Sp}_w(c_q)] \cdot [\widehat{Se}_w(c_q) + \widehat{Se}_w(c_{q-1})]}{2}.
\end{aligned} \tag{6.25}$$

Then, the area under the $ROC_w(\cdot)$ curve ($\mathcal{A}$) can be calculated as the sum of the areas defined in eqs. (6.24) and (6.25). Note that, $\widehat{Se}_w(c_0) = 0$ and $\widehat{Sp}_w(c_0) = 1$. Then, eq. (6.24) that defines $\mathcal{A}_1$ can be rewritten in terms of those values for convenience. Finally, the area under the curve can be easily calculated as follows:

$$\begin{aligned}
\mathcal{A} &= \mathcal{A}_1 + \sum_{q=2}^{Q} \mathcal{A}_q \\
&= \frac{[\widehat{Sp}_w(c_0) - \widehat{Sp}_w(c_1)] \cdot [\widehat{Se}_w(c_1) + \widehat{Se}_w(c_0)]}{2} \\
&\quad + \sum_{q=2}^{Q} \frac{[\widehat{Sp}_w(c_{q-1}) - \widehat{Sp}_w(c_q)] \cdot [\widehat{Se}_w(c_q) + \widehat{Se}_w(c_{q-1})]}{2} \\
&= \frac{1}{2} \sum_{q=1}^{Q} [\widehat{Sp}_w(c_{q-1}) - \widehat{Sp}_w(c_q)] \cdot [\widehat{Se}_w(c_q) + \widehat{Se}_w(c_{q-1})] \\
&= \frac{1}{2} \sum_{q=1}^{Q} [\widehat{Sp}_w(c_{q-1}) \cdot \widehat{Se}_w(c_q) - \widehat{Sp}_w(c_q) \cdot \widehat{Se}_w(c_{q-1})].
\end{aligned} \tag{6.26}$$

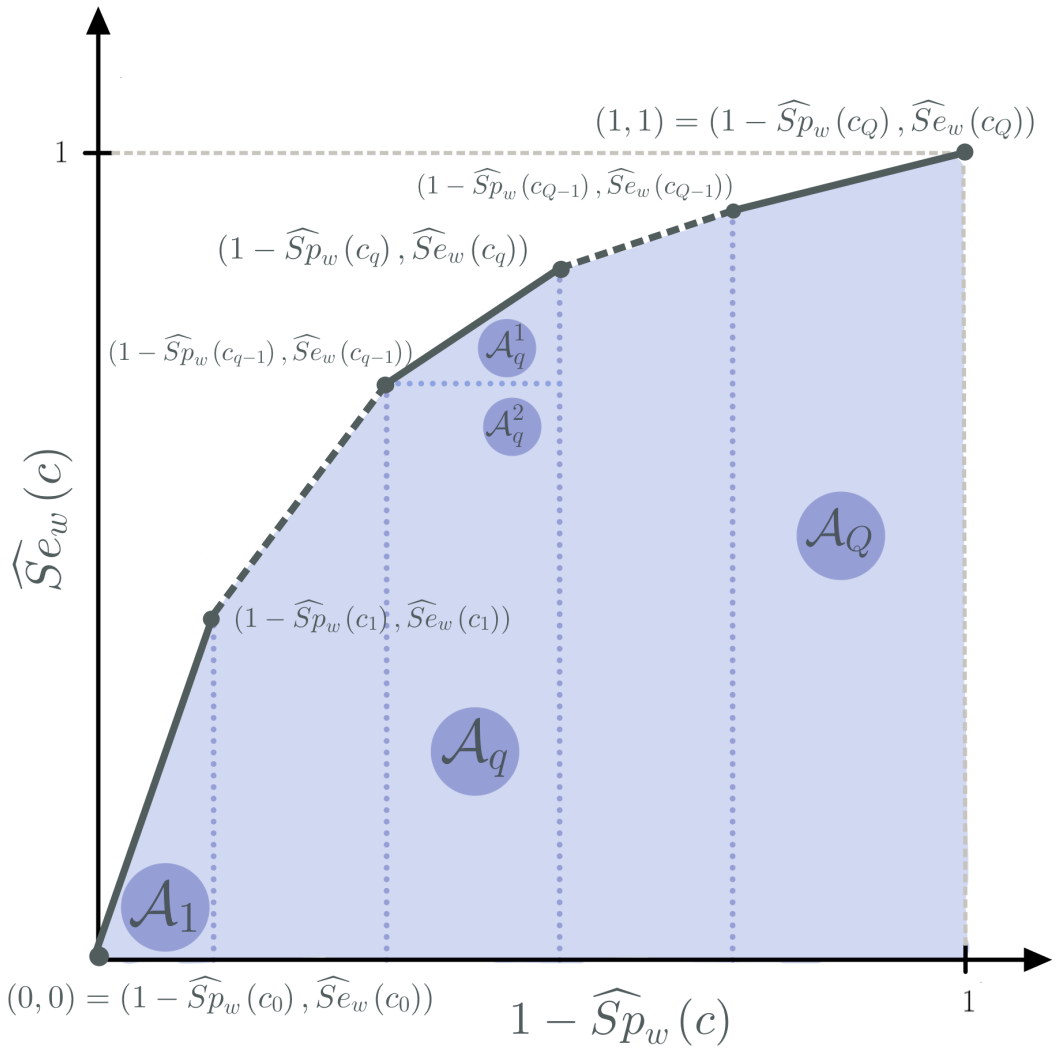


Figure 6.1: Graphical explanation of the weighted estimate of the ROC curve, where $c_Q < c_{Q-1} < \ldots < c_q < \ldots < c_1 < c_0$.

In the above explanations and graphical description, we have assumed the general case in which there may be ties (or, in other words, equal predicted probabilities) between units with and without the event of interest. It should be noted that the case in which there are no ties between the predicted probabilities of any of those pairs of units is a particular case of the general explanations we have previously made, and hence, all the equations described above are also valid for this particular situation.

Equivalence between the area under the $\widehat{ROC}_w(\cdot)$ curve and Mann-Whitney U-statistic

We propose to incorporate the marginal sampling weights into the Mann-Whitney U-Statistic as follows to estimate the weighted AUC:

$$\widehat{AUC}_w = \frac{\sum_{i_0 \in S_0} \sum_{i_1 \in S_1} w_{i_0} w_{i_1} \left[I(\hat{p}_{i_0} < \hat{p}_{i_1}) + 0.5 \cdot I(\hat{p}_{i_0} = \hat{p}_{i_1}) \right]}{\sum_{i_0 \in S_0} \sum_{i_1 \in S_1} w_{i_0} w_{i_1}}. \tag{6.27}$$

In the following lines, we show that the area under the estimated ROC curve $\mathcal{A}$ defined in eq. (6.26) is equivalent to the Mann-Whitney U-statistic considering marginal sampling weights as defined in eq. (6.27). In order to prove the equivalence between both approaches, our goal is to rewrite eq. (6.27) in terms of sensitivity and specificity parameters. Let us rewrite it as follows as the first step:

$$\widehat{AUC}_w = \frac{\sum_{i_0 \in S_0} \sum_{i_1 \in S_1} w_{i_0} w_{i_1} I(\hat{p}_{i_0} < \hat{p}_{i_1})}{\sum_{i_0 \in S_0} w_{i_0} \sum_{i_1 \in S_1} w_{i_1}} + \frac{1}{2} \cdot \frac{\sum_{i_0 \in S_0} \sum_{i_1 \in S_1} w_{i_0} w_{i_1} I(\hat{p}_{i_0} = \hat{p}_{i_1})}{\sum_{i_0 \in S_0} w_{i_0} \sum_{i_1 \in S_1} w_{i_1}}. \tag{6.28}$$

Then, we can rewrite the expressions $I(\hat{p}_{i_0} < \hat{p}_{i_1})$ and $I(\hat{p}_{i_0} = \hat{p}_{i_1})$ as a function of the previously defined cut-off points. Given that $c_Q < c_{Q-1} < \ldots < c_2 < c_1 < c_0$, let us denote:

$$I(c_q \leq \hat{p}_{i_1} < c_{q-1}) = I(\hat{p}_{i_1} \geq c_q) - I(\hat{p}_{i_1} \geq c_{q-1}),\ \forall q = 1, \ldots, Q. \tag{6.29}$$

Note that $\forall i_1 \in S_1, \exists! q \in \{1, \ldots, Q\} : \hat{p}_{i_1} \in [c_q,\ c_{q-1})$. Then, $\forall i_0 \in S_0$ the inequality $\hat{p}_{i_0} < \hat{p}_{i_1}$ will be satisfied if and only if $\hat{p}_{i_0} < c_q$, as graphically shown in Figure 6.2.

Thus, note that $I(\hat{p}_{i_0} < \hat{p}_{i_1})$ can be rewritten as follows. $\forall i_0 \in S_0$ and $\forall i_1 \in S_1$,

$$I(\hat{p}_{i_0} < \hat{p}_{i_1}) = \sum_{q=1}^{Q} I(\hat{p}_{i_0} < c_q) \cdot \left[I(\hat{p}_{i_1} \geq c_q) - I(\hat{p}_{i_1} \geq c_{q-1}) \right]. \tag{6.30}$$

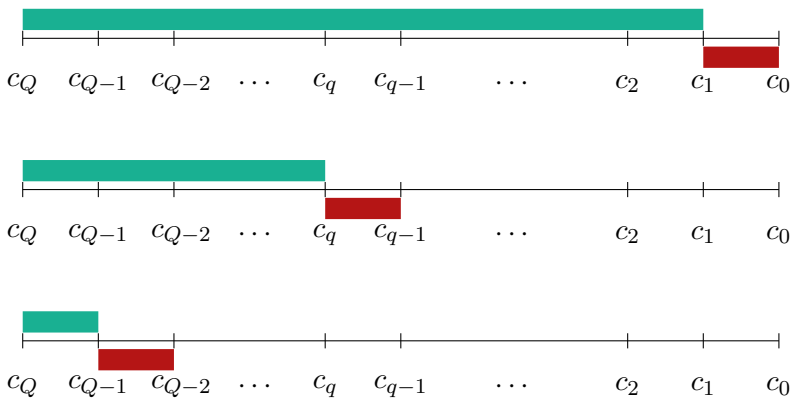


Figure 6.2: This image is intended to be helpful to better understand eq. (6.30) and indicates in which situations $I(\hat{p}_{i_0} < \hat{p}_{i_1}) = 1$. Given that $\forall i_1 \in S_1$, $\exists!\, q \in \{1, \ldots, Q-1\}$: $\hat{p}_{i_1} \in [c_q, c_{q-1})$ (in red), locations for $\hat{p}_{i_0}$ ($\forall i_0 \in S_0$) which satisfy $I(\hat{p}_{i_0} < \hat{p}_{i_1}) = 1$ are indicated in green ($\hat{p}_{i_0} < c_q$).

Then, following eq. (6.30) and the definitions given in eq. (6.19), let us rewrite the first term of eq. (6.28) in terms of sensitivity and specificity parameters as follows:

$$
\begin{aligned}
&\frac{\sum_{i_0 \in S_0} \sum_{i_1 \in S_1} w_{i_0} w_{i_1} I(\hat{p}_{i_0} < \hat{p}_{i_1})}{\sum_{i_0 \in S_0} w_{i_0} \sum_{i_1 \in S_1} w_{i_1}} \\
&= \frac{\sum_{i_0 \in S_0} \sum_{i_1 \in S_1} w_{i_0} w_{i_1} \sum_{q=1}^{Q} I(\hat{p}_{i_0} < c_q) \cdot [I(\hat{p}_{i_1} \geq c_q) - I(\hat{p}_{i_1} \geq c_{q-1})]}{\sum_{i_0 \in S_0} w_{i_0} \sum_{i_1 \in S_1} w_{i_1}} \\
&= \sum_{q=1}^{Q} \left\{ \frac{\sum_{i_0 \in S_0} w_{i_0} I(\hat{p}_{i_0} < c_q)}{\sum_{i_0 \in S_0} w_{i_0}} \cdot \frac{\sum_{i_1 \in S_1} w_{i_1} [I(\hat{p}_{i_1} \geq c_q) - I(\hat{p}_{i_1} \geq c_{q-1})]}{\sum_{i_1 \in S_1} w_{i_1}} \right\} \\
&= \sum_{q=1}^{Q} \widehat{Sp}_w(c_q) \cdot [\widehat{Se}_w(c_q) - \widehat{Se}_w(c_{q-1})].
\end{aligned}
\tag{6.31}
$$

In the same way, we will now proceed to rewrite the expression $I(\hat{p}_{i_0} = \hat{p}_{i_1})$. As stated above, $\forall i_1 \in S_1$, $\exists!\, q \in \{1, \ldots, Q\}$: $\hat{p}_{i_1} \in [c_q,\, c_{q-1})$. Thus, $\forall i_0 \in S_0$, the equality $\hat{p}_{i_0} = \hat{p}_{i_1}$ will only be satisfied if $\hat{p}_{i_0}$ is in the same range as $\hat{p}_{i_1}$, that is, $\hat{p}_{i_0} \in [c_q,\, c_{q-1})$ (see Figure 6.3).

For convenience, let us rewrite:

$$
I(c_q \leq \hat{p}_{i_0} < c_{q-1}) = I(\hat{p}_{i_0} < c_{q-1}) - I(\hat{p}_{i_0} < c_q). \tag{6.32}
$$

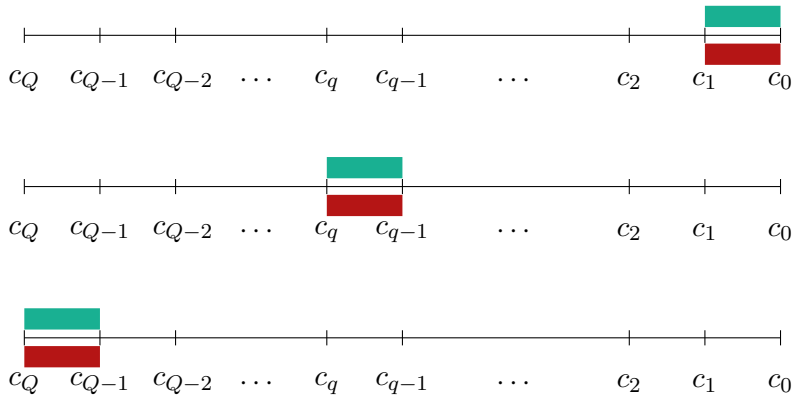


Figure 6.3: This image is intended to be helpful to better understand eq. (6.33) and indicates in which situations $I(\hat{p}_{i_0} = \hat{p}_{i_1}) = 1$. Given that $\forall i_1 \in S_1$, $\exists!\, q \in \{1, \dots, Q-1\}$: $\hat{p}_{i_1} \in [c_q, c_{q-1})$ (in red), locations for $\hat{p}_{i_0}$ ($\forall i_0 \in S_0$) which satisfy $I(\hat{p}_{i_0} = \hat{p}_{i_1}) = 1$ are indicated in green ($\hat{p}_{i_0} \in [c_q, c_{q-1})$).

Then, note that, $\forall i_0 \in S_0$, and $\forall i_1 \in S_1$:

$$I(\hat{p}_{i_0} = \hat{p}_{i_1}) = \sum_{q=1}^{Q} \left[I(\hat{p}_{i_0} < c_{q-1}) - I(\hat{p}_{i_0} < c_q) \right] \cdot \left[I(\hat{p}_{i_1} \geq c_q) - I(\hat{p}_{i_1} \geq c_{q-1}) \right]. \tag{6.33}$$

Following eq. (6.33) and the definitions given in eq. (6.19), let us rewrite the second term of eq. (6.28) in terms of sensitivity and specificity parameters as follows:

$$\begin{aligned}
&\frac{\sum_{i_0 \in S_0} \sum_{i_1 \in S_1} w_{i_0} w_{i_1} I(\hat{p}_{i_0} = \hat{p}_{i_1})}{\sum_{i_0 \in S_0} w_j \sum_{i_1 \in S_1} w_{i_1}} \\
&= \frac{\sum_{i_0 \in S_0} \sum_{i_1 \in S_1} w_{i_0} w_{i_1} \sum_{q=1}^{Q} \left[I(\hat{p}_{i_0} < c_{q-1}) - I(\hat{p}_{i_0} < c_q) \right] \cdot \left[I(\hat{p}_{i_1} \geq c_q) - I(\hat{p}_{i_1} \geq c_{q-1}) \right]}{\sum_{i_0 \in S_0} w_{i_0} \sum_{i_1 \in S_1} w_{i_1}} \\
&= \sum_{q=1}^{Q} \left\{ \frac{\sum_{i_0 \in S_0} w_{i_1} \left[I(\hat{p}_{i_0} < c_{q-1}) - I(\hat{p}_{i_0} < c_q) \right]}{\sum_{i_0 \in S_0} w_{i_0}} \cdot \frac{\sum_{i_1 \in S_1} w_{i_1} \left[I(\hat{p}_{i_1} \geq c_q) - I(\hat{p}_{i_1} \geq c_{q-1}) \right]}{\sum_{i_1 \in S_1} w_{i_1}} \right\} \\
&= \sum_{q=1}^{Q} [\widehat{Sp}_w(c_{q-1}) - \widehat{Sp}_w(c_q)] \cdot [\widehat{Se}_w(c_q) - \widehat{Se}_w(c_{q-1})].
\end{aligned} \tag{6.34}$$

Finally, eq. (6.28) can be rewritten as the sum of eqs. (6.31) and (6.34):

$$\begin{aligned}
\widehat{AUC}_w &= \sum_{q=1}^{Q} \widehat{Sp}_w(c_q) \cdot [\widehat{Se}_w(c_q) - \widehat{Se}_w(c_{q-1})] \\
&\quad + \frac{1}{2} \sum_{q=1}^{Q} [\widehat{Sp}_w(c_{q-1}) - \widehat{Sp}_w(c_q)] \cdot [\widehat{Se}_w(c_q) - \widehat{Se}_w(c_{q-1})] \\
&= \frac{1}{2} \sum_{q=1}^{Q} \Big\{ 2\widehat{Sp}_w(c_q)[\widehat{Se}_w(c_q) - \widehat{Se}_w(c_{q-1})] \\
&\quad + [\widehat{Sp}_w(c_{q-1}) - \widehat{Sp}_w(c_q)] \cdot [\widehat{Se}_w(c_q) - \widehat{Se}_w(c_{q-1})] \Big\} \\
&= \frac{1}{2} \sum_{q=1}^{Q} [\widehat{Sp}_w(c_{q-1}) + \widehat{Sp}_w(c_q)] \cdot [\widehat{Se}_w(c_q) - \widehat{Se}_w(c_{q-1})] \\
&= \frac{1}{2} \sum_{q=1}^{Q} [\widehat{Sp}_w(c_{q-1}) \cdot \widehat{Se}_w(c_q) - \widehat{Sp}_w(c_q) \cdot \widehat{Se}_w(c_{q-1})].
\end{aligned} \tag{6.35}$$

Note that eq. (6.26) and eq. (6.35) are equal, so we have shown that $\mathcal{A} = \widehat{AUC}_w$.

6.2.3 Estimation of the AUC with pairwise sampling weights

In Yao et al. (2015), the authors propose a weighted estimator for the AUC that considers pairwise sampling weights instead of marginal sampling weights (which we consider in our proposal as shown in eq. (6.27)). In order to better understand the differences between both approaches, let us first define the pairwise sampling weights $w_{i^*i^{**}}\, \forall i^*,\, i^{**} \in S$. Let us remind that, as introduced in Section 6.1, $\forall i^*, i^{**} \in S$, the pairwise sampling weight $w_{i^*i^{**}}$ is defined as the inverse joint inclusion probability, i.e., $w_{i^*i^{**}} = 1/\pi_{i^*i^{**}}$ where,

$$\pi_{i^*i^{**}} = P[(i^* \in S) \cap (i^{**} \in S)] = P(i^{**} \in S | i^* \in S) P(i^* \in S) = \pi_{i^{**}|i^*} \pi_{i^*}. \tag{6.36}$$

Taking this into account, in the following lines, we first define the pairwise sampling weights for both sampling designs considered in this study: the one-stage stratified sampling design and the two-stage stratified cluster sampling design. Then, we introduce the proposal made by Yao et al. (2015) for the estimation of the AUC.

Pairwise sampling weights in one-stage stratified sampling designs

Before defining the pairwise sampling weights for one-stage stratified sampling designs, let us briefly recall the basis of this sampling process (more information is given in Section 2.1.1). The finite population U is partitioned into H mutually excluding strata as explained in eq. (2.14), i.e., $U = \bigcup_{h=1}^{H} U_h$. $\forall h \in \{1, \ldots, H\}$, out of the N_h units in the population subset U_h, a total of n_h are randomly selected to be sampled and form the sample subset S_h corresponding to stratum h. Finally, $S = \bigcup_{h=1}^{H} S_h$.

In order to define the pairwise sampling weights in this context, let us consider two sampled units i^*, $i^{**} \in S$, each of them corresponding to a particular stratum h^*, $h^{**} \in \{1, \ldots, H\}$, i.e., $i^* \in S_{h^*}$ and $i^{**} \in S_{h^{**}}$. To calculate the pairwise sampling weight $w_{i^*i^{**}}$ for these units, we first need to calculate their joint inclusion probability $\pi_{i^*i^{**}}$ which is the product between the inclusion probability π_{i^*} and the conditional inclusion probability $\pi_{i^{**}|i^*}$, as previously defined in eq. (6.36).

The marginal inclusion probability π_{i^*}, $\forall i^* \in S_{h^*} \subset U_{h^*}$ is shown in eq. (6.37) (defined in eq. (2.3)):

$$\pi_{i^*} = \frac{n_h}{N_h}, \quad \forall i^* \in S_{h^*} \subset U_{h^*}. \tag{6.37}$$

In contrast, in order to properly define the conditional probability $\pi_{i^{**}|i^*}$, two situations need to be distinguished:

(i) If both units $i^* \in S_{h^*}$ and $i^{**} \in S_{h^{**}}$ are from different strata (i.e., $h^* \neq h^{**}$), then, $\pi_{i^{**}|i^*}$ is equal to the marginal inclusion probability $\pi_{i^{**}}$ as defined in eq. (6.38):

$$\pi_{i^{**}|i^*} = \pi_{i^{**}} = \frac{n_{h^{**}}}{N_{h^{**}}}. \tag{6.38}$$

(ii) If both units $i^* \in S_{h^*}$ and $i^{**} \in S_{h^{**}}$ are from the same stratum (i.e., $h^* = h^{**}$), then, the conditional and marginal inclusion probabilities differ. The conditional probability $\pi_{i^{**}|i^*}$ is then defined as in eq. (6.39):

$$\pi_{i^{**}|i^*} = \frac{n_{h^*} - 1}{N_{h^*} - 1}. \tag{6.39}$$

Then, the joint inclusion probability $\pi_{i^*i^{**}}$ is defined in eq. (6.40):

$$\pi_{i^*i^{**}} = \pi_{i^{**}|i^*}\pi_{i^*} \begin{cases} \dfrac{n_{h^{**}}}{N_{h^{**}}} \cdot \dfrac{n_{h^*}}{N_{h^*}}, & \text{if } h^* \neq h^{**}, \\ \dfrac{n_{h^*}-1}{N_{h^*}-1} \cdot \dfrac{n_{h^*}}{N_{h^*}}, & \text{if } h^* = h^{**}. \end{cases} \tag{6.40}$$

Finally, the pairwise sampling weights are defined as in eq. (6.41), $\forall i^* \in S_{h^*}$ and $\forall i^{**} \in S_{h^{**}}$:

$$w_{i^*i^{**}} = \frac{1}{\pi_{i^*i^{**}}} = \begin{cases} \dfrac{N_{h^*}}{n_{h^*}} \cdot \dfrac{N_{h^{**}}}{n_{h^{**}}} = w_{i^*}w_{i^{**}}, & \text{if } h^* \neq h^{**}, \\ \dfrac{N_{h^*}}{n_{h^*}} \cdot \dfrac{(N_{h^*}-1)}{(n_{h^*}-1)}, & \text{if } h^* = h^{**}. \end{cases} \tag{6.41}$$

Pairwise sampling weights in two-stage stratified cluster sampling designs

In order to define the pairwise sampling weights for two-stage stratified cluster samples, we first summarize this sampling process (defined in detail in Section 2.1.2). In this case, the finite population U is partitioned into H strata, which at the same time, are partitioned into A_h clusters, i.e., $U = \bigcup_{h=1}^{H} \bigcup_{\alpha=1}^{A_h} U_{h,\alpha}$. In the first stage of the sampling process, a_h clusters are randomly selected out of the A_h clusters in the population stratum U_h, $\forall h \in \{1, \ldots, H\}$. We denote as $\mathbb{A}_h$ the set of indexes of the clusters selected from stratum h in the first stage of the sampling, as shown in eq. (6.42) (defined in eq. (2.24)):

$$\mathbb{A}_h = \{\alpha \in \{1, \ldots, A_h\} \,:\, 1_h(\alpha) = 1\}, \quad \forall h \in \{1, \ldots, H\}, \tag{6.42}$$

where, $1_h(\alpha)$ is the indicator function that takes the value 1 if $U_{h,\alpha}$ is selected in the first stage and 0 otherwise. In the second stage of the sampling process, $\forall h \in \{1, \ldots, H\}$ and $\forall \dot{\alpha} \in \mathbb{A}_h$, out of $N_{h,\dot{\alpha}}$ units in $U_{h,\dot{\alpha}}$, $n_{h,\dot{\alpha}}$ are sampled and form $S_{h,\dot{\alpha}}$. Finally, $S = \bigcup_{h=1}^{H} \bigcup_{\dot{\alpha} \in \mathbb{A}_h} S_{h,\dot{\alpha}}$.

Let us consider two sampled units, $i^* \in S_{h^*,\alpha'}$ and $i^{**} \in S_{h^{**},\alpha''}$, where h^*, $h^{**} \in \{1, \ldots, H\}$, $\alpha' \in \mathbb{A}_{h^*}$ and $\alpha'' \in \mathbb{A}_{h^{**}}$. In order to define the pairwise sampling weight $w_{i^*i^{**}}$, let us first calculate the joint inclusion probability $\pi_{i^*i^{**}} = \pi_{i^{**}|i^*}\pi_{i^*}$ (see eq. (6.36)). The marginal inclusion probability π_{i^*} is shown in eq. (6.43) (previously

defined in eq. (2.20)):

$$\pi_{i^*} = \frac{n_{h^*,\alpha'}}{N_{h^*,\alpha'}} \cdot \frac{a_{h^*}}{A_{h^*}}, \quad \forall i^* \in S_{h^*,\alpha'} \subset U_{h^*,\alpha'}. \tag{6.43}$$

However, in order to properly define the conditional inclusion probability $\pi_{i^{**}|i^*}$ in this context, we need to distinguish three different situations:

(i) If both units, $i^* \in S_{h^*,\alpha'}$ and $i^{**} \in S_{h^{**},\alpha''}$, are from different strata (i.e., $h^* \neq h^{**}$) the conditional inclusion probability is equal to the marginal one as shown in eq. (6.44):

$$\pi_{i^{**}|i^*} = \pi_{i^{**}} = \frac{n_{h^{**},\alpha''}}{N_{h^{**},\alpha''}} \cdot \frac{a_{h^{**}}}{A_{h^{**}}}. \tag{6.44}$$

(ii) If the sampled units $i^* \in S_{h^*,\alpha'}$ and $i^{**} \in S_{h^{**},\alpha''}$ are from the same stratum but different clusters, (i.e., $h^* = h^{**}$ and $\alpha' \neq \alpha''$) the conditional inclusion probability can be defined as in eq. (6.45):

$$\pi_{i^{**}|i^*} = \frac{n_{h^*,\alpha''}}{N_{h^*,\alpha''}} \cdot \frac{a_{h^*} - 1}{A_{h^*} - 1}. \tag{6.45}$$

(iii) If the sampled units $i^* \in S_{h^*,\alpha'}$ and $i^{**} \in S_{h^{**},\alpha''}$ are sampled from the same cluster (i.e., $h^* = h^{**}$ and $\alpha' = \alpha''$), then, the expression for the conditional probability is the one given in eq. (6.46):

$$\pi_{i^{**}|i^*} = \frac{n_{h^*,\alpha''} - 1}{N_{h^*,\alpha''} - 1} \cdot 1. \tag{6.46}$$

Then the joint inclusion probability for any pair of units $i^* \in S_{h^*,\alpha'}$ and $i^{**} \in S_{h^{**},\alpha''}$, is defined as in eq. (6.47):

$$\pi_{i^*i^{**}} = \begin{cases} \left(\frac{a_{h^*}}{A_{h^*}} \cdot \frac{n_{h^*,\alpha'}}{N_{h^*,\alpha'}}\right) \cdot \left(\frac{a_{h^{**}}}{A_{h^{**}}} \cdot \frac{n_{h^{**},\alpha''}}{N_{h^{**},\alpha''}}\right), & \text{if } h^* \neq h^{**}, \\ \left(\frac{a_{h^*}}{A_{h^*}} \cdot \frac{n_{h^*,\alpha'}}{N_{h^*,\alpha'}}\right) \cdot \left(\frac{(a_{h^*} - 1)}{(A_{h^*} - 1)} \cdot \frac{n_{h^*,\alpha''}}{N_{h^*,\alpha''}}\right), & \text{if } h^* = h^{**} \text{ but } \alpha' \neq \alpha'', \\ \left(\frac{a_{h^*}}{A_{h^*}} \cdot \frac{n_{h^*,\alpha'}}{N_{h^*,\alpha'}}\right) \cdot \frac{(n_{h^*,\alpha'} - 1)}{(N_{h^*,\alpha'} - 1)}, & \text{if } h^* = h^{**} \text{ and } \alpha' = \alpha'', \end{cases} \tag{6.47}$$

from which the definition of the pairwise sampling weights is defined in eq. (6.48),

taking into account that $w_{i^*i^{**}} = 1/\pi_{i^*i^{**}}$:

$$w_{i^*i^{**}} = \begin{cases} \left(\frac{A_{h^*}}{a_{h^*}} \cdot \frac{N_{h^*,\alpha'}}{n_{h^*,\alpha'}}\right) \cdot \left(\frac{A_{h^{**}}}{a_{h^{**}}} \cdot \frac{N_{h^{**},\alpha''}}{n_{h^{**},\alpha''}}\right) = w_{i^*} w_{i^{**}}, & \text{if } h^* \neq h^{**}, \\ \left(\frac{A_{h^*}}{a_{h^*}} \cdot \frac{N_{h^*,\alpha'}}{n_{h^*,\alpha'}}\right) \cdot \left(\frac{(A_{h^*}-1)}{(a_{h^*}-1)} \cdot \frac{N_{h^*,\alpha''}}{n_{h^*,\alpha''}}\right), & \text{if } h^* = h^{**} \text{ but } \alpha' \neq \alpha'', \\ \left(\frac{A_{h^*}}{a_{h^*}} \cdot \frac{N_{h^*,\alpha'}}{n_{h^*,\alpha'}}\right) \cdot \frac{(N_{h^*,\alpha'}-1)}{(n_{h^*,\alpha'}-1)}, & \text{if } h^* = h^{**} \text{ and } \alpha' = \alpha''. \end{cases} \tag{6.48}$$

AUC estimator approach with pairwise sampling weights

The estimator proposed in Yao et al. (2015), is defined in eq. (6.49), i.e.,

$$\widehat{AUC}_{\text{pairw}} = \frac{1}{\sum_{i_0 \in S_0} w_{i_0} \cdot \sum_{i_1 \in S_1} w_{i_1}} \sum_{i_0 \in S_0} \sum_{i_1 \in S_1} w_{i_0 i_1} \left[I(\hat{p}_{i_0} < \hat{p}_{i_1}) + 0.5 I(\hat{p}_{i_0} = \hat{p}_{i_1}) \right], \tag{6.49}$$

where $w_{i_0 i_1} = 1/\pi_{i_0 i_1}$ and $\pi_{i_0 i_1}$ indicates the joint inclusion probability for $i_0 \in S_0$ and $i_1 \in S_1$ for both of them being sampled, as defined in eq. (6.36). Given that $w_{i_0 i_1} \neq w_{i_0} w_{i_1}$ (unless for a pair of units, being each of them in two different strata), the proposal that considers the pairwise sampling weights (eq. (6.49)) and our proposal that considers the marginal sampling weights (eq. (6.27)) differ.

AUC estimator in matrix form: pairwise or marginal weights?

Yao et al. (2015) state that their proposal, defined in eq. (6.49), can be summarized in matrix form as follows for one-stage stratified sampling designs:

$$(\hat{\boldsymbol{p}}_{(H)}^{Y=0})^T \cdot \widehat{\Theta}_{(H \times H)} \cdot \hat{\boldsymbol{p}}_{(H)}^{Y=1}, \tag{6.50}$$

where, $\hat{\boldsymbol{p}}_{(H)}^{Y=0}$ and $\hat{\boldsymbol{p}}_{(H)}^{Y=1}$ are two vectors of dimension H defined as in eq. (6.51), respectively:

$$\begin{aligned} \hat{\boldsymbol{p}}_{(H)}^{Y=0} &= \left(\frac{\hat{N}_{0,1}}{\hat{N}_0}, \ldots, \frac{\hat{N}_{0,h}}{\hat{N}_0}, \ldots, \frac{\hat{N}_{0,H}}{\hat{N}_0} \right)^T, \\ \hat{\boldsymbol{p}}_{(H)}^{Y=1} &= \left(\frac{\hat{N}_{1,1}}{\hat{N}_1}, \ldots, \frac{\hat{N}_{1,h}}{\hat{N}_1}, \ldots, \frac{\hat{N}_{1,H}}{\hat{N}_1} \right)^T, \end{aligned} \tag{6.51}$$

being $\hat{N}_0 = \hat{N}_{Y=0}$ and $\hat{N}_1 = \hat{N}_{Y=1}$ the ones previously defined in eq. (3.4) and we set again in eq. (6.52) (given that we consider only one response variable, Y, we suppress it from the subindices for ease of notation), i.e.,

$$\begin{aligned} \hat{N}_0 &= \sum_{i \in S} w_i \cdot I(y_i = 0) = \sum_{i_0 \in S_0} w_{i_0}, \\ \hat{N}_1 &= \sum_{i \in S} w_i \cdot I(y_i = 1) = \sum_{i_1 \in S_1} w_{i_1}, \end{aligned} \tag{6.52}$$

and $\forall h \in \{1, \ldots, H\}$, $\hat{N}_{0,h}$ and $\hat{N}_{1,h}$ are defined in eq. (6.53):

$$\begin{aligned} \hat{N}_{0,h} &= \sum_{i \in S_h} w_i \cdot I(y_i = 0), \\ \hat{N}_{1,h} &= \sum_{i \in S_h} w_i \cdot I(y_i = 1), \end{aligned} \tag{6.53}$$

or equivalently, if we consider the sample subsets of units without and with the event of interest (i.e., S_0 and S_1, respectively) and given that those subsets can be separated into disjoint groups depending on the strata as $S_0 = \bigcup_{h=1}^{H} S_{0,h}$ and $S_1 = \bigcup_{h=1}^{H} S_{1,h}$, then note that eq. (6.53) can be rewritten as follows:

$$\begin{aligned} \hat{N}_{0,h} &= \sum_{i \in S_{0,h}} w_{i_{0,h}}, \\ \hat{N}_{1,h} &= \sum_{i \in S_{1,h}} w_{i_{1,h}}. \end{aligned} \tag{6.54}$$

In addition, if we take into account that $\forall h \in \{1, \ldots, H\}$, all the units in S_h have the same sampling weight, which is defined as $w_i = \dfrac{N_h}{n_h}$, $\forall i \in S_h$, as explained in (2.6), then let us rewrite eq. (6.54) as follows for convenience:

$$\begin{aligned} \hat{N}_{0,h} &= \sum_{i_{0,h} \in S_{0,h}} w_{i_{0,h}} = n_{0,h} \cdot \frac{N_h}{n_h}, \\ \hat{N}_{1,h} &= \sum_{i_{1,h} \in S_{1,h}} w_{i_{1,h}} = n_{1,h} \cdot \frac{N_h}{n_h}. \end{aligned} \tag{6.55}$$

where $n_{0,h}$ and $n_{1,h}$ indicate the size of $S_{0,h}$ and $S_{1,h}$, $\forall h \in \{1,\ldots,H\}$, respectively. Finally, $\widehat{\Theta}_{(H\times H)}$ is defined as in eq. (6.56):

$$\widehat{\Theta}_{(H\times H)} = \begin{pmatrix} \widehat{\theta}(1,1) & \cdots & \widehat{\theta}(1,h) & \cdots & \widehat{\theta}(1,H) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \widehat{\theta}(h,1) & \cdots & \widehat{\theta}(h,h) & \cdots & \widehat{\theta}(h,H) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \widehat{\theta}(H,1) & \cdots & \widehat{\theta}(H,h) & \cdots & \widehat{\theta}(H,H) \end{pmatrix}, \tag{6.56}$$

where $\forall\ h^*,\ h^{**} \in \{1,\ldots,H\}$, $\widehat{\theta}(h^*,h^{**})$ indicates the unweighted estimate of the AUC considering all the possible pairs between the non-events in stratum h^* and the events in h^{**}, i.e.,

$$\widehat{\theta}(h^*,h^{**}) = \frac{1}{n_{0,h^*}\cdot n_{1,h^{**}}} \sum_{i_{0,h^*}\in S_{0,h^*}} \sum_{i_{1,h^{**}}\in S_{1,h^{**}}} \psi(\hat{p}_{i_{0,h^*}},\hat{p}_{i_{1,h^{**}}}), \tag{6.57}$$

where, $\forall i_{0,h^*} \in S_{0,h^*}$ and $\forall i_{1,h^{**}} \in S_{1,h^{**}}$,

$$\psi(\hat{p}_{i_{0,h^*}},\hat{p}_{i_{1,h^{**}}}) = \left[I(\hat{p}_{i_{0,h^*}} < \hat{p}_{i_{1,h^{**}}}) + 0.5I(\hat{p}_{i_{0,h^*}} = \hat{p}_{i_{1,h^{**}}})\right], \tag{6.58}$$

and n_{0,h^*} and $n_{1,h^{**}}$ indicate the number of units in the subsets S_{0,h^*} and $S_{1,h^{**}}$, respectively. We use indicators i_{0,h^*} and $i_{1,h^{**}}$ to easily differentiate between units without and with the event of interest and the stratum they belong to.

However, it can be proven that eq. (6.50) is not equivalent to the $\widehat{AUC}_{\text{pairw}}$ defined in eq. (6.49) (i.e., Yao's proposal based on pairwise sampling weights). For this purpose, we only need to develop the matrix calculations of eq. (6.50) considering the definitions given in eqs. (6.51) and (6.57), i.e.,

$$\begin{aligned} (\hat{\boldsymbol{p}}_{(H)}^{Y=0})^T \cdot \widehat{\Theta}_{(H\times H)} \cdot \hat{\boldsymbol{p}}_{(H)}^{Y=1} &= \sum_{h^*=1}^{H}\sum_{h^{**}=1}^{H} \frac{\hat{N}_{0,h^*}}{\hat{N}_0}\cdot\hat{\theta}(h^*,h^{**})\cdot\frac{\hat{N}_{1,h^{**}}}{\hat{N}_1} \\ &= \frac{1}{\hat{N}_0\cdot\hat{N}_1}\sum_{h^*=1}^{H}\sum_{h^{**}=1}^{H}\hat{N}_{0,h^*}\hat{N}_{1,h^{**}}\,\hat{\theta}(h^*,h^{**}) \end{aligned} \tag{6.59}$$

Taking into account the equalities given in eq. (6.55) and the definition of $\widehat{\theta}(h^*,h^{**})$

in eq. (6.57), we can continue as follows:

$$
\begin{aligned}
&\frac{1}{\hat{N}_0 \cdot \hat{N}_1} \sum_{h^*=1}^{H} \sum_{h^{**}=1}^{H} \hat{N}_{0,h^*} \hat{N}_{1,h^{**}} \, \hat{\theta}(h^*, h^{**}) \\
&= \frac{1}{\hat{N}_0 \cdot \hat{N}_1} \sum_{h^*=1}^{H} \sum_{h^{**}=1}^{H} n_{0,h^*} \frac{N_{h^*}}{n_{h^*}} n_{1,h^{**}} \frac{N_{h^{**}}}{n_{h^{**}}} \, \hat{\theta}(h^*, h^{**}) \\
&= \frac{1}{\hat{N}_0 \cdot \hat{N}_1} \sum_{h^*=1}^{H} \sum_{h^{**}=1}^{H} n_{0,h^*} \frac{N_{h^*}}{n_{h^*}} n_{1,h^{**}} \frac{N_{h^{**}}}{n_{h^{**}}} \cdot \{ \\
&\quad \frac{1}{n_{0,h^*} \cdot n_{1,h^{**}}} \sum_{i_{0,h^*} \in S_{0,h^*}} \sum_{i_{1,h^{**}} \in S_{1,h^{**}}} \psi(\hat{p}_{i_{0,h^*}}, \hat{p}_{i_{1,h^{**}}}) \, \} \\
&= \frac{1}{\hat{N}_0 \cdot \hat{N}_1} \sum_{h^*=1}^{H} \sum_{h^{**}=1}^{H} \frac{N_{h^*}}{n_{h^*}} \cdot \frac{N_{h^{**}}}{n_{h^{**}}} \sum_{i_{0,h^*} \in S_{0,h^*}} \sum_{i_{1,h^{**}} \in S_{1,h^{**}}} \psi(\hat{p}_{i_{0,h^*}}, \hat{p}_{i_{1,h^{**}}}) \\
&= \frac{1}{\hat{N}_0 \cdot \hat{N}_1} \sum_{h^*=1}^{H} \sum_{h^{**}=1}^{H} \sum_{i_{0,h^*} \in S_{0,h^*}} \sum_{i_{1,h^{**}} \in S_{1,h^{**}}} \frac{N_{h^*}}{n_{h^*}} \cdot \frac{N_{h^{**}}}{n_{h^{**}}} \cdot \psi(\hat{p}_{i_{0,h^*}}, \hat{p}_{i_{1,h^{**}}})
\end{aligned}
\tag{6.60}
$$

Given that $\forall i \in S_h,\ w_i = \dfrac{N_h}{n_h}$ as defined in eq. (2.6), note that,

$$
\begin{aligned}
w_{i_{0,h^*}} &= \frac{N_{h^*}}{n_{h^*}}, \quad \forall i_{0,h^*} \in S_{0,h^*} \subset S_{h^*}, \\
w_{i_{1,h^{**}}} &= \frac{N_{h^{**}}}{n_{h^{**}}}, \quad \forall i_{1,h^{**}} \in S_{1,h^{**}} \subset S_{h^{**}}.
\end{aligned}
\tag{6.61}
$$

Then, we have that:

$$
\begin{aligned}
&\frac{1}{\hat{N}_0 \cdot \hat{N}_1} \sum_{h^*=1}^{H} \sum_{h^{**}=1}^{H} \sum_{i_{0,h^*} \in S_{0,h^*}} \sum_{i_{1,h^{**}} \in S_{1,h^{**}}} \frac{N_{h^*}}{n_{h^*}} \cdot \frac{N_{h^{**}}}{n_{h^{**}}} \cdot \psi(\hat{p}_{i_{0,h^*}}, \hat{p}_{i_{1,h^{**}}}) \\
&= \frac{1}{\hat{N}_0 \cdot \hat{N}_1} \sum_{h^*=1}^{H} \sum_{h^{**}=1}^{H} \sum_{i_{0,h^*} \in S_{0,h^*}} \sum_{i_{1,h^{**}} \in S_{1,h^{**}}} w_{i_{0,h^*}} w_{i_{1,h^{**}}} \, \psi(\hat{p}_{i_{0,h^*}}, \hat{p}_{i_{1,h^{**}}}).
\end{aligned}
\tag{6.62}
$$

In addition, taking into account that, as we consider all the strata, we can summarize the four sum operators in two, considering in this way the whole subsamples of non-

events (S_0) and events (S_1) instead of the disjoint subgroups separately for each stratum, i.e.,

$$\begin{aligned}&\frac{1}{\hat{N}_0 \cdot \hat{N}_1} \sum_{h^*=1}^{H} \sum_{h^{**}=1}^{H} \sum_{i_{0,h^*} \in S_{0,h^*}} \sum_{i_{1,h^{**}} \in S_{1,h^{**}}} w_{i_{0,h^*}} w_{i_{1,h^{**}}} \, \psi(\hat{p}_{i_{0,h^*}}, \hat{p}_{i_{1,h^{**}}}) \\ &= \frac{1}{\hat{N}_0 \cdot \hat{N}_1} \sum_{i_0 \in S_0} \sum_{i_1 \in S_1} w_{i_0} w_{i_1} \, \psi(\hat{p}_{i_0}, \hat{p}_{i_1}).\end{aligned} \tag{6.63}$$

Finally, regarding eq. (6.58) and rewriting $\hat{N}_0$ and $\hat{N}_1$ by considering eq. (6.52),

$$\begin{aligned}&\frac{1}{\hat{N}_0 \cdot \hat{N}_1} \sum_{i_0 \in S_0} \sum_{i_1 \in S_1} w_{i_0} w_{i_1} \, \psi(\hat{p}_{i_0}, \hat{p}_{i_1}) \\ &= \frac{1}{\sum_{i_0 \in S_0} w_{i_0} \sum_{i_1 \in S_1} w_{i_1}} \sum_{i_0 \in S_0} \sum_{i_1 \in S_1} w_{i_0} w_{i_1} \left[I(\hat{p}_{i_0} < \hat{p}_{i_1}) + 0.5 I(\hat{p}_{i_0} = \hat{p}_{i_1}) \right] \\ &= \frac{\sum_{i_0 \in S_0} \sum_{i_1 \in S_1} w_{i_0} w_{i_1} \left[I(\hat{p}_{i_0} < \hat{p}_{i_1}) + 0.5 I(\hat{p}_{i_0} = \hat{p}_{i_1}) \right]}{\sum_{i_0 \in S_0} \sum_{i_1 \in S_1} w_{i_0} w_{i_1}} = \widehat{AUC}_w.\end{aligned} \tag{6.64}$$

Therefore, beginning in eq. (6.59) and finishing in eq. (6.64), we have shown that,

$$(\hat{\boldsymbol{p}}_{(H)}^{Y=0})^T \cdot \widehat{\Theta}_{(H \times H)} \cdot \hat{\boldsymbol{p}}_{(H)}^{Y=1} = \widehat{AUC}_w. \tag{6.65}$$

All the definitions given above can also be extended to the two-stage stratified cluster samples, and hence, the $\widehat{AUC}_w$ can also be summarized in matrix form when this complex sampling design is considered.

6.3 Simulation study

The goal of this simulation study is to analyze the performance of the proposed estimators in comparison to the traditional unweighted estimators of the ROC curve and AUC. In Section 6.3.1, we describe the data generation process and different scenarios considered throughout the study. In Section 6.3.2, we describe the simulation set-up. In Section 6.3.3, we summarize the main results. Finally, in Section 6.3.4, we extend the simulation study to analyze the performance of the proposed estimators under uninformative sampling designs.

6.3.1 Data generation and scenarios

The data simulation process described below is similar to the one described in Section 5.3.1. Let us define as $N = 10\,000$ the finite population size. A set of $p = 5$ covariates $(X_1, \ldots, X_5)$ and two latent variables (Z_1 and Z_2, which are used to define the response variable and the sampling design, but are not available in the samples when fitting models) have been generated.

A total of three different scenarios have been defined based on different sampling designs. On the one hand, a one-stage stratified sampling design was defined (let us denote this scenario as SH, hereinafter), in which different strata are defined in the finite population, and a number of individuals are sampled from each stratum. On the other hand, we defined a two-stage stratified cluster sampling design (scenario SC), in which different strata are defined in the finite population, a number of clusters or groups of units are selected from each stratum, and finally, a number of individuals are sampled from each selected cluster. In addition, in this scenario SC, two situations have been distinguished: first, all the variables have been considered as unit-level variables (we denote this scenario as SC.0, given that there are $d = 0$ cluster-level variables), and second, in the other scenario, one cluster-level variable ($d = 1$) has been considered (scenario SC.1). Note that in scenario SH, all the variables must be defined at unit-level ($d = 0$) since there is no cluster. We proceed below to explain the data generation process for each of these scenarios:

1. For $d = 0$ (SH and SC.0) and $d = 1$ (SC.1), N realizations have been made following the Gaussian distribution defined in eq. (6.66):

$$(X_{d+1}, \ldots, X_5, Z_1, Z_2) \sim N(\boldsymbol{\mu}_{(p-d)}, \Sigma_{(p-d)\times(p-d)}), \tag{6.66}$$

   where $\boldsymbol{\mu}_{(p-d)}$ indicates the null vector of dimension $1 \times (p-d)$ and $\Sigma_{(p-d)\times(p-d)}$ a matrix of dimension $(p-d) \times (p-d)$ defined by values of 1 on the diagonal and $\eta = 0.15$ off-diagonal, i.e.,

$$\boldsymbol{\mu}_{(p-d)} = (0, \ldots, 0)^T \quad \text{and} \quad \Sigma_{(p-d)\times(p-d)} = (1-\eta)\cdot I_{(p-d)\times(p-d)} + \eta \cdot J_{(p-d)\times(p-d)}, \tag{6.67}$$

   being $I_{(p-d)\times(p-d)}$ the identity matrix and $J_{(p-d)\times(p-d)}$ the matrix of 1s.

2. Let us denote as $\{\boldsymbol{z}_i = (z_{i,1}, z_{i,2})\}_{i=1}^N$ the set of N realizations of Z_1 and Z_2.

Data is sort based on $\boldsymbol{z}_i\boldsymbol{\beta}^{\boldsymbol{Z}}$, $\forall i = 1, \ldots, N$, where:

$$\boldsymbol{\beta}^{\boldsymbol{Z}} = (\beta_1^{Z_1}, \beta_2^{Z_2})^T = (-3.5,\ -3.5)^T. \tag{6.68}$$

Strata are defined by partitioning the ordered population data set on sets of the same size ($H = 10$ strata) in all the scenarios, being each stratum of size $N_h = 1000$, $\forall h = 1, \ldots, H$. In addition, in scenarios SC.0 and SC.1, each stratum has been partitioned into $A_h = 10$ clusters $\forall h = 1, \ldots, H$. In this way, a total of $A = 100$ clusters of size $N_{h,\alpha} = 100$ are generated, $\forall h = 1, \ldots, H$ and $\forall \alpha = 1, \ldots A_h$.

3. If $d = 1$, then X_1 is a cluster-level variable (SC.1). We generate it by making $A = \sum_{h=1}^{H} A_h$ realizations of $X_1 \sim N(0, 1)$. Note that for two different units in the same cluster, their corresponding cluster-level covariates should take the same value, i.e., $\forall i, j$ in the same cluster, $x_{i,1} = x_{j,1}$. Therefore, we repeat each realization $N_{h,\alpha}$ times, $\forall \alpha = 1, \ldots, A_h$, $\forall h = 1, \ldots, H$.

4. We now have defined the values corresponding to $X_1, \ldots, X_5$ variables for all the units in the finite population: $\{\boldsymbol{x}_i = (x_{i,1}, \ldots, x_{i,5})\}_{i=1}^{N}$. Let us define $\boldsymbol{\beta}^{\boldsymbol{X}}$ as follows:

$$\boldsymbol{\beta}^{\boldsymbol{X}} = (\beta_1^{X_1}, \ldots, \beta_5^{X_5})^T = (2.5, \ldots, 2.5)^T. \tag{6.69}$$

Then, we generate the probabilities of event as follows:

$$p(\boldsymbol{x}_i, \boldsymbol{z}_i) = \frac{e^{\beta_0 + \boldsymbol{x}_i\boldsymbol{\beta}^{\boldsymbol{X}} + \boldsymbol{z}_i\boldsymbol{\beta}^{\boldsymbol{Z}}}}{1 + e^{\beta_0 + \boldsymbol{x}_i\boldsymbol{\beta}^{\boldsymbol{X}} + \boldsymbol{z}_i\boldsymbol{\beta}^{\boldsymbol{Z}}}}, \quad \forall i = 1, \ldots, N, \tag{6.70}$$

and the value for the response variable y_i is randomly generated by following $Bernoulli(p(\boldsymbol{x}_i, \boldsymbol{z}_i))$, i.e.,

$$y_i \sim Bernoulli(p(\boldsymbol{x}_i, \boldsymbol{z}_i)). \tag{6.71}$$

We set $\beta_0 = -5$, defining in this way a probability of event of around 25%.

The finite population U is defined as the set of values corresponding to the response variable y_i and the covariates $\boldsymbol{x}_i$, $\forall i = 1, \ldots, N$ (excluding the latent variables $\boldsymbol{z}_i$), as well as strata and cluster indicators corresponding to each of them.

5. Different sampling schemes have been considered in this simulation study. On

the one hand, in the scenario in which a one-stage stratified sampling design is defined (SH), the following number of units have been sampled from each stratum ($n_h, \forall h = 1, \dots H$):

$$\textbf{SH (a)} \quad n_1 = n_{10} = 150,\ n_2 = n_9 = 100,\ n_3 = n_8 = 50,\ n_4 = n_7 = 40,\ n_5 = n_6 = 30,$$

$$\textbf{SH (b)} \quad n_1 = n_{10} = 30,\ n_2 = n_9 = 40,\ n_3 = n_8 = 50,\ n_4 = n_7 = 100,\ n_5 = n_6 = 150.$$

On the other hand, in the scenarios in which a two-stage stratified cluster sampling design is considered (SC.0 and SC.1), $a_h = 2$, $\forall h = 1, \dots, H$ clusters have been selected in the first stage. Then, a different number of units is sampled from each selected cluster of each stratum ($n_{h,\dot{\alpha}}$). In particular, $\forall h \in \{1, \dots, H\}$ and $\forall \dot{\alpha} \in \mathbb{A}_h$ the following number of units have been sampled in each scenario:

$$\textbf{SC.0 (a), SC.1 (a)} \quad n_{1,\dot{\alpha}} = n_{10,\dot{\alpha}} = 75,\ n_{2,\dot{\alpha}} = n_{9,\dot{\alpha}} = 50,\ n_{3,\dot{\alpha}} = n_{8,\dot{\alpha}} = 25,\ n_{4,\dot{\alpha}} = n_{7,\dot{\alpha}} = 20,\ n_{5,\dot{\alpha}} = n_{6,\dot{\alpha}} = 15,$$

$$\textbf{SC.0 (b), SC.1 (b)} \quad n_{1,\dot{\alpha}} = n_{10,\dot{\alpha}} = 15,\ n_{2,\dot{\alpha}} = n_{9,\dot{\alpha}} = 20,\ n_{3,\dot{\alpha}} = n_{8,\dot{\alpha}} = 25,\ n_{4,\dot{\alpha}} = n_{7,\dot{\alpha}} = 50,\ n_{5,\dot{\alpha}} = n_{6,\dot{\alpha}} = 75.$$

It should be noted that due to the way in which the sample design has been defined, the probabilities of event given the covariates are roughly ordered from highest to lowest in the different strata. Therefore, by sampling many units from the strata at the edges, we are sampling more individuals with higher and lower probabilities (scheme (a)). In contrast, when sampling more individuals from the central strata, more individuals with medium probabilities of event are sampled (scheme (b)). The two sampling schemes differ on this point.

6. Depending on the sampling design defined in each scenario, sampling weights are calculated as follows. In scenarios designed based on one-stage stratified sampling (SH), the sampling weights are calculated as defined in eq. (2.7), i.e.,

$$w_i = \sum_{h=1}^{H} 1_{S_h}(i) \cdot \frac{N_h}{n_h}, \quad \forall i \in S. \tag{6.72}$$

Similarly, in the scenarios based on two-stage stratified cluster sampling (SC.0 and SC.1), the sampling weights are calculated as previously defined in eq. (2.28):

$$w_i = \sum_{h=1}^{H} \sum_{\dot{\alpha} \in \mathbb{A}_h} \frac{N_{h,\dot{\alpha}}}{n_{h,\dot{\alpha}}} \cdot \frac{A_h}{a_h} \cdot 1_{S_{h,\dot{\alpha}}}(i), \quad \forall i \in S. \tag{6.73}$$

6.3.2 Set-up

Considering the scenarios described in Section 6.3.1, a finite population was simulated in each scenario. The theoretical model is fitted to the finite population, and the ROC curve (ROC_{pop}) and AUC (AUC_{pop}) of this model are calculated following eqs. (6.10) and (6.12). Note that, as explained in Section 6.2.1, these parameters measure the performance of the theoretical finite population model. Each population is sampled $R = 500$ times, following in each case the corresponding complex sampling design. In each of the samples, a weighted logistic regression model was fitted and its ROC curve and AUC were estimated, ignoring the sampling weights (unweighted method, $\widehat{ROC}$ and $\widehat{AUC}$) and considering them (weighted method, $\widehat{ROC}_w$ and $\widehat{AUC}_w$). Note that in practice, we aim to analyze how those estimators perform in order to estimate the fitted model's ROC curve and AUC in the finite population. Therefore, in order to analyze and compare the performance of both estimators, we compare each of the estimates to the true finite population ROC curve and AUC estimates of the model fitted to the sample ($\widehat{ROC}_{\text{true}}$ and $\widehat{AUC}_{\text{true}}$, which are calculated by extending the fitted sample model to the finite population), rather than to the theoretical population model parameters. These parameters indicate the true performance of the fitted sample model in the finite population. This process is described in detail below and summarized in Figure 6.4. For each scenario:

Step 1. Generate the finite population U following the process described in Section 6.3.1. In particular, note that two finite populations have been generated: one for the scenarios SH and SC.0 (without cluster-level variables) and the other for the scenario SC.1 (with a cluster-level variable).

Step 2. Obtain the finite population model coefficients $\boldsymbol{\beta}^{\text{pop}}$ and the corresponding probabilities of event p_i^{pop}, $\forall i \in U$, following eqs. (6.8) and (6.9), respectively.

Step 3. Compute the ROC curve and AUC of the finite population model following eqs. (6.10) (ROC_{pop}) and (6.12) (AUC_{pop}), respectively.

Step 4. For $r = 1, \ldots, R$:

Step 4.1 Obtain a sample $S^r \subset U$ by means of one of the sampling designs described in Section 6.3.1 and calculate the sampling weights $w_i^r, \forall i \in S^r$ following the corresponding equation, eq. (6.72) or eq. (6.73).

Step 4.2 Fit the model to S^r maximizing the pseudo-likelihood function in eq. (6.13) by means of the covariate values $\boldsymbol{x}_i$ and the sampling weights w_i^r, $\forall i \in S^r$ (note that the latent variable values $\boldsymbol{z}_i$ are only considered to define the sampling design and are not considered in the model estimation process). Obtain $\widehat{\boldsymbol{\beta}}^r$ and the estimated probabilities of event $\hat{p}_i^r$, $\forall i \in S^r$.

Step 4.3 Estimate the ROC curve ($\widehat{ROC}^r_{\text{unw}}$ following eq. (6.6) and $\widehat{ROC}^r_{w}$ following eq. (6.18)) and AUC ($\widehat{AUC}^r_{\text{unw}}$ following eq. (6.7) and $\widehat{AUC}^r_{w}$ following eq. (6.27)), to obtain unweighted and weighted estimates, respectively. In addition, we estimate the AUC by means of pairwise sampling weights following the proposal of Yao et al. (2015) ($\widehat{AUC}^r_{\text{pairw}}$).

Step 4.4 By means of the $\widehat{\boldsymbol{\beta}}^r$ estimated in **Step 4.2**, estimate the probabilities of event for all the units in the finite population, $\hat{p}_i^r, \forall i = 1, \ldots, N$. Estimate the true ROC curve and AUC in the population following eqs. (6.15) and (6.17): $\widehat{ROC}^r_{\text{true}}$ and $\widehat{AUC}^r_{\text{true}}$.

Step 4.5 Calculate the difference between the unweighted or weighted estimates and the true population AUC:

$$\text{diff}^r_{\text{unw}} = \widehat{AUC}^r_{\text{unw}} - \widehat{AUC}^r_{\text{true}}, \quad \text{and} \quad \text{diff}^r_{w} = \widehat{AUC}^r_{w} - \widehat{AUC}^r_{\text{true}}. \tag{6.74}$$

In addition, in order to compare our proposal that considers marginal sampling weights to the proposal considering pairwise sampling weights, we define the difference between pairwise estimates to the true population model and to our proposal as follows:

$$\text{diff}^r_{\text{pairw}} = \widehat{AUC}^r_{\text{pairw}} - \widehat{AUC}^r_{\text{true}}, \quad \text{and} \quad \text{wdiff}^r = \widehat{AUC}^r_{\text{pairw}} - \widehat{AUC}^r_{w}. \tag{6.75}$$

All computations were performed in (64 bit) R 4.2.2 (R Core Team 2022) and a MacBook Pro equipped with 16GB of RAM, Apple M1 Chip, and macOS Monterey 12.2.1 operating system. Logistic regression models were fitted by means of `survey` R package (Lumley (2010), Lumley (2020)).

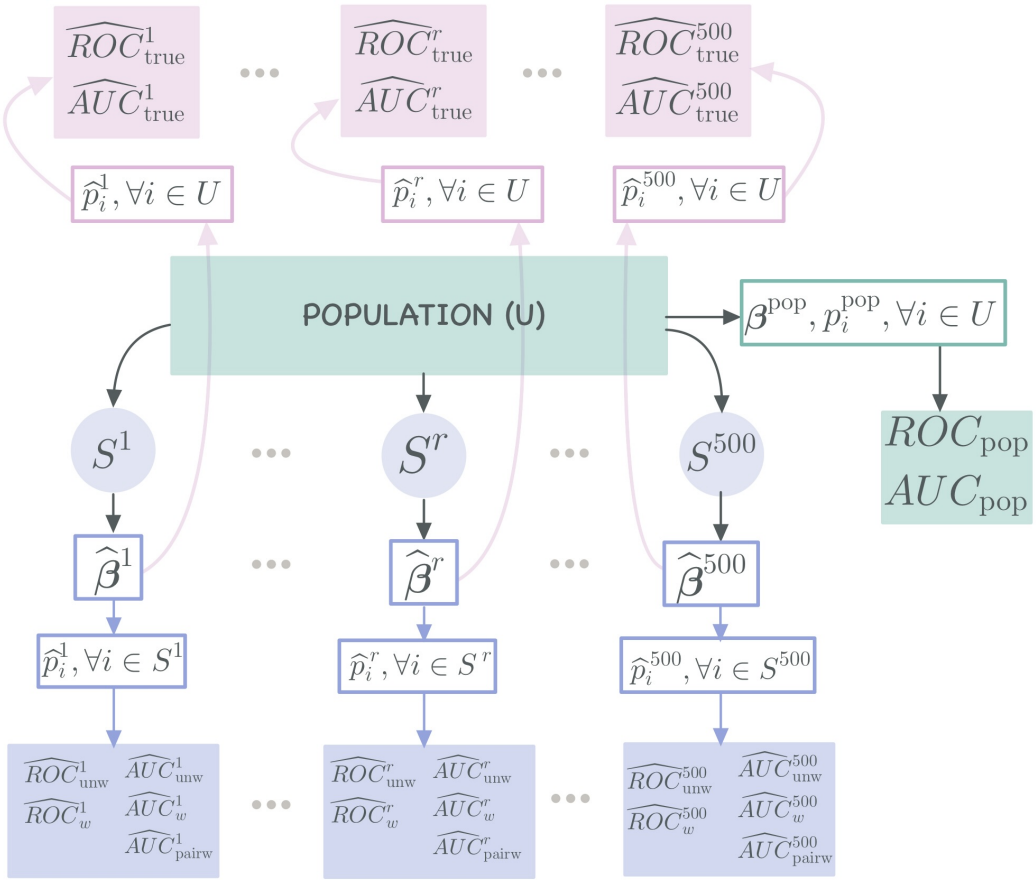


Figure 6.4: Graphical explanation of the simulation study set-up.

6.3.3 Results

In this section, we summarize the main results we obtained from the simulation study. In Figure 6.5, the ROC curve of the finite population model (ROC_{pop}), as well as the true population ROC curves ($\widehat{ROC}^r_{\text{true}}$) and the weighted ($\widehat{ROC}^r_{w}$) and unweighted ($\widehat{ROC}^r_{\text{unw}}$) estimates obtained based on the models fitted across $R = 500$ samples are shown. Figure 6.6 depicts the boxplots of the differences between the unweighted ($\text{diff}^r_{\text{unw}}$) and weighted ($\text{diff}^r_{w}$) estimates and the true population AUC of the models fitted to the samples (see eq. (6.74)). Figure 6.7 depicts the boxplots of the differences between the AUC estimates obtained by means of the pairwise and marginal sampling weights (wdiff^r). Table 6.1 summarizes the numerical results. Due to the large number of results we obtained, we begin by summarizing the main

conclusions, and then we proceed to analyze the differences between the different scenarios.

As shown in Figure 6.5, the ROC curve of the population model (ROC_{pop}) is above most of the true ROC ($\widehat{ROC}_{\text{true}}$) curves of the models fitted to the samples (i.e., estimated in the finite population based on the sample models). Similarly, as can be observed in Table 6.1, the average true population AUCs are lower than the AUCs of the population model. This indicates that population models have a greater discrimination ability than the models fitted to the samples, which is not at all surprising, given that the population model has been fitted with more data than the sample model. Therefore, in order to make fairer comparisons and compare the AUCs of the same models, we compare the ROC curve and AUC estimates obtained with different methods, to the true population parameters rather than the theoretical ones.

In general terms, the results of the simulation study show that, under the scenarios that have been considered, the weighted estimates of the AUC are closer than the unweighted ones to the true population AUC. The weighted estimates are slightly optimistic on average, given that a bit greater AUCs than the true ones have been estimated. In contrast, unweighted estimates sometimes overestimate the true finite population AUC, and other times underestimate it, depending on the scenario (in any case, showing a greater absolute bias than the weighted estimates). In terms of variability, no major differences have been observed between the two estimators, and, depending on the scenario, one estimator or the other shows more variability.

The marginal and pairwise weighted estimators perform quite similarly in all the scenarios, both in terms of bias as well as variability. However, it should be noted that as shown in Figure 6.7, the estimates based on pairwise sampling weights are slightly greater than the ones obtained based on marginal sampling weights. Thus, the estimates based on pairwise weights overestimate the true population AUC a little more than the estimates based on marginal weights, even though those differences are minimal in terms of bias. In contrast, computation times are considerably improved with the estimator proposed in this work (up to five times more efficient, as can be seen in Table 6.1), given that the pairwise sampling weights need to be calculated for each particular sampled pair, in contrast to the marginal ones, which are easily available in most cases when working with this kind of data.

We now proceed to comment on the results in more detail. First of all, it is important to understand the differences between the different scenarios. As explained above, the scenarios SH, SC.0 and SC.1 differ in the sampling design based on which

the finite population has been sampled and the number of cluster-level covariates available. In contrast, the sampling schemes (a) and (b) differ in the number of units sampled from each stratum and, more specifically, in the number of units sampled with (a) higher and lower (extreme) predicted probabilities or (b) central predicted probabilities. See Figures 6.8 (scheme (a)) and 6.9 (scheme (b)) for a clearer explanation of this point. These figures depict for each stratum the probabilities of event for all the sampled units in the iteration $r = 250$ in scenario SH obtained based on the fitted model, separately for units without (i.e., $\hat{p}_{i_0}$, $\forall i_0 \in S_0$) and with (i.e., $\hat{p}_{i_1}$, $\forall i_1 \in S_1$) the event of interest. Note that the predicted probabilities for both, units without and with the event of interest, decrease across different strata. In sampling scheme (a), more units are sampled from the strata located at the edges (see Figure 6.8) in comparison to sampling scheme (b), in which more units are sampled from central strata (see Figure 6.9). Therefore, note that in scheme (a) more comparisons are made between non-events from $h = 1$ and events from $h = 10$ than the ones would be made if the sample were obtained totally at random (i.e., by simple random sampling). Hence, note that more non-event and event pairs whose probabilities of event are "wrongly ordered" (i.e., $\hat{p}_{i_0} > \hat{p}_{i_1}$, where $i_0 \in S_0$, and $i_1 \in S_1$) are sampled in proportion, which decreases the estimate of the AUC for the unweighted method. This is the reason why the unweighted estimates of the AUC in scenarios with sampling scheme (a) underestimate the true AUC. In contrast, in sampling scheme (b), just the opposite situation occurs, given that fewer "wrongly ordered" pairs than the ones in the population have been sampled in proportion, which leads to an increment in the unweighted estimate of the AUC overestimating the true AUC. For the weighted estimates, no great differences have been observed in terms of difference in comparison to the true population AUC, neither in sampling scheme (a) nor in (b), given that the sampling weights correct the imbalances of sampling schemes giving to the pairs of units the relevance they should have in the finite population. For example, as can be observed in Table 6.1, in Scenario SC.0 (a) the average difference between the unweighted estimates and the true population AUC is -0.081, while in Scenario SC.0 (b) the average difference is 0.073. For the weighted estimates, under the same scenarios, the average differences are 0.005 and 0.008, respectively. These differences can also be observed in Figure 6.5, where the unweighted ROC curves are under the true population ROC curves in scenarios (a) while in scenarios (b) the unweighted ROC curves are over the true ones, as well as, over the weighted ones, indicating that the unweighted estimates overestimate more than the weighted ones in these scenarios. However, in terms of variability,

the performance of the unweighted and weighted estimates differ under sampling schemes (a) and (b). In scenarios considering sampling scheme (a) the variability of the unweighted estimates is greater than the variability of the weighted ones, while in scenarios considering sampling scheme (b) the difference is reversed. As shown in Table 6.1, in Scenario SH (a) the standard deviation of the unweighted estimates is 0.018, slightly greater than the variability of the weighted estimates which is 0.014. In contrast, in Scenarios SH (b) the standard deviation of the unweighted and weighted estimates are 0.012 and 0.020, respectively. In addition, the variability of the unweighted estimates is greater in (a) than in (b) (for the weighted estimates this difference is not as remarkable as for the unweighted estimates). For example, in SC.0 (a) the standard deviation of the unweighted estimates (0.035) is 2.5 times greater than the standard deviation in SC.0 (b) (0.014).

Results also show that the performance of the two estimators differs depending on the sampling design. In particular, a greater optimism of the weighted estimates has been observed in scenarios with cluster-level variables SC.1 than in scenarios SC.0 and SH. For example, in scenario SC.1 (a) the average difference between the weighted estimates and the true population has been 0.023 while in scenario SH (a) the average difference has been 0.005. This effect can also be observed in Figure 6.6. The ROC curves depicted in Figure 6.5 also show that in Scenarios SC.1 (a) and SC.1 (b) most of the weighted ROC curves are above the true population curves, while in the rest of the scenarios, the true population ROC curves are more or less in the center of the weighted ROC curves' band. This effect has not been observed for the unweighted estimates. In contrast, the sampling design has affected the variability of both, unweighted and weighted estimates. Specifically, the standard deviation of the estimates in scenario SH is lower than that in scenario SC.0 which, in turn, is lower than the standard deviation in scenario SC.1 (see Table 6.1 for more details). It should also be noted that the standard deviation of the true population AUCs across $R = 500$ samples is greater in scenarios SC.1 than in the rest of the scenarios (Table 6.1). This can also be observed in Figure 6.5, where the true population ROC curves show the greatest variability in scenarios SC.1.

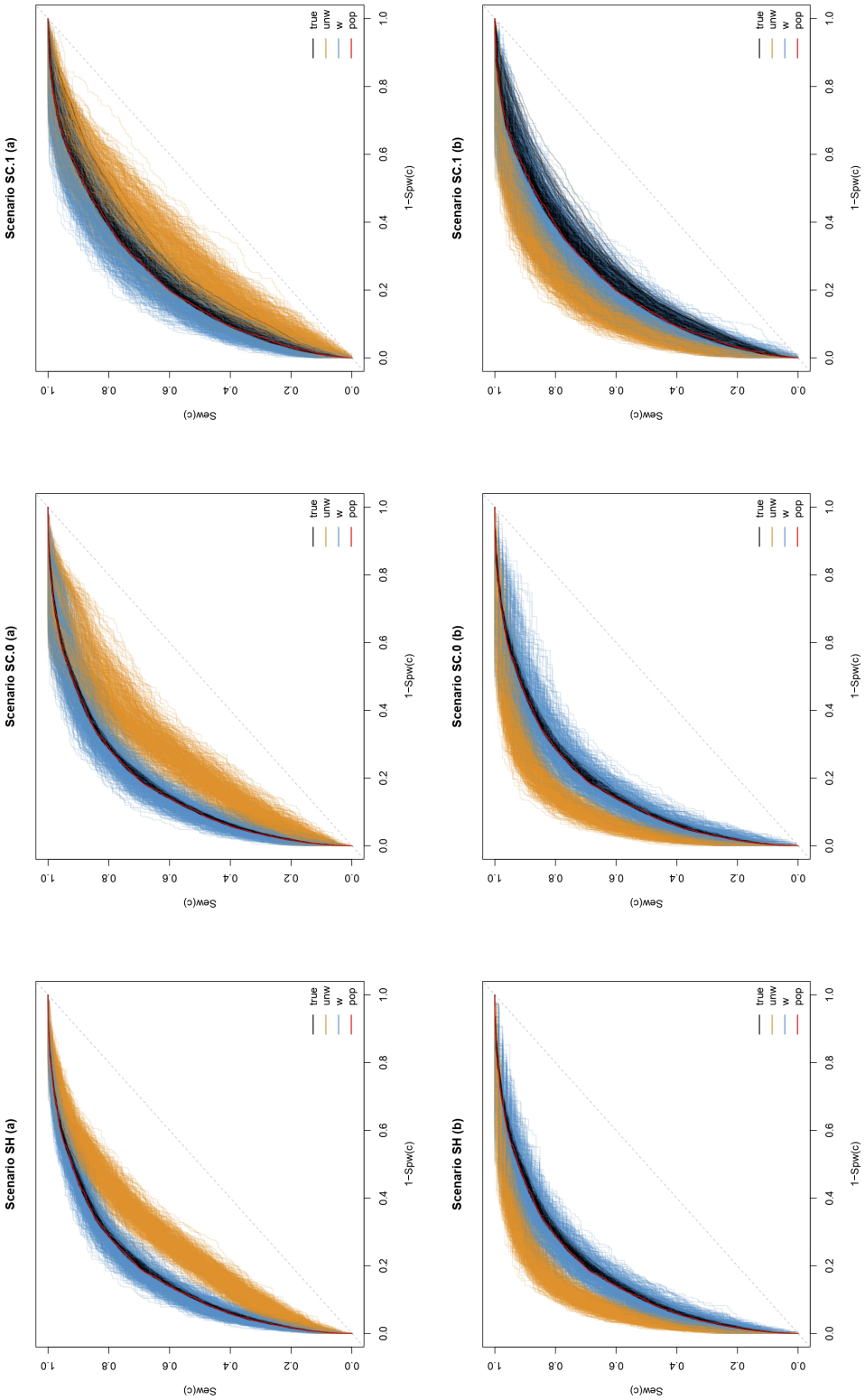


Figure 6.5: Unweighted (unw, see eq. (6.6)) and weighted (w, eq. (6.18)) estimates of the ROC curves, as well as the true population ROC curves (true) of the models fitted across $r = 1, \ldots, 500$ samples, together with the theoretical ROC curve (pop) of the model fitted to the finite population in each scenario drawn in the simulation study.

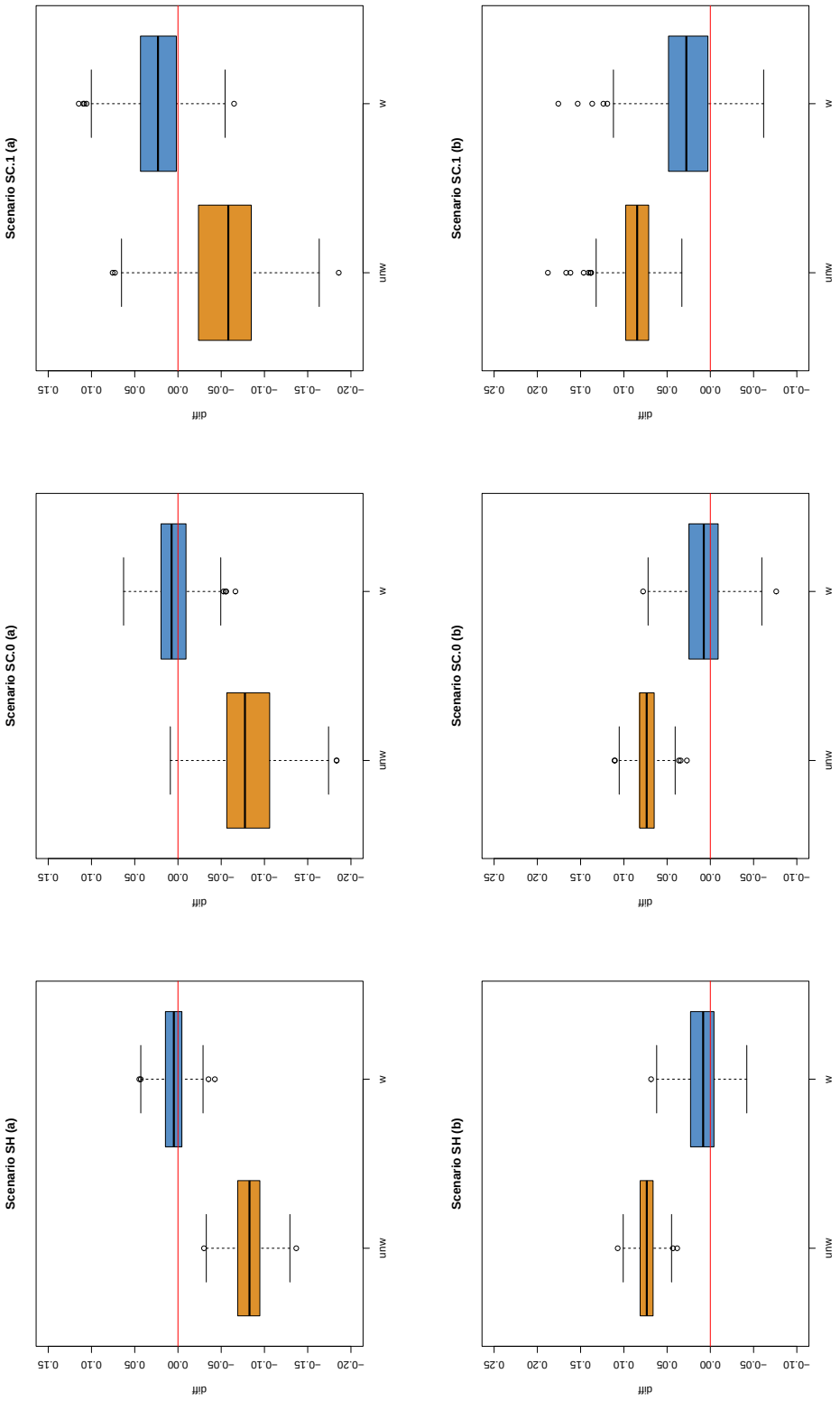


Figure 6.6: Boxplots of the difference (see eq. (6.74)) between the estimated AUCs by means of the unweighted (unw, eq. (6.7)) and weighted (w, eq. (6.27)) estimators and the true population AUC of the models fitted across $r = 1, \ldots, 500$ samples in all the scenarios drawn in the simulation study.

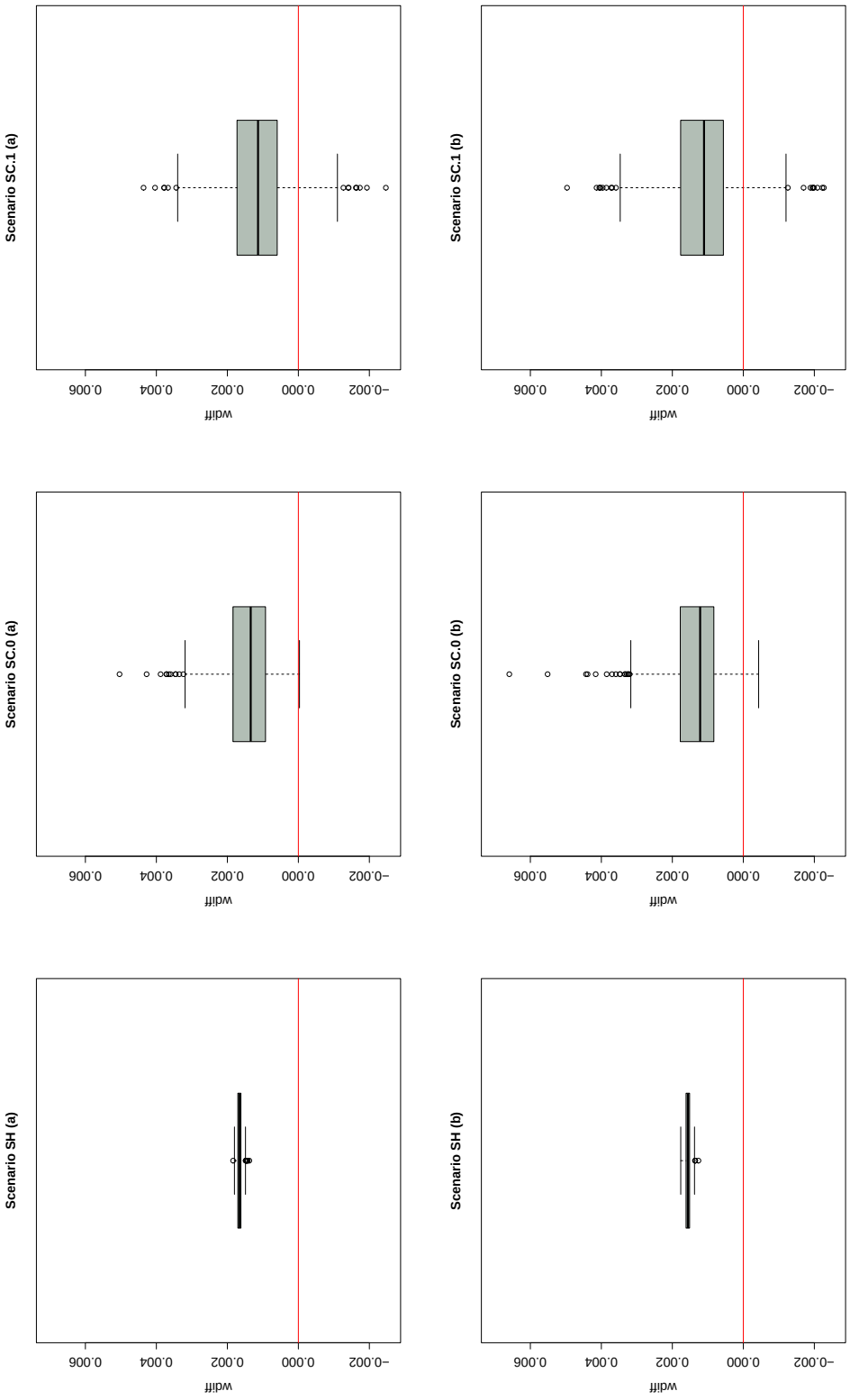


Figure 6.7: Boxplots of the differences between the estimated AUCs by means of the AUC estimator based on pairwise sampling weights and the one that considers marginal sampling weights (wdiff, see eq. (6.75)), when estimating the AUC of the models fitted across $r = 1, \ldots, 500$ samples in all the scenarios drawn in the simulation study.

Table 6.1: Numerical results of the minimum value (min), maximum value (max), average (mean) and standard deviation (sd) of the population AUC (true, eq. (6.17)), unweighted (unw, eq. (6.7)), weighted (w, eq. (6.27)) and pairwise (pairw, eq. (6.49)) estimates of the AUC of the models fitted across $r = 1, \ldots, 500$ samples. The average difference (av. diff) of the unweighted, weighted and pairwise estimates to the true population AUC estimates (see eqs. (6.74) and (6.75)) with their standard deviations (sd). The average computational times (av. time, in seconds) of each method with their standard deviations (sd) are also shown. In addition, the theoretical AUC (AUC_{pop}) of the finite population model in each scenario is available.

	AUC_{pop}		**min**	**max**	**mean (sd)**	**av. diff (sd)**	**av. time (sd)**
SH (a)	0.835	true	0.823	0.835	0.832 (0.002)		
		unw	0.697	0.801	0.750 (0.018)	-0.082 (0.018)	0.003 (0.001)
		w	0.789	0.878	0.838 (0.014)	0.005 (0.014)	0.004 (0.004)
		pairw	0.791	0.880	0.839 (0.014)	0.007 (0.014)	0.010 (0.004)
SC.0 (a)	0.835	true	0.824	0.835	0.832 (0.002)		
		unw	0.650	0.840	0.751 (0.035)	-0.081 (0.035)	0.003 (0.005)
		w	0.767	0.891	0.837 (0.022)	0.005 (0.022)	0.004 (0.002)
		pairw	0.768	0.892	0.839 (0.022)	0.006 (0.022)	0.016 (0.007)
SC.1 (a)	0.784	true	0.712	0.784	0.772 (0.011)		
		unw	0.595	0.835	0.718 (0.042)	-0.054 (0.046)	0.003 (0.001)
		w	0.718	0.876	0.795 (0.028)	0.023 (0.032)	0.004 (0.004)
		pairw	0.721	0.879	0.796 (0.028)	0.024 (0.032)	0.014 (0.004)

Table 6.2: Numerical results of the minimum value (min), maximum value (max), average (mean) and standard deviation (sd) of the population AUC (true, eq. (6.17)), unweighted (unw, eq. (6.7)), weighted (w, eq. (6.27)) and pairwise (pairw, eq. (6.49)) estimates of the AUC of the models fitted across $r = 1, \ldots, 500$ samples. The average difference (av. diff) of the unweighted, weighted and pairwise estimates to the true population AUC estimates (see eqs. (6.74) and (6.75)) with their standard deviations (sd), and the average computational times (av. time, in seconds) of each method with their standard deviations (sd) are also shown. In addition, the theoretical AUC (AUC_{pop}) of the finite population model in each scenario is available.

	$AUC_{\textbf{pop}}$		**min**	**max**	**mean (sd)**	**av. diff (sd)**	**av. time (sd)**
SH (b)	0.835	true	0.816	0.835	0.831 (0.003)		
		unw	0.870	0.939	0.904 (0.012)	0.073 (0.011)	0.003 (0.004)
		w	0.791	0.896	0.839 (0.020)	0.008 (0.020)	0.004 (0.001)
		pairw	0.792	0.898	0.842 (0.020)	0.010 (0.020)	0.011 (0.005)
SC.0 (b)	0.835	true	0.818	0.835	0.831 (0.003)		
		unw	0.857	0.943	0.904 (0.014)	0.073 (0.013)	0.003 (0.004)
		w	0.754	0.910	0.838 (0.026)	0.008 (0.026)	0.004 (0.002)
		pairw	0.756	0.911	0.840 (0.026)	0.009 (0.026)	0.016 (0.004)
SC.1 (b)	0.784	true	0.696	0.783	0.771 (0.012)		
		unw	0.785	0.905	0.856 (0.019)	0.086 (0.021)	0.003 (0.004)
		w	0.713	0.878	0.798 (0.031)	0.027 (0.036)	0.003 (0.004)
		pairw	0.713	0.879	0.799 (0.031)	0.028 (0.036)	0.015 (0.003)

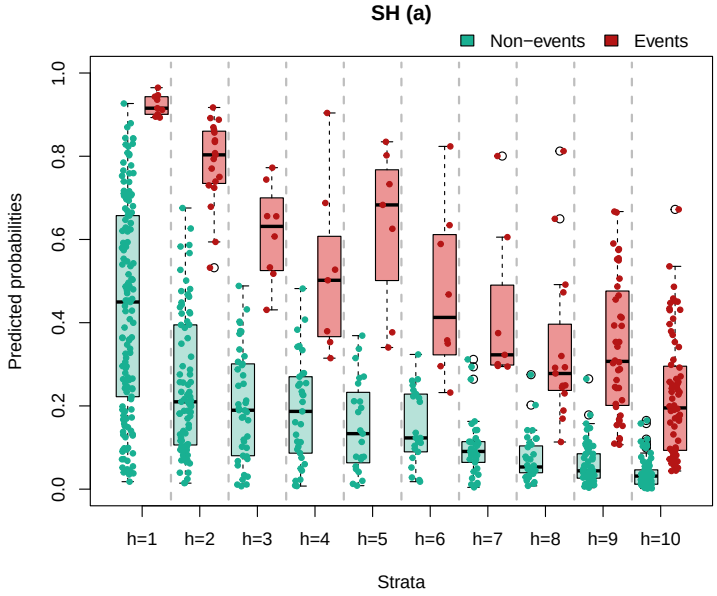


Figure 6.8: Probabilities of event for the sampled units obtained based on the fitted model in scenario SH (a) for the sample $r = 250$.

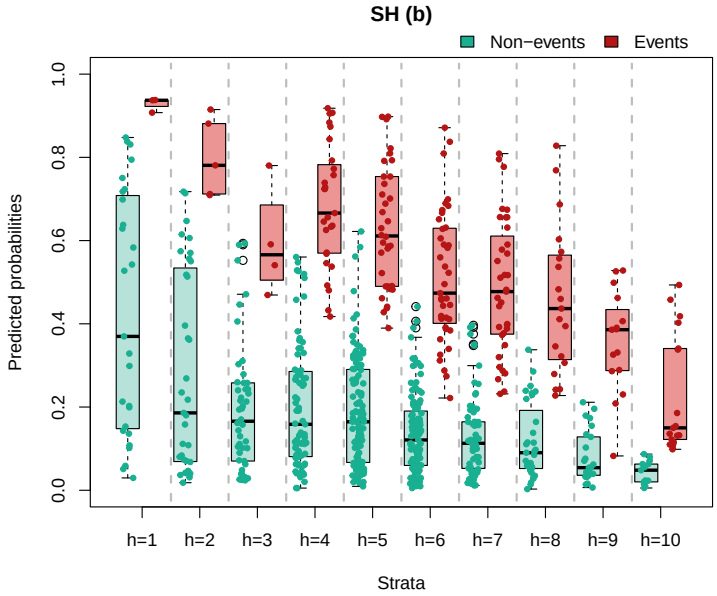


Figure 6.9: Probabilities of event for the sampled units obtained based on the fitted model in scenario SH (b) for the sample $r = 250$.

6.3.4 What if the design is uninformative to fit a particular model?

In previous chapters (particularly, in Chapter 4), we have discussed that if a "perfect" model is fitted, in the sense that it contains all the information of the sampling design as covariates in the model, previous works show that, then sampling weights are not needed to fit that model (see, e.g., Scott and Wild (1986) and Pfeffermann (1993)). In the simulation study depicted in Section 6.3.2, we avoid fitting "perfect" logistic regression models in order to simulate more realistic scenarios. However, we also find it interesting to give an answer to the following question: if we fit a "perfect" logistic regression model, can sampling weights also be ignored when estimating the ROC curve and AUC of that model? Or, in contrast, do they play an important role in the estimation process of those parameters even though they are not necessary to fit the model? The goal of this section is to shed light on this question.

With this goal in mind, we modify one of the scenarios considered in the above-mentioned study. In particular, we focus on the one-stage stratification (**SH**) in order to fit perfect prediction models in which the design variables Z_1 and Z_2 are also considered as covariates when fitting the models. First, we slightly modify the data generation process described in Section 6.3.1. Specifically, we define $p = 3$ covariates (X_1, X_2, X_3) and two design-variables (Z_1, Z_2), being all of them unit-level variables (i.e., $d = 0$ cluster-level variables). We increase the correlation between covariates by setting $\eta = 0.25$ in eq. (6.67) and we modify the coefficient values as indicated in eq. (6.76):

$$\begin{aligned} \beta_0 &= -1.5, \\ \boldsymbol{\beta^X} &= (0.5, 0.5, 0.5)^T, \\ \boldsymbol{\beta^Z} &= (-1, -1)^T. \end{aligned} \tag{6.76}$$

Finally, in the simulation set-up defined in Section 6.3.2, in **Step 4.2** in addition to the covariates values $\boldsymbol{x}_i$, we also consider the variable values $\boldsymbol{z}_i, \forall i \in S^r$ when fitting the models. We keep the rest of the data generation and the simulation set-up steps as described in Sections 6.3.1 and 6.3.2 (including the number of units sampled from each stratum, denoted as sampling schemes **SH (a)** and **SH (b)**).

Coefficient estimates are obtained by maximizing the pseudo-likelihood function defined in eq. (6.13). We have checked that the unweighted and weighted model coefficients (estimated by maximizing the likelihood (eq. (6.4)) and the pseudo-likelihood (eq. (6.13)) functions, respectively) are indeed quite similar, indicating

that sampling weights are not needed when fitting the models (even though we did not find it necessary to mention in the previous section, we take this opportunity to clarify that in the previous simulation study, in which design-variables are not considered as model covariates, unweighted and weighted model estimates clearly differ).

Results of the small simulation study described above are depicted in Figures 6.10, 6.11 and 6.12. These figures show that the sampling weights, although unnecessary when fitting the models, may be crucial when estimating the ROC curve and AUC of those models. The results are similar to the ones shown and discussed in Section 6.3.3. Briefly, as shown in Figures 6.11 and 6.11, unweighted estimates sometimes overestimate and otherwise underestimate the true ROC curve and AUC of the fitted models, and for obtaining better estimates, sampling weights should be considered. It should be noted that the differences between sampling schemes **SH (a)** and **SH (b)** are reversed to the ones discussed in Section 6.3.3. However, note that this effect is not surprising given that when changing the theoretical model coefficients to the ones in eq. (6.76), then the probabilities of event of the units in the finite population are also different to the ones shown in Figures 6.8 and 6.9. Anyway, the differences in the behavior of unweighted and weighted estimates can be explained in a similar way. In addition, as can be observed in Figure 6.12, differences between the weighted estimates based on marginal and pairwise sampling weights are similar to the ones depicted in Figure 6.7.

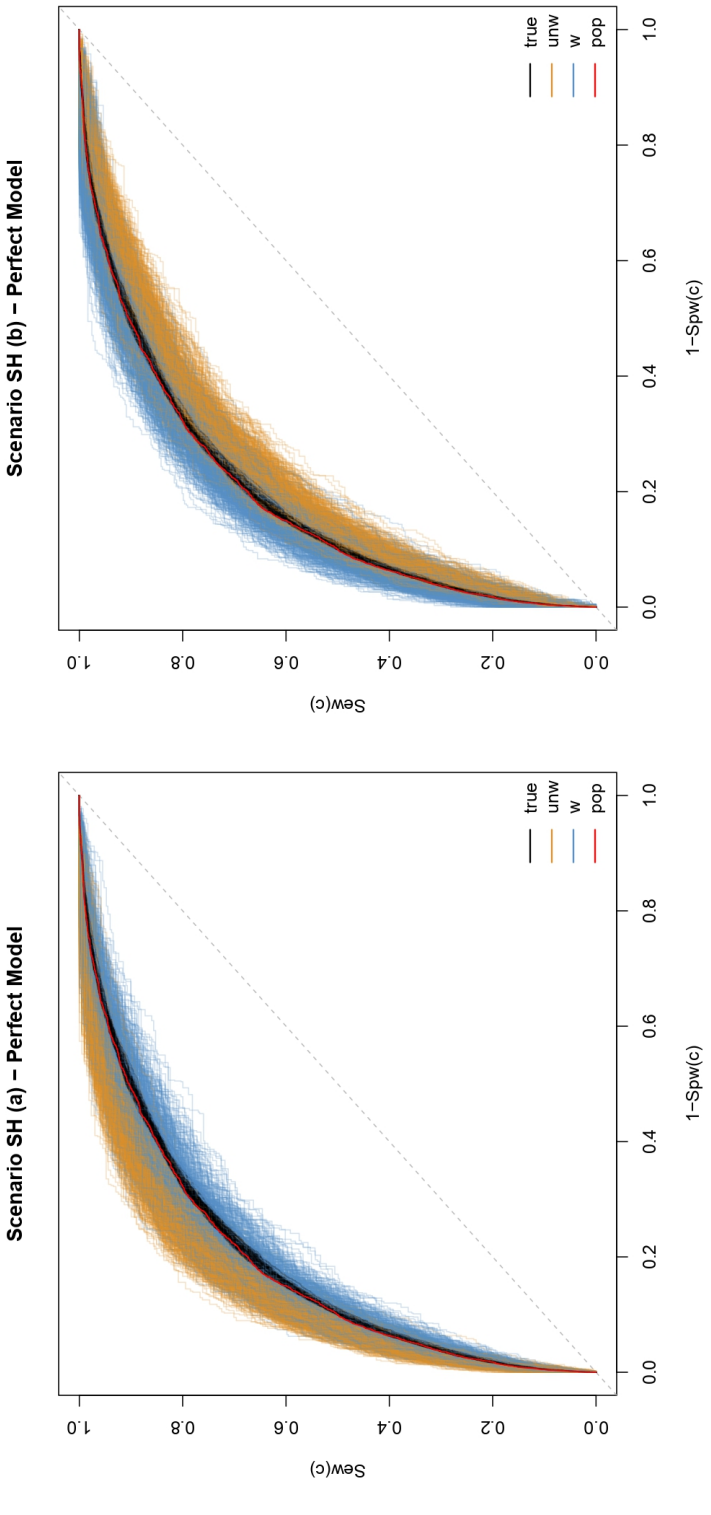


Figure 6.10: Unweighted (unw, see eq. (6.6)) and weighted (w, eq. (6.18)) estimates of the ROC curves, as well as the true population ROC curves (true) of the models fitted across $r = 1, \ldots, 500$ samples, together with the theoretical ROC curve (pop) of the model fitted to the finite population, when considering "perfect" models.

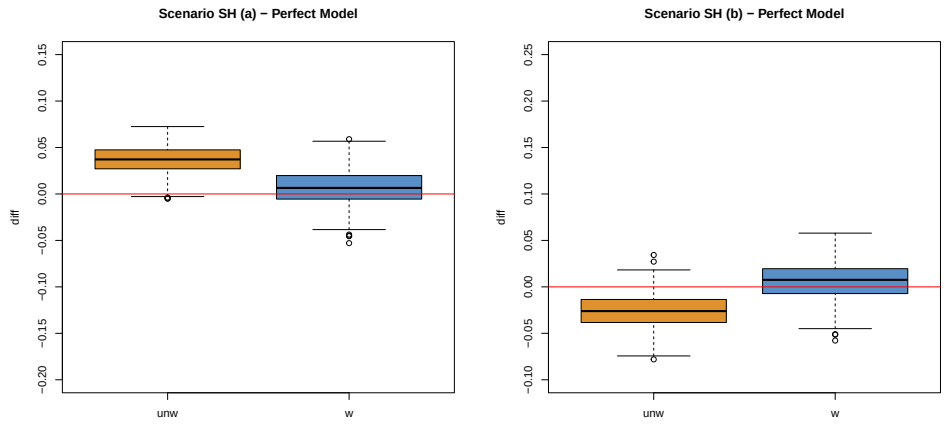


Figure 6.11: Boxplots of the difference (see eq. (6.74)) between the estimated AUCs by means of the unweighted (unw, eq. (6.7)) and weighted (w, eq. (6.27)) estimators and the true population AUC of the models fitted across $r = 1, \ldots, 500$ samples considering "perfect" models.

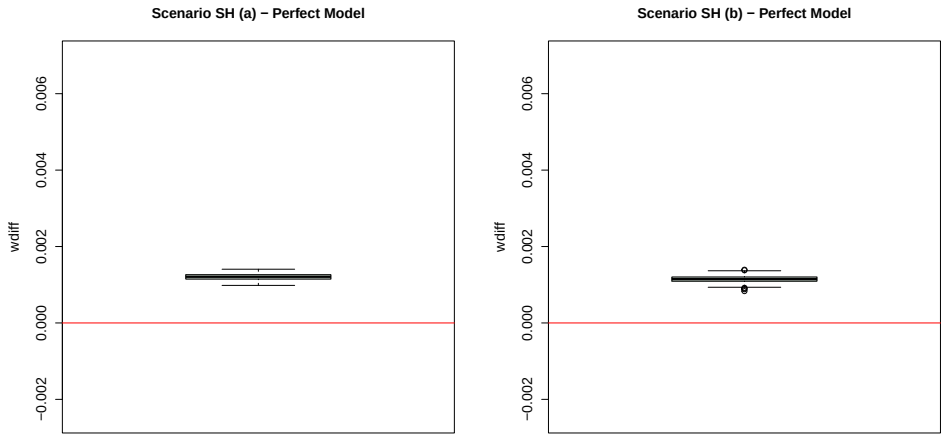


Figure 6.12: Boxplots of the differences between the estimated AUCs by means of the AUC estimator based on pairwise sampling weights and the one that considers marginal sampling weights (wdiff, see eq. (6.75)), when estimating the AUC of the models fitted across $r = 1, \ldots, 500$ samples considering "perfect" models.

6.4 Application to ESIE survey data

The methodology proposed in Section 6.2 has been applied to the ESIE survey, which was described in detail in Chapter 3. All the establishments in BC were considered for this application (i.e., the whole sample with $n = 7\,725$ was used) and the AUC of the model fitted in Chapter 4 was estimated. Covariates included in the model represent the activity of the company, the number of employees, and the ownership.

The unweighted and weighted AUC estimates and the corresponding Bootstrap 95% confidence-intervals (CI) are shown in Table 6.3. For the 95% CI of the unweighted estimate, the Bootstrap confidence interval is calculated by means of the `pROC` R package (Robin et al. 2011), while the 95% CI of the weighted estimate is calculated by generating Bootstrap resamples based on replicate weights (Rao and Wu 1988) using the `survey` R package (Lumley 2020), both of them considering $B = 2000$ Bootstrap resamples. The unweighted and weighted ROC curve estimates are depicted in Figure 6.13. Note that in this case, as we are working with real survey data, we cannot know which the true population ROC curve and AUC are.

Even though the differences between the unweighted and weighted estimates are not as large as the ones analyzed in the simulation study, the unweighted estimate is larger than the weighted estimate, as it happens in sampling scheme (b) of the simulation study. Considering the results of the simulation study, we can assume that the weighted estimate will be a bit above the true population AUC, and therefore, we can conclude that probably the unweighted estimate of the AUC is overestimating it. In addition, note that the overlap between the two confidence intervals is very slight.

Table 6.3: Estimated unweighted and weighted AUCs and the corresponding Bootstrap 95% CI of the model fitted to ESIE survey data.

	Estimated AUC	**95% CI (Bootstrap)**
Unweighted	0.831	$0.823 - 0.840$
Weighted	0.809	$0.795 - 0.823$

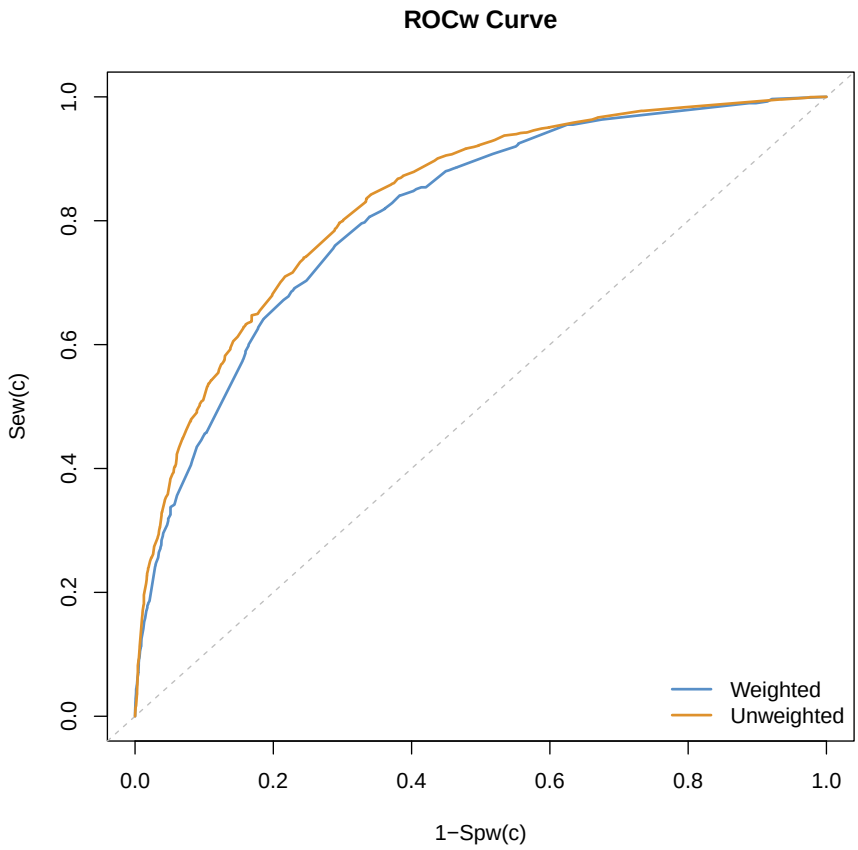


Figure 6.13: Weighted and unweighted ROC curves of the models fitted to the ESIE survey data.

6.5 Discussion

In this work, we propose new weighted estimators to estimate the ROC curve and AUC of logistic regression models fitted to complex survey data. In addition, we show that the area under the proposed weighted estimator of the ROC curve is equivalent to the weighted Mann-Whitney U-statistic incorporating marginal sampling weights, which are defined as the inverse probability weights for each sampled unit. A simulation study has been conducted in order to analyze the performance of the proposed estimators and they have also been applied to real survey data.

The results of the simulation study suggest the use of the proposed weighted estimators rather than the unweighted ones. In addition, the results of this study

also show that even when sampling weights are not needed for estimating logistic regression models, they may be important when estimating the ROC curve and AUC of those models.

The unweighted estimators overestimate or underestimate the true population parameters, depending on the proportion of units sampled from each stratum. In particular, as more units with extreme (higher and lower) predicted probabilities are sampled in proportion, more non-events with higher predicted probabilities as well as events with lower predicted probabilities are also sampled. This results in a lower estimate of the AUC. In contrast, as more central predicted probabilities are sampled, less extreme (higher and lower) predicted probabilities than necessary to properly represent the finite population will be sampled, leading to a greater estimate of the AUC due to the same reason. Weighted estimates correct for this bias, providing ROC curve and AUC estimates that are closer to the true finite population parameters since the presence of sampling weights gives each pair of individuals with and without the event of interest the relevance that they should have in representing the finite population.

The ROC curve and AUC estimated by means of the proposed estimators slightly overestimate the true population parameters. Moreover, we have also observed that the optimism is higher when a variable at the cluster level is included in the model, increasing in this way, the cluster effect. We decided to compare weighted and unweighted estimates to the true finite population parameters of the sample model rather than to the theoretical parameters of the population model. At this point, it should be noted that the bias we just mentioned would be lower if the comparisons were made considering the parameters of the population model. However, we believe that comparing the estimates to the population parameters of the sample model is fairer (due to the reasons given throughout the chapter) and we consider it is important to mention the optimism of the weighted estimates.

In any case, we believe that the optimism observed in the weighted estimates is not due to a bias of the estimator itself, but rather to the fact that the same data set is being used both to estimate the model and to estimate its discrimination ability. In other words, when working with a simple random sample, it is well known that if the same data is used to fit the model as well as to evaluate its predictive performance, this estimate is usually optimistic given that the fitted models are optimal for data that have been considered for that purpose. This effect is known as "overfitting" (see, e.g., Copas (2002), Steyerberg (2008)). Several validation techniques such as Bootstrap and K-fold cross-validation are proposed to correct for the optimism of

the AUC estimates when working with simple random samples (see, e.g., Airola et al. (2011), Austin and Steyerberg (2017), Copas (2002), Iparragirre et al. (2019), Steyerberg et al. (2001)). We believe it would be interesting to understand how the optimism of the weighted estimates varies in different scenarios, as well as, to analyze the performance of different replicate weights methods to correct for the optimism of the proposed estimators in the context of complex survey data. This study will be conducted as further work.

In this work, we propose the use of marginal sampling weights rather than pairwise sampling weights as proposed by Yao et al. (2015) to estimate the AUC (in order to avoid confusion with the title of the paper, it should be noted that even though the authors mention the ROC curve in the main title, they make a proposal for estimating the AUC but not the ROC curve). The results suggest that the differences between the two weighted estimators are small, at least under the scenarios that have been considered, but the estimates obtained based on pairwise sampling weights are slightly more optimistic than the ones obtained based on marginal sampling weights. Furthermore, it is worth noting that another disadvantage is that the AUC estimates obtained considering the pairwise sampling weights may result in AUCs greater than 1, given the way in which the estimator is defined (see eq. (6.49)). In addition, computation times are considerably improved with the estimator proposed in this work. As further work, it would be interesting to analyze and compare the mathematical properties of both estimators, which consider the marginal and pairwise sampling weights. In addition, in this chapter, we show that the matrix form expression proposed by Yao et al. (2015) to summarize their estimator is not equal to their proposal but to ours, which considers marginal sampling weights. This issue has caused us many troubles in carrying out this work, specifically when comparing both proposals and analyzing the differences between them, and therefore, we have considered it convenient to mention it.

Another interesting issue that is not handled in this chapter is the variance of the proposed ROC curve and AUC estimators. Estimation of the variance for those estimators would allow us to define confidence intervals for estimates of the ROC curve and AUC in the context of complex survey data. Yao et al. (2015) proposed the Jackknife and Balanced Repeated Replication methods (Wolter 2007) to estimate the variance of their AUC estimator. Those methods could also be applied to estimate the variance of the AUC estimator we propose in this work. In addition, we believe that the Rescaling Bootstrap (Rao and Wu 1988), which is a modification of the traditional Bootstrap (Efron and Tibshirani 1994) and which has

been defined in Chapter 5, could also be an interesting technique for this purpose, given the popularity of the Bootstrap for variance estimation in simple random samples (we have implemented this method to calculate the confidence interval of the weighted AUC estimate in the application). However, in addition to checking the validity of these methods for variance estimation, it could also be interesting to obtain analytical expressions for the variance of the proposed AUC as well as ROC curve estimators. Comparison of the performance of different validation techniques to estimate the variance of the proposed estimators by means of a simulation study, as well as the analytical expression for the variance of those parameters, will be developed as future work.

Note that in this study, we work with probability samples and we assume 100% of response rate. Neither other types of sampling, such as sampling probability proportional to sample size nor post-stratification or calibration of the weights have been considered, so the conclusions obtained are limited to the scenarios that have been analyzed.

To sum up, in order to obtain unbiased estimates, we recommend the use of the estimators proposed in this work to estimate the ROC curve and AUC of logistic regression models when working with complex survey data. An R package (`wROC`) has been developed and is available on GitHub and explained in detail in Chapter 8 to plot ROC curves and estimate the AUC to complex survey data by means of the proposed methods.

CHAPTER 7

Estimation of optimal cut-off points for individual classification

The paper related to the work presented in this chapter has been published:

Iparragirre, A., Barrio, I., Aramendi, J. & Arostegui, I. (2022) Estimation of cut-off points under complex-sampling design data. SORT-Statistics and Operations Research Transactions*, 46(1), 137–158.*

This chapter mostly replicates the above-mentioned article. However, some changes have been made to keep the notation and ensure cohesion with the rest of the document. In addition, Section 7.3.3 incorporates new contents that were not included in the above-mentioned paper.

The methods proposed in this chapter have been incorporated into the R package `wROC`*, which is available on GitHub*[10]*.*

[10] https://github.com/aiparragirre/wROC

Summary

In the context of logistic regression models, a cut-off point is usually selected to dichotomize the predicted probabilities of event estimated based on the model. The techniques proposed to estimate optimal cut-off points in the literature are commonly developed to be applied in simple random samples, and their applicability to complex sampling designs could be limited. Therefore, in this work, we propose a methodology to incorporate sampling weights in the estimation process of the optimal cut-off points, and we evaluate its performance by means of a simulation study based on real survey data. The results suggest the convenience of considering sampling weights for estimating optimal cut-off points in the context of complex survey data.

7.1 Introduction

In this chapter, we focus on the logistic regression framework to predict a dichotomous response variable Y. As discussed in previous chapters, from a practical point of view, one of the most important characteristics of these kinds of models is the support they provide for decision-making, since increasing knowledge about potential predictors helps the decision-making process (Baker and Gerdin 2017, Steyerberg 2008). In this context, decisions such as whether or not to recommend a patient to start treatment, or to give a diagnosis about a disease, are based on the individual risk or probability of event given by the estimates of the logistic regression model. In order to make these decisions, first, for each individual, the predicted probability of event is classified based on a cut-off point. In this way, for example, if the individual's probability of suffering from extreme poverty is greater than the selected cut-off point, he or she is assigned a social benefit, while in contrast, if that is lower, no social support is provided (Pauker and Kassirer 1980, Steyerberg 2008). Hence, cut-off point estimation is widely employed in practice in the field of prediction models, especially but not exclusively, in clinical prediction models (Chen et al. 2015, Spence et al. 2018, Steyerberg et al. 1999).

At this point, the main issue is usually to select a valid cut-off point that will provide the best classification of individuals in practice. Many strategies have been proposed in the literature in order to estimate optimal cut-off points. It should be noted that we can not talk about optimal cut-off points in general terms. In

contrast, a cut-off point will or will not be optimal depending on the objective of a particular study. Therefore, when we talk about selecting an optimal cut-off point, we are talking about selecting the one that satisfies a certain optimality criterion. Hence, as we have mentioned above, different techniques have been proposed to select optimal cut-off points, given a particular criterion. For instance, some of those methods select the optimal cut-off point with the aim of obtaining a certain value of sensitivity or specificity (i.e., probability of classifying correctly an individual with or without the event of interest) or maximizing a function of these two parameters as for example the Youden index (Youden 1950). Some others select the cut-off point that maximizes some particular indexes, such as Kappa (Cohen 1960, Greiner et al. 2000). Greiner (1995; 1996) proposed a method to select the optimal cut-off point that minimizes the error or either maximizes the accuracy of the classification rule. There are some other methods that select optimal cut-off points based on some other criteria related to several parameters such as predicted values (i.e., probability of event/non-event for an individual classified as event/non-event) (Vermont et al. 1991) or the probability of event in the population (Manel et al. 2001), among others. Besides, other methods are based on the analysis of the cost of incorrect and the benefit of correct diagnosis (Pauker and Kassirer 1980, Swets 1992, Wynants et al. 2019). An extensive review of those techniques can be found in López-Ratón et al. (2014).

However, those techniques have usually been designed and applied for simple random samples and, as far as we know, there is a lack of proposals to consider complex sampling designs, and in particular sampling weights, throughout the estimation process of optimal cut-off points. Following the same direction as in the previous chapters of this dissertation, we believe that sampling weights should not be ignored when estimating optimal cut-off points when working with complex survey data. Therefore, in this chapter, we propose a modification of the methods that have been proposed in the literature to select optimal cut-off points of the probability of event in the logistic regression framework, considering sampling weights in the estimation process. In addition, the performance of the proposed methods is compared to the performance of those which ignore the sampling weights, by means of a simulation study. In particular, we focus on surveys that are based on one-stage stratified sample designs.

The rest of the chapter is organized as follows. In Section 7.2, we recall some basic notation that will be used throughout the rest of the chapter. Then, we describe some of the methods that are usually applied in practice to estimate optimal cut-off

points of the probability of event in the logistic regression framework, and finally, we propose a new methodology that takes into account the effect of the sampling weights in the cut-off point estimation process. In Section 7.3, we describe the simulation process that has been carried out so as to study the performance and effectiveness of the proposed methods to incorporate sampling weights into the estimation process of optimal cut-off points and we show the results we have obtained. The methodology proposed in this work has been applied to ESIE survey data and this application is described in Section 7.4. Finally, we conclude with a discussion in Section 7.5.

7.2 Methods

In this section, we begin by describing some of the methods that are usually applied for estimating optimal cut-off points in the context of diagnostic tests in general and logistic regression models in particular, based on different optimality criteria for simple random samples. Next, we develop a new estimation method, in which we propose introducing the sampling weights in these existing methods for simple random samples so that they are valid in samples derived from complex sampling designs.

Let $\{(y_i, \boldsymbol{x}_i, w_i)\}_{i \in S}$ indicate the set of observations of the response variable Y, predictor variables $(1, X_1, \ldots, X_p)$ and sampling weights for the individuals in sample S. For each sampled unit $i \in S$, its probability of event is described as $p(\boldsymbol{x}_i) = P(Y = 1 | \boldsymbol{X} = \boldsymbol{x}_i)$, which can be estimated as follows (previously defined in eq. (6.14)):

$$\hat{p}_i = \hat{p}(\boldsymbol{x}_i) = \frac{e^{\boldsymbol{x}_i \hat{\boldsymbol{\beta}}}}{1 + e^{\boldsymbol{x}_i \hat{\boldsymbol{\beta}}}} \quad \forall i \in S, \tag{7.1}$$

where the estimated regression coefficients $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \ldots, \hat{\beta}_p)^T$, are obtained by maximizing the pseudo-likelihood function in eq. (7.2) in the complex survey data framework (defined in eq. (2.58)):

$$PL(\boldsymbol{\beta}) = \prod_{i \in S} p(\boldsymbol{x}_i)^{y_i w_i} (1 - p(\boldsymbol{x}_i))^{(1-y_i) w_i}. \tag{7.2}$$

7.2.1 Optimal cut-off point estimation methods

As introduced above, it is usually very useful in practice to select a cut-off point in order to distinguish between units with and without the event of interest. In our particular case, we are interested in discriminating between units with and without

the event of interest based on their estimated probability of event. In this context, one observation $i \in S$ is usually classified as event if its estimated probability of event $\hat{p}_i$ exceeds a previously selected cut-off point c (Magder 2003, Pepe 2003). However, it should be noted that this classification may be correct or incorrect depending on the selected cut-off point c. The correct classification of an observation with the event of interest is usually denoted as *true positive*, while the correct classification of an observation without the event of interest is commonly denoted as *true negative*. But usually, those classifications are not entirely accurate. Therefore, some of the observations are commonly classified incorrectly: an observation with the event of interest may be classified as non-event (*false negative*) or an observation without the event of interest may be classified as event (*false positive*).

Therefore, different methods of estimation of the optimal cut-off point have been developed in the literature, with the aim of optimizing diverse measures. In particular, many methods consist on the optimization of an objective function of the Receiver Operating Characteristic (ROC) curve, which as described in Chapter 6, is a curve that measures the global accuracy of a logistic regression model (Bamber 1975, Pepe 2003). Let us recall that, as described in eq. (6.6), the ROC curve of a logistic regression model can be defined as follows (Hosmer and Lemeshow 2000, Pepe 2003):

$$ROC(\cdot) = \{(1 - Sp(c),\ Se(c))\,,\ c \in (-\infty, \infty)\}\,, \tag{7.3}$$

where specificity ($Sp(c)$) and sensitivity ($Se(c)$) parameters are defined as the probabilities of properly classifying units without and with the event of interest, respectively, as shown in eq. (7.4) (note that these parameters have previously been defined in eq. (6.2)):

$$Sp(c) = P\left[p(\boldsymbol{x}_i) < c | Y = 0\right] \quad \text{and} \quad Se(c) = P\left[p(\boldsymbol{x}_i) \geq c | Y = 1\right], \tag{7.4}$$

As previously stated in eq. (6.5), specificity and sensitivity parameters are estimated as follows in the simple random sample context:

$$\widehat{Sp}(c) = \frac{1}{n_0} \sum_{i_0 \in S_0} I(\hat{p}_{i_0} < c) \quad \text{and} \quad \widehat{Se}(c) = \frac{1}{n_1} \sum_{i_1 \in S_1} I(\hat{p}_{i_1} \geq c), \tag{7.5}$$

where S_0 and S_1 indicate the subsets of units without and with the event of interest, respectively. In other words, specificity and sensitivity parameters are estimated as the proportions of correctly classified sampled units without and with the event of

interest, respectively, based on a particular cut-off point c.

In this study, we have focused on some of the optimal cut-off point estimation methods based on several optimality criteria related to sensitivity and specificity parameters. In particular, in this work, we consider the four following methods:

- *Youden* (Greiner et al. 2000, Youden 1950): This method selects the cut-off point (c^{Youden}) that maximizes the Youden Index, which is defined as the sum of sensitivity and specificity parameters minus one, i.e.,

$$c^{\text{Youden}} = \underset{c}{\text{argmax}} \left\{ \widehat{Se}(c) + \widehat{Sp}(c) - 1 \right\}. \tag{7.6}$$

- *MaxProdSpSe* (Lewis et al. 2008): This method selects the cut-off point c that maximizes the product between sensitivity and specificity parameters, i.e.,

$$c^{\text{MaxProdSpSe}} = \underset{c}{\text{argmax}} \left\{ \widehat{Se}(c) \cdot \widehat{Sp}(c) \right\}. \tag{7.7}$$

- *ROC01* (Metz 1978, Vermont et al. 1991): This method selects the cut-off point c that minimizes the distance between the ROC curve and the point (0,1), i.e.,

$$c^{\text{ROC01}} = \underset{c}{\text{argmin}} \left\{ (\widehat{Se}(c) - 1)^2 + (\widehat{Sp}(c) - 1)^2 \right\}. \tag{7.8}$$

- *MaxEfficiency* (Greiner 1995; 1996): This method selects the cut-off point c that maximizes the efficiency or, in other words, minimizes the error, i.e.,

$$c^{\text{MaxEfficiency}} = \underset{c}{\text{argmax}} \left\{ \widehat{p}_Y \widehat{Se}(c) + (1 - \widehat{p}_Y) \widehat{Sp}(c) \right\}, \tag{7.9}$$

  where $\widehat{p}_Y$ is the population probability of event estimated based on the sample and is calculated as follows:

$$\widehat{p}_Y = \frac{1}{n} \sum_{i \in S} I(y_i = 1). \tag{7.10}$$

7.2.2 Cut-off point estimation proposal with sampling weights

Although sensitivity and specificity parameters, as well as the population probability of event, can be estimated by expressions in eqs. (7.5) and (7.10) in any kind of data (including complex survey data), these expressions have been defined in a simple random sampling scenario, and we believe that the estimates obtained by

means of the above-mentioned formulas may be misleading for complex survey data and they should be pondered so that they incorporate the sampling weights in order to select more appropriate cut-off points. In this way, instead of the proportions of correct or incorrect classifications in sample S, they should be considered the proportions that these correctly or incorrectly classified observations represent in the finite population. For this reason, we propose to consider the sampling weights w_i to estimate sensitivity ($\widehat{Se}_w(c)$) and specificity ($\widehat{Sp}_w(c)$) parameters as previously defined in eq. (6.19), i.e.,

$$\widehat{Sp}_w(c) = \frac{\sum_{i_0 \in S_0} w_{i_0} \cdot I(\hat{p}_{i_0} < c)}{\sum_{i_0 \in S_0} w_{i_0}} \quad \text{and} \quad \widehat{Se}_w(c) = \frac{\sum_{i_1 \in S_1} w_{i_1} \cdot I(\hat{p}_{i_1} \geq c)}{\sum_{i_1 \in S_1} w_{i_1}}. \tag{7.11}$$

In addition, note that sampling weights should also be considered to estimate the probability of event in the population ($\widehat{p}_{Y,w}$), as previously defined in eq. (3.1), i.e.,

$$\hat{p}_{Y,w} = \frac{\sum_{i \in S} w_i \cdot I(y_i = 1)}{\sum_{i \in S} w_i}. \tag{7.12}$$

Therefore, we propose to estimate the optimal cut-off points based on the modified parameters of sensitivity ($\widehat{Se}_w(c)$), specificity ($\widehat{Sp}_w(c)$), and probability of event ($\widehat{p}_{Y,w}$) when working with complex survey data, i.e.:

$$c_w^{\text{Youden}} = \underset{c}{\text{argmax}} \left\{ \widehat{Se}_w(c) + \widehat{Sp}_w(c) - 1 \right\}, \tag{7.13}$$

$$c_w^{\text{MaxProdSpSe}} = \underset{c}{\text{argmax}} \left\{ \widehat{Se}_w(c) \cdot \widehat{Sp}_w(c) \right\}, \tag{7.14}$$

$$c_w^{\text{ROC01}} = \underset{c}{\text{argmin}} \left\{ (\widehat{Se}_w(c) - 1)^2 + (\widehat{Sp}_w(c) - 1)^2 \right\}, \tag{7.15}$$

$$c_w^{\text{MaxEfficiency}} = \underset{c}{\text{argmax}} \left\{ \widehat{p}_{Y,w} \widehat{Se}_w(c) + (1 - \widehat{p}_{Y,w}) \widehat{Sp}_w(c) \right\}. \tag{7.16}$$

7.3 Simulation study

This section describes the simulation process developed in the work presented in this chapter and the scenarios that have been drawn. The results obtained in this simulation study are also presented in this section.

As stated above, the aim of the work presented in this chapter of the dissertation is to study the influence of sampling weights in the estimation process of optimal

cut-off points for the methods described in Section 7.2. Since the decision of which optimal cut-off point estimation method to use in practice depends on the research of interest, the objective of this work is not to compare the behavior of the methods among them, but to compare the estimates that we obtain for each of these methods when sampling weights are considered (Section 7.2.2) or not (Section 7.2.1) in the estimation of sensitivity and specificity parameters.

In addition, we study the impact that the proposed estimators have in the estimation of the probability of event in the finite population. Therefore, a theoretical finite population is required, in which the response variable is known for all the units in the finite population. Thus, a simulation study was carried out based on ESIE survey data (described in Section 3.1), and the process followed to generate and sample the pseudo-population is explained in detail in Section 3.1.2. In particular, in this study, we limited to the establishments of the BC with at least 10 employees, and a total of four covariates have been considered (see Table 3.2): ownership (X_1), activity (X_2), number of employees (X_3) and province (X_4).

7.3.1 Scenarios and set up

Let U be the pseudo-population generated by following the steps described in Section 3.1.2 to which $\{(y_i, \boldsymbol{x}_i)\}_{i=1}^{N}$ are assigned. From this pseudo-population, a total of $R = 500$ samples have been obtained by one-stage stratification and the sampling weights have been calculated as explained in Section 3.1.2. The optimal cut-off points estimation methods that have been considered in this study are the ones described in Section 7.2, i.e., $m \in \{\text{Youden, MaxProdSpSe, ROC01, MaxEfficiency}\}$.

We describe below the simulation study set-up. For $r = 1, \ldots, R$:

Step 1. Draw a sample $S^r \subset U$ by one-stage stratification, mimicking the sampling process carried out for the real-life dataset as described in Section 3.1.2.

Step 2. Fit the logistic regression model to S^r and estimate $\hat{\boldsymbol{\beta}}^r$ by eq. (7.2).

Step 3. Estimate $\hat{p}_i^r$ by means of $\hat{\boldsymbol{\beta}}^r$ following eq. (7.1), $\forall i \in S^r$.

Step 4. Estimate the optimal cut-off points, $c^{m,r}$ (see eqs. (7.6), (7.7), (7.8), (7.9)) and $c_w^{m,r}$ (see eqs. (7.13), (7.14), (7.15), (7.16)) for each method m.

As mentioned above, the selection of the optimality criteria for selecting the cut-off points is based on the particular goal of each study. Therefore, our goal is not to compare the performance of the described methods between them. That is, the

aim is not to compare the performance of a method $m \in \{$Youden, MaxProdSpSe, ROC01, MaxEfficiency$\}$, to the rest of the methods, but to compare the cut-off points selected by means of the method m when sampling weights are considered or not in the estimation process. Thus, we define the difference and absolute difference between weighted and unweighted cut-off points as follows:

$$\text{diff}^{\,m,r} = c^{m,r} - c_w^{m,r} \quad \text{and} \quad \text{abs.diff}^{\,m,r} = |c^{m,r} - c_w^{m,r}|\,. \tag{7.17}$$

In addition, we would also like to regard the impact that the decision to select weighted or unweighted optimal cut-off points may have in the classification of all the units in the finite population. Thus, we continue with the simulation study as follows:

Step 5. Calculate $\hat{p}_i^r$ by means of $\hat{\boldsymbol{\beta}}^r$ (estimated in **Step 2.**) following eq. (7.1), $\forall i \in U$.

Step 6. Classify each unit as event or non-event based on $\hat{p}_i^r$, $\forall i \in U$ and the selected cut-off points. That is, we define two estimated responses ($\hat{y}_i^{m,r}$ and $\hat{y}_{w,i}^{m,r}$) for each unit based on the cut-off points $c^{m,r}$ and $c_w^{m,r}$ (selected in **Step 4.**) as in eq. (7.18). For each method m and $\forall i \in U$:

$$\hat{y}_i^{m,r} = \begin{cases} 1 & \text{if} \quad \hat{p}_i^r \geq c^{m,r}, \\ 0 & \text{if} \quad \hat{p}_i^r < c^{m,r}, \end{cases} \quad \text{and} \quad \hat{y}_{w,i}^{m,r} = \begin{cases} 1 & \text{if} \quad \hat{p}_i^r \geq c_w^{m,r}, \\ 0 & \text{if} \quad \hat{p}_i^r < c_w^{m,r}. \end{cases} \tag{7.18}$$

Finally, in order to account for the error that may be introduced in the classification of the units in the finite population by the selected optimal cut-off points, one more parameter is defined. The error is estimated by comparing the population probability of event estimated by means of the estimated responses (defined in **Step 6**) to the true probability of event in the finite population considering the true values of the response variable for the units in the pseudo-population, y_i, $\forall i \in U$. For this purpose, we split the finite population U in K disjointed subsets of the same size where $U = \bigcup_{k=1}^{K} U_k$. We repeat this process $\mathcal{T} = 10$ times, where $\forall \mathfrak{t} \in \{1, \ldots, \mathcal{T}\}$, $U = \bigcup_{k=1}^{K} U_k^{\mathfrak{t}}$. In this way, we get $\mathcal{T} \times K$ subsets from U and the population probability of event will be estimated in each one of these subsets. Let $N_k^{\mathfrak{t}}$ indicate the number of units in the subset $U_k^{\mathfrak{t}}$, $\forall k = 1, \ldots, K$ and $\forall \mathfrak{t} = 1, \ldots, \mathcal{T}$.

We denote as the global mean squared error (GMSE) of the population proba-

bility of event with $\mathcal{T} = 10$ replicates the following parameters:

$$\begin{aligned} \text{GMSE}^{m,r}_{(K)} &= \frac{1}{\mathcal{T} \times K} \sum_{\mathfrak{t}=1}^{\mathcal{T}} \sum_{k=1}^{K} \left(\frac{1}{N_k^{\mathfrak{t}}} \sum_{i \in U_k^{\mathfrak{t}}} \hat{y}_i^{m,r} - \frac{1}{N_k^{\mathfrak{t}}} \sum_{i \in U_k^{\mathfrak{t}}} y_i \right)^2, \\ \text{GMSE}^{m,r}_{w,(K)} &= \frac{1}{\mathcal{T} \times K} \sum_{\mathfrak{t}=1}^{\mathcal{T}} \sum_{k=1}^{K} \left(\frac{1}{N_k^{\mathfrak{t}}} \sum_{i \in U_k^{\mathfrak{t}}} \hat{y}_{w,i}^{m,r} - \frac{1}{N_k^{\mathfrak{t}}} \sum_{i \in U_k^{\mathfrak{t}}} y_i \right)^2, \end{aligned} \tag{7.19}$$

Different number of subsets have been selected in order to evaluate the impact that the sample size of each subset may have: $K \in \{1, 10, 100, 500\}$. In addition, we considered the GMSE evaluated considering the H strata as the subsets where $U_h, \forall h = 1, \ldots, H$ indicates the subset corresponding to stratum h and $U = \bigcup_{h=1}^{H} U_h$, i.e.,

$$\begin{aligned} \text{GMSE}^{m,r}_{(H)} &= \frac{1}{H} \sum_{h=1}^{H} \left(\frac{1}{N_h} \sum_{i \in U_h} \hat{y}_i^{m,r} - \frac{1}{N_h} \sum_{i \in U_h} y_i \right)^2, \\ \text{GMSE}^{m,r}_{w,(H)} &= \frac{1}{H} \sum_{h=1}^{H} \left(\frac{1}{N_h} \sum_{i \in U_h} \hat{y}_{w,i}^{m,r} - \frac{1}{N_h} \sum_{i \in U_h} y_i \right)^2. \end{aligned} \tag{7.20}$$

This simulation study has been carried out by means of the statistical software R. In particular, some functions of the R package `OptimalCutpoints` (López-Ratón et al. 2014) have been modified in order to incorporate an argument that provides us with the option to consider sampling weights in the estimation process of the optimal cut-off points for the described methods.

7.3.2 Results

In this section, we show the results obtained in the simulation study described in Section 7.3.1. Figures 7.1, 7.2, 7.3 and 7.4 depict the box-plots of unweighted and weighted estimates of the optimal cut-off points and the results of the parameters diff and GMSE (see eqs. (7.17), (7.19) and (7.20)) for Youden, MaxProdSpSe, ROC01 and MaxEfficiency methods, respectively. Numerical results of the simulation study are summarized in Table 7.1.

In general, except for the MaxEfficiency method, the results suggest that the optimal cut-off point estimates differ when sampling weights are ignored or considered in the estimation process. The difference has always been positive (i.e. the unweighted estimates have been greater than the weighted ones), except in the MaxEfficiency method, where both positive and negative differences have been observed. For this reason, the mean and standard deviation of the difference and absolute dif-

ference parameters are equal for all the methods except for MaxEfficiency (see Table 7.1). The error generated and accounted in terms of GMSE described in eq. (7.19) decreases considerably when sampling weights are taken into account. In addition, similar results have been obtained for different $K \in \{1, 10, 100, 500\}$ values, which indicates that the difference between estimated and true prevalence in the whole finite population is similar to that in smaller homogeneous subsets. However, it could be observed that the average of GMSE becomes slightly greater as the number of subsets K increases (for both, weighted and unweighted estimates), indicating that the differences between the estimated and true prevalence tend to be a little bit greater in smaller subsets. When considering the strata as non-homogeneous subsets defined by the H strata of the population, the GMSE obtained as described in eq. (7.20) with the weighted estimates is still smaller than with the unweighted ones. However, the difference between weighted and unweighted GMSE is slightly smaller for the non-homogenous partition than for homogeneous partitions. We believe that the reason is that the difference obtained between estimated and true prevalence differs depending on the number of individuals sampled in each strata, being increased in very small strata. Note that if the population size of a particular stratum is 1, then the error in this stratum is 0 (if the unit is classified correctly) or 1 (otherwise). This is not common when working with homogeneous strata where, in all the randomly selected subsets, the difference between estimated and true prevalence seems to be similar (results not shown). In addition, note that even though strata are of different sizes, the stratum size is not taken into account when computing the GMSE parameter. The behavior of each of the methods that have been studied throughout this work will be analyzed one by one below.

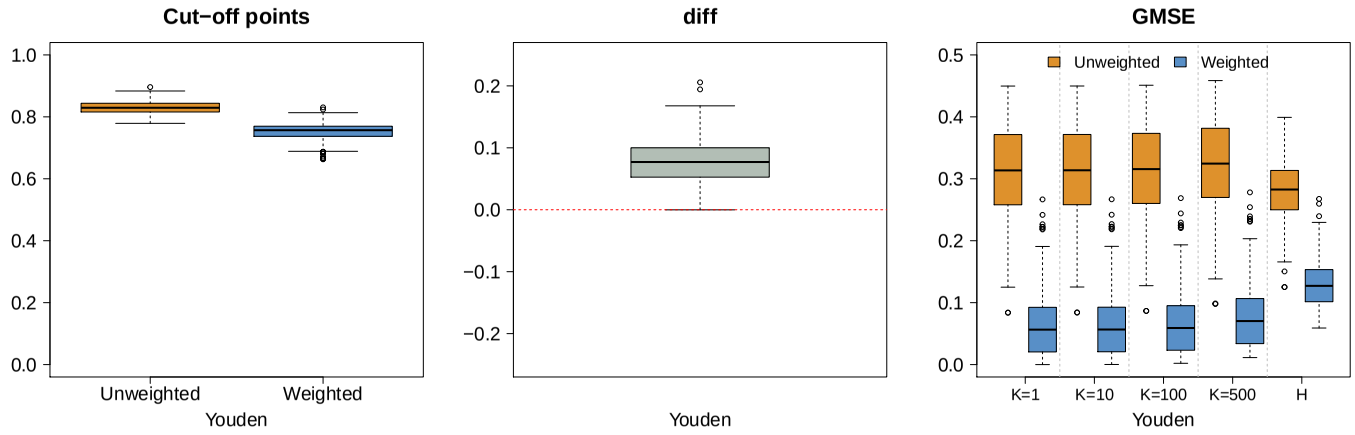
The optimal cut-off point estimated by the Youden method in this simulation study, is 0.830 on average when sampling weights are not taken into account while the weighted estimates are smaller on average (0.752), with standard deviations of 0.021 and 0.028, respectively. The difference between the unweighted and weighted estimates is on average 0.078 with a standard deviation of 0.034 (see Figure 7.1). In terms of GMSE, the error produced by means of the weighted estimates in the finite population is more or less 5 times smaller than the error produced by means of the unweighted estimates on average. The standard deviation is also smaller for the weighted estimates. When the GMSE is computed over the $H = 325$ strata, the GMSE turns out to be 0.130 and 0.281, for weighted and unweighted estimates, respectively.

The unweighted estimates obtained by the MaxProdSpSe method are again

greater than the weighted ones, being on average 0.812 and 0.753, respectively (see Figure 7.2). The difference between those estimates is 0.058 on average with a standard deviation of 0.019. GMSE becomes again 5 times smaller when sampling weights are considered in the estimation process and the standard deviation of the weighted estimates is half of that of the unweighted ones. The GMSE measured over the different strata for weighted and unweighted estimates is 0.126 and 0.243, respectively.

For the ROC01 method, weighted estimates are also lower than the unweighted ones (0.753 and 0.808 on average, respectively) and the standard deviations are slightly greater (0.017 and 0.015, respectively) (see Figure 7.3 and Table 7.1). The error generated by the weighted estimates in the finite population is lower than the error produced by the unweighted estimates in terms of GMSE.

Finally, in contrast to the results obtained by the rest of the methods, for the MaxEfficiency method no significant differences are observed among the unweighted and weighted estimates. Optimal cut-off point estimates throughout the $R = 500$ samples are quite similar in terms of mean and standard deviation. The average of the unweighted estimates is 0.511, while for the weighted estimates the average is 0.530. In particular, in more than 50% of the cases, the difference between weighted and weighted estimates is 0. The difference in the error produced by those estimates in the finite population is also negligible. For $K = 1$, for example, the GMSE produced by the unweighted estimates is on average 0.048 with a standard deviation of 0.013, while the average GMSE of the weighted estimates is 0.045 with a standard deviation of 0.014. The GMSE calculated over the $H = 325$ strata is 0.070 for weighted estimates and 0.071 for unweighted estimates.



(a) (b) (c)

Figure 7.1: Box-plots of the results obtained for the Youden method across $R = 500$ samples: (a) unweighted and weighted estimates of the optimal cut-off points, (b) differences between unweighted and weighted estimates (diff), and (c) GMSE produced by the unweighted and weighted estimates for $K \in \{1, 10, 100, 500\}$ and H.

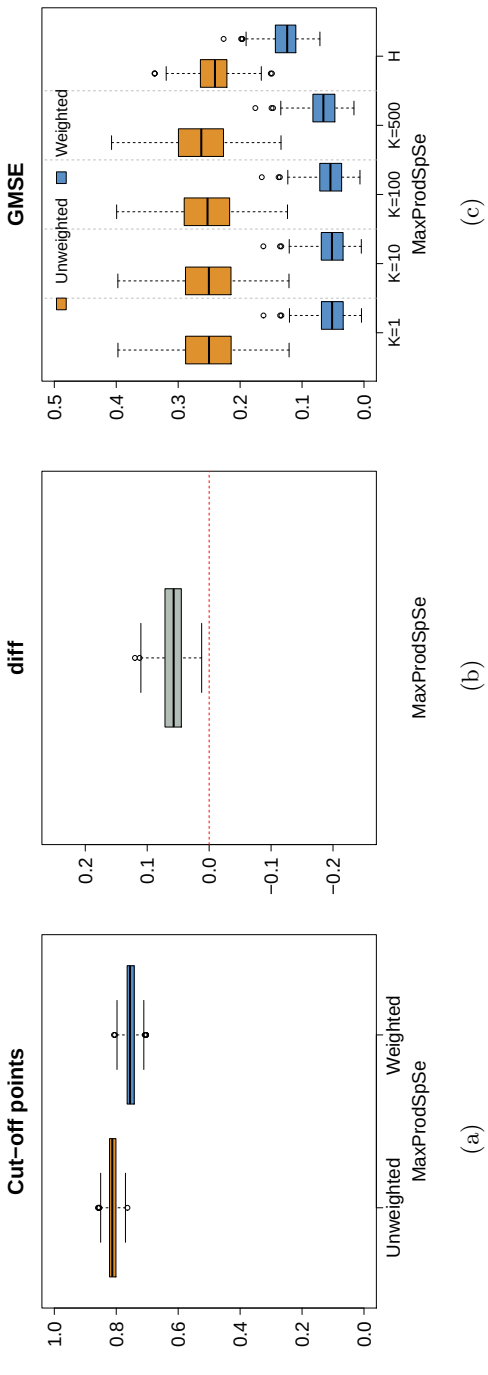
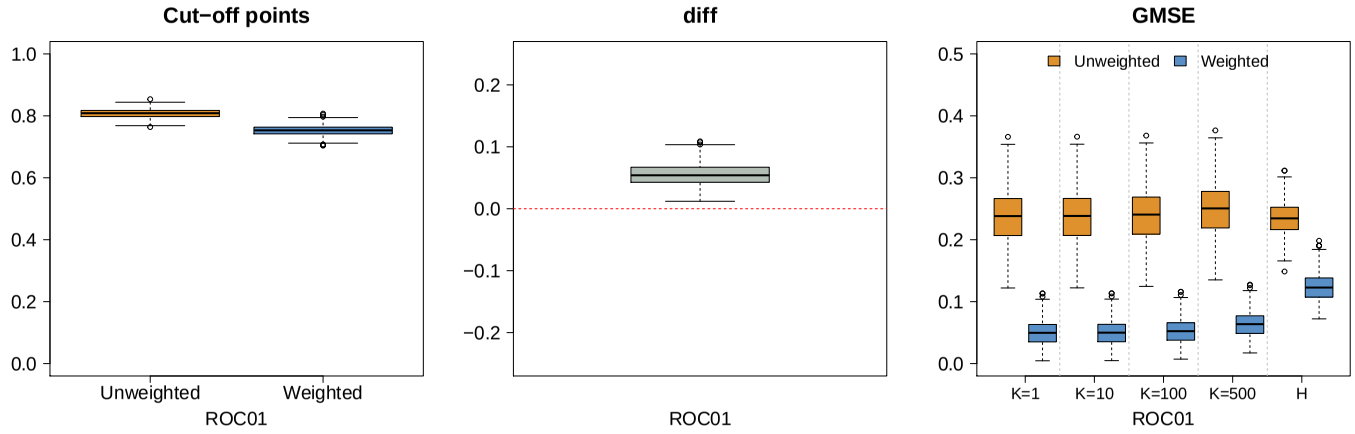
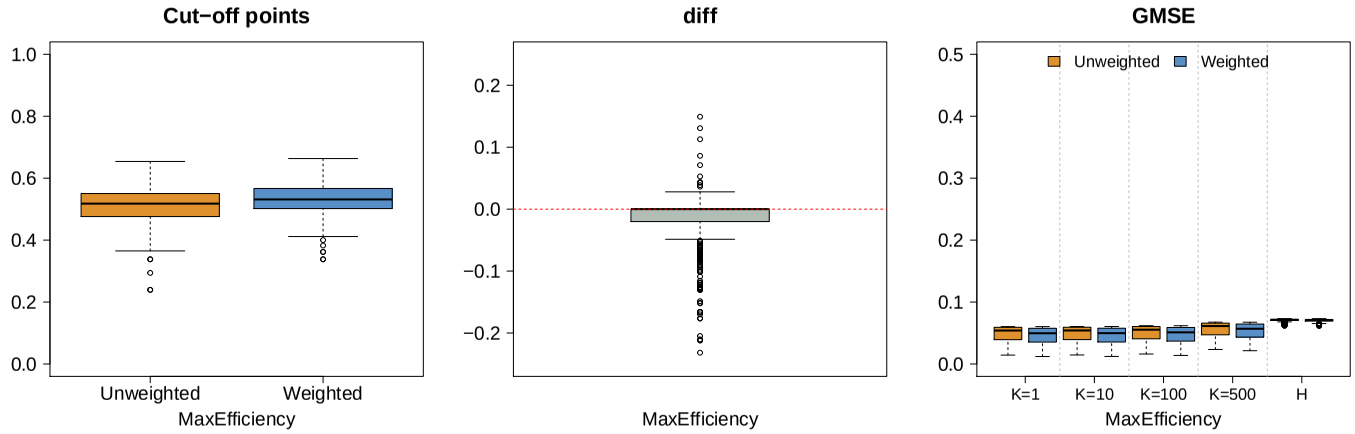


Figure 7.2: Box-plots of the results obtained for the MaxProdSpSe method across $R = 500$ samples: (a) unweighted and weighted estimates of the optimal cut-off points, (b) differences between unweighted and weighted estimates (diff), and (c) GMSE produced by the unweighted and weighted estimates for $K \in \{1, 10, 100, 500\}$ and H.



(a) (b) (c)

Figure 7.3: Box-plots of the results obtained for the ROC01 method across $R = 500$ samples: (a) unweighted and weighted estimates of the optimal cut-off points, (b) differences between unweighted and weighted estimates (diff), and (c) GMSE produced by the unweighted and weighted estimates for $K \in \{1, 10, 100, 500\}$ and H.



(a) (b) (c)

Figure 7.4: Box-plots of the results obtained for the MaxEfficiency method across $R = 500$ samples: (a) unweighted and weighted estimates of the optimal cut-off points, (b) differences between unweighted and weighted estimates (diff), and (c) GMSE produced by the unweighted and weighted estimates for $K \in \{1, 10, 100, 500\}$ and H.

Table 7.1: Average (mean) and standard deviation (sd) of the a) unweighted and weighted optimal cut-off points, b) difference (diff) and absolute difference (abs.diff) among them and, c) GMSE produced by the unweighted and weighted optimal cut-off points when classifying units in the finite population for $K \in \{1, 10, 100, 500\}$ and H across $R = 500$ samples for all the methods considered.

		Youden	**MaxProdSpSe**	**ROC01**	**MaxEfficiency**
		Mean (sd)	**Mean (sd)**	**Mean (sd)**	**Mean (sd)**
Cut-off	**unw**	0.830 (0.021)	0.812 (0.016)	0.808 (0.015)	0.511 (0.058)
points	w	0.752 (0.028)	0.753 (0.018)	0.753 (0.017)	0.530 (0.052)
diff		0.078 (0.034)	0.058 (0.019)	0.055 (0.017)	-0.019 (0.046)
abs.diff		0.078 (0.034)	0.058 (0.019)	0.055 (0.017)	0.023 (0.044)
GMSE	**unw**	0.311 (0.075)	0.251 (0.053)	0.237 (0.044)	0.048 (0.013)
(K=1)	w	0.063 (0.050)	0.053 (0.024)	0.051 (0.020)	0.045 (0.014)
GMSE	**unw**	0.311 (0.076)	0.251 (0.054)	0.237 (0.046)	0.048 (0.014)
(K=10)	w	0.063 (0.051)	0.053 (0.025)	0.051 (0.021)	0.046 (0.014)
GMSE	**unw**	0.313 (0.090)	0.253 (0.071)	0.239 (0.064)	0.050 (0.021)
(K=100)	w	0.066 (0.057)	0.056 (0.034)	0.053 (0.031)	0.047 (0.021)
GMSE	**unw**	0.322 (0.136)	0.263 (0.120)	0.249 (0.115)	0.056 (0.042)
(K=500)	w	0.076 (0.079)	0.067 (0.062)	0.064 (0.059)	0.053 (0.041)
GMSE	**unw**	0.281 (0.047)	0.243 (0.033)	0.234 (0.027)	0.071 (0.002)
(H)	w	0.130 (0.038)	0.126 (0.025)	0.125 (0.024)	0.070 (0.003)

7.3.3 Analyzing the performance of MaxEfficiency

In the results shown and discussed in Section 7.3.2, we have pointed out that the MaxEfficiency method is the only one by which we obtain similar weighted and unweighted optimal cut-off point estimates. Therefore, in this section, we aim to keep an eye on this method and further analyze whether this behavior is particular to the simulated scenario, or in contrast, the similar performance of the weighted and unweighted estimates in this particular case can be explained by some mathematical properties of the method itself. In particular, we aim to know whether there can be any scenario in which there might be differences between unweighted and weighted optimal cut-off points estimated by means of the MaxEfficiency method.

In this method, the efficiency function, which can be defined as in eq. (7.21), is maximized:

$$Ef(c) = p_Y Se(c) + (1 - p_Y)Sp(c), \tag{7.21}$$

where $p_Y = P(Y = 1)$. Hence, note that the efficiency function is a weighted average of the sensitivity and specificity parameters, where the "importance" or "ponderation" of each parameter is determined by means of the probability of event in the population, p_Y. In particular, sensitivity is a decreasingly monotonous function; that is, the greater the value of c, the lower or equal the sensitivity. In contrast, specificity is increasingly monotonous by definition (see, eq. (7.4)). The efficiency, being a weighted average of them, is not necessarily a monotonous function, but note that the greater the probability of event in the population, p_Y, it will tend to be more similar to the sensitivity curve, while the lower the probability of event, it will tend to be more similar to the specificity curve. Figure 7.5 is displayed below for illustration purposes.

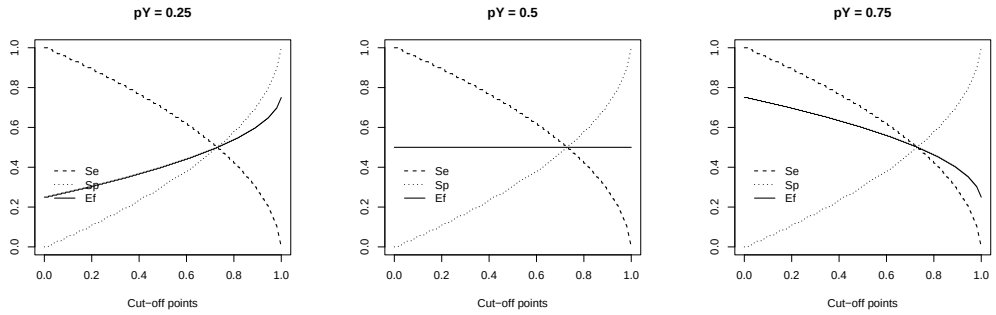


Figure 7.5: Example of the efficiency function for different values of $p_Y = P(Y = 1)$: $p_Y = 0.25$ (left), $p_Y = 0.5$ (center), $p_Y = 0.75$ (right).

In practice, unweighted estimates of the population probability of event ($\widehat{p}_Y$) and sensitivity ($\widehat{Se}(c)$) and specificity ($\widehat{Sp}(c)$) are obtained as defined in eqs. (7.10) and (7.5), respectively, while the corresponding weighted estimates ($\widehat{p}_{Y,w}$, $\widehat{Se}_w$ and $\widehat{Sp}_w$) as in eqs. (7.12) and (7.11). In this simulation study, considering all the $R = 500$ samples, the unweighted estimate of the population probability of event (see eq. (7.10)) is, on average, 0.833 (with a standard deviation of 0.005), and hence, the unweighted estimate of the efficiency function will be more similar to the $\widehat{Se}(c), \forall c \in (-\infty, \infty)$ function. Similarly, regarding the weighted estimates, the weighted estimate of the population probability of event (see eq. (7.12)) is on average 0.756, with a standard deviation of 0.014, and hence, the efficiency function will be closer to the $\widehat{Se}_w(c), \forall c \in (-\infty, \infty)$ function. Hence, both the weighted and unweighted efficiency functions are more similar to monotonously decreasing functions. Thus, it seems reasonable to think they might take their maximum values for

relatively low cut-off points. To illustrate these explanations, we depict Figure 7.6, corresponding to the sample $r = 100$, in which the unweighted and weighted sensitivity, specificity and efficiency functions are depicted, jointly with the corresponding optimal cut-off point estimates.

Nevertheless, what if the estimated weighted and unweighted probabilities of event differ more considerably, and one of the efficiency functions tends to be more similar to the specificity (that is, to an increasingly monotonous function)? Then, the efficiency function would be expected to take maximum values for greater cut-off points. In order to analyze this situation, we artificially manipulate the original sampling weights, as if the samples were obtained from a different population than the simulated one. In particular, the new sampling weights for $\forall i \in S$ have been defined as in eq. (7.22):

$$\begin{aligned} w'_{i_0} &= 5 \cdot w_{i_0}, \quad \forall i_0 \in S_0, \\ w'_{i_1} &= w_{i_1}, \quad \forall i_1 \in S_1. \end{aligned} \tag{7.22}$$

In this way, the weighted estimate of the population probability of event ($\hat{p}_{Y,w}$ as defined in eq. (7.12)) of the new samples is on average 0.382 (with a standard deviation of 0.018), being the corresponding efficiency functions more similar to the monotonously increasing function specificity. We depict in Figure 7.7 the example of the sample $r = 100$, in which can be observed that the situation has changed considerably, and the weighted and unweighted optimal cut-off points differ in this case.

The boxplots corresponding to the unweighted and weighted estimates of the optimal cut-off points across the $R = 500$ samples are depicted in Figure 7.8, in which it can be observed that in this scenario, we obtain different unweighted and weighted optimal cut-off point estimates for the MaxEfficiency method. The average of the optimal cut-off point values estimated with the unweighted method is 0.175 with a standard deviation of 0.035, while the optimal cut-off point value obtained by means of the weighted method is 0.491 on average with a standard deviation of 0.044. Note that we are not able to calculate the GMSE in this scenario, given that the samples have been manipulated by changing the original weights, and hence, the corresponding finite population does not exist. However, considering the results obtained for the rest of the methods, as well as regarding the results obtained in the previous chapter, we believe that the weighted optimal cut-off point estimates will probably lead us to classify the individuals more properly.

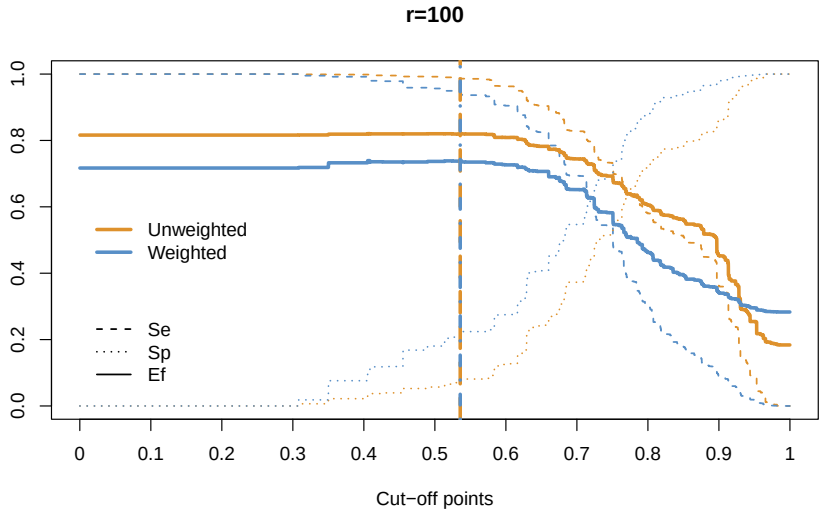


Figure 7.6: Unweighted (in orange color) and weighted (in blue color) estimates of the sensitivity (dashed line), specificity (dotted line), and efficiency (solid line), jointly with weighted and unweighted optimal cut-off point estimates for the sample $r = 100$ in the original scenario.

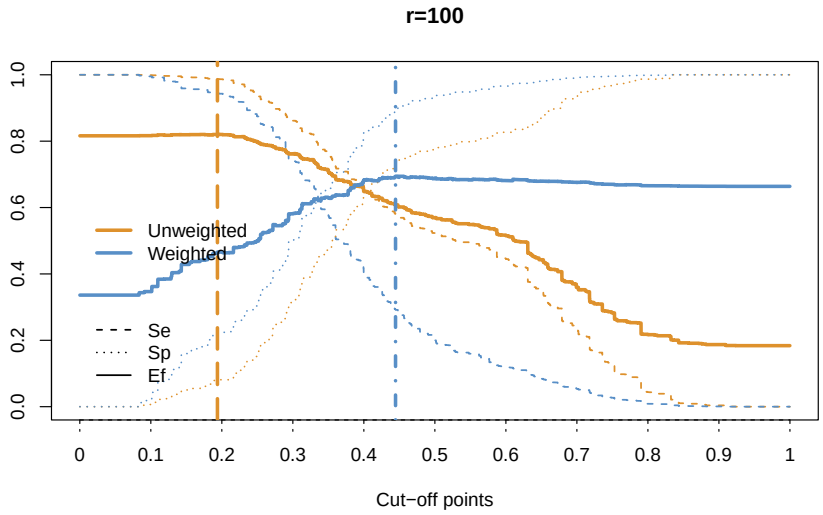


Figure 7.7: Unweighted (in orange color) and weighted (in blue color) estimates of the sensitivity (dashed line), specificity (dotted line), and efficiency (solid line) across all the possible cut-off points, jointly with weighted and unweighted optimal cut-off point estimates for the sample $r = 100$ in the new scenario with manipulated sampling weights.

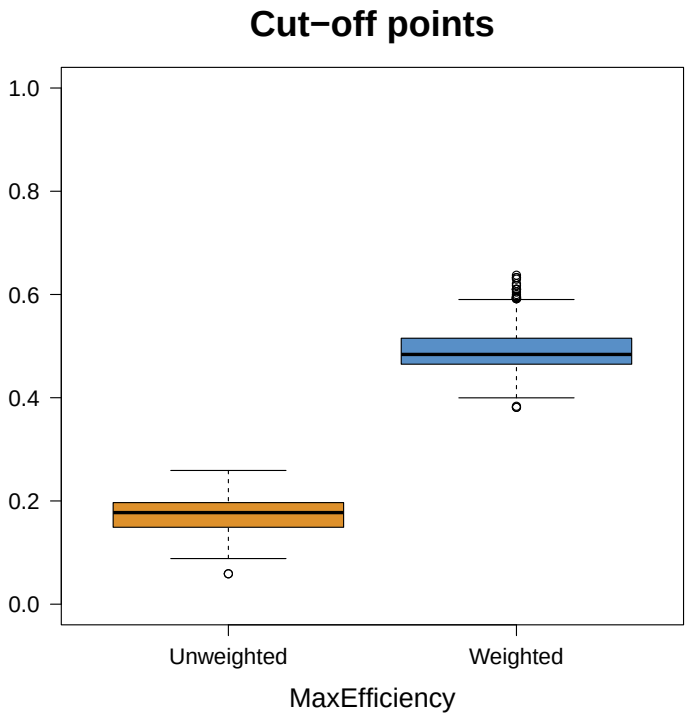


Figure 7.8: Boxplots of the unweighted and weighted estimates of the optimal cut-off points obtained based on the MaxEfficiency method across $R = 500$ samples in the new scenario with manipulated sampling weights.

7.4 Application to ESIE survey data

The methodology proposed in Section 7.2 could also be applied to real-world surveys. In particular, for illustration purposes, we have applied this methodology to the real ESIE survey data described in Section 3.1.

As explained at the beginning of Section 7.3, in the simulation study of this work, as well as in the application, establishments with at least 10 employees of the BC have been considered, and four categorical variables have been used as predictors: ownership (X_1), activity (X_2), number of employees (X_3) and province (X_4). In this way, a logistic regression model was fitted to the sample considering these four covariates, the regression coefficients were estimated by maximizing the pseudo-likelihood function in eq. (7.2) and for all the sampled units $\forall i' \in \mathcal{S}$, $\hat{p}_{i'}$ were obtained.

We have applied the methods described in Section 7.2 for the selection of optimal

cut-off points, which have been estimated by both, ignoring and considering sampling weights. The results are shown in Table 7.2. It can be observed that the unweighted and weighted estimates differ when Youden, MaxProdSpSe, and ROC01 methods are applied, which is in line with the results obtained in the simulation study. In particular, the unweighted estimates are greater than the weighted estimates, which are similar to the ones observed in Section 7.3.2 (see Table 7.1). The unweighted and weighted estimates obtained by means of the MaxEfficiency method are equal, which is also in line with the results observed in the simulation study. Those estimates obtained by the MaxEfficiency method are lower than the average of the estimates obtained in the simulation study. However, it should be noted that this may be justified by the large standard deviation observed previously for the cut-off points estimated by means of the MaxEfficiency method (see Figure 7.4 and Table 7.1), as well as due to the differences between the real survey and simulated data.

Table 7.2: Optimal cut-off point estimates obtained by means of Youden, MaxProdSpSe, ROC01, and MaxEfficiency methods, considering or not the sampling weights.

	Youden	**MaxProdSpSe**	**ROC01**	**MaxEfficiency**
Unweighted	0.800	0.800	0.800	0.388
Weighted	0.752	0.752	0.747	0.388

7.5 Discussion

In this work, new estimators have been proposed for estimating optimal cut-off points of the probability of event in the logistic regression framework considering sampling weights in the estimation process. In particular, we have focused on data derived from complex sampling designs. For this purpose, four optimal cut-off point estimation methods (which are denoted as Youden, MaxProdSpSe, ROC01, and MaxEfficiency (López-Ratón et al. 2014)) have been selected and modified in order to incorporate sampling weights in the estimation process. These four methods have been selected for being the ones most commonly applied in the literature. In particular, the so widely used `pROC` package in R (Robin et al. 2011) has incorporated the Youden and ROC01 methods for the estimation of optimal cut-off points. All these methods are based on different optimality criteria that are related to sensitivity and specificity parameters. Therefore, we propose a methodology for considering sampling weights in the estimation process of sensitivity and specificity parameters,

as well as in the estimation of the probability of event, in order to estimate optimal cut-off points based on these parameters by taking into account the sampling weights. A simulation study has been carried out in order to analyze the behavior of both methodologies by comparing the optimal cut-off point estimates obtained by means of the above-mentioned methods when sampling weights are considered or ignored in the estimation process. The error that those estimates generate in the estimation of the probability of event in the finite population has also been analyzed in this simulation study. In particular, we considered the GMSE in order to evaluate the behavior of the estimated probability of event in the population (once the cut-off point and the response for each unit based on it were estimated) by comparing it to the true probability of event in the finite population.

We also considered it interesting to study the differences in estimating sensitivity and specificity based on the cut-off points estimated with and without sampling weights. We have observed that the differences are in line with those observed when studying the GMSE (results not shown).

In general, the results suggest the convenience of incorporating sampling weights into the estimation process of optimal cut-off points. For three out of the four methods studied, estimates obtained differ depending on whether the sampling weights were considered or not. Furthermore, it can be observed that the error in the estimates of the response variable obtained by taking into account sampling weights is much smaller than that generated by the estimates obtained by ignoring them for the units in the finite population. Although the cut-off point estimates may not seem very different from each other in some cases, it is observed that the effect of applying one or the other estimate for the classification of units in the population is considerable. In our opinion, the reason for this is that a large amount of individuals of the finite population (specifically, more than 20% of all the units on average) has estimated probabilities of event that range in the interval defined by the unweighted and weighted estimates and thus, choosing the unweighted cut-off point leads to misclassify a larger number of units in the finite population.

Nevertheless, the results related to the MaxEfficiency method appear to be different compared to Youden, MaxProdSpSe and ROC01. In general, in the results obtained using this method, there are no great differences between the estimates obtained by ignoring or considering the sampling weights, and furthermore, in most cases, the two estimates coincide. Therefore, the errors generated in the population by these estimates are also similar, and there are no significant differences between them. Hence, we can say that, at least under the scenario we have worked on, there

is no difference between the unweighted and weighted estimates obtained by the MaxEfficiency method. However, as discussed in Section 7.3.3, we believe that this could be due to a particular characteristic of the scenario in which we have worked and not a specific property of the method itself, as we have seen in the scenario considered by modifying the sampling weights in Section 7.3.3. It should be mentioned that, in that section, we analyze the effect of the MaxEfficiency in other situations for illustration purposes. Nevertheless, studying the mathematical properties of this behavior is part of further research. It should also be stated that, in this illustration, we have not been able to analyze the differences in the performance of weighted and unweighted cut-off point estimates, which should be analyzed based on another simulation study. However, due to the results obtained for the rest of the methods, we believe that selecting weighted cut-off points would also be preferable for the MaxEfficiency method.

Finally, we would like to comment on the limitations of this study. First of all, it should be noted that we have conducted this simulation study based on real survey data. Therefore, the effect that the sampling technique chosen may have on the differences between weighted and unweighted optimal cut-off point estimates remains to be studied as further work. For example, it should be mentioned that in this study, we have only analyzed the effect of the sampling weights obtained by means of one-stage stratification. Data derived from other sampling techniques, such as the two-stage cluster sampling design, have not been considered. The performance needs to be analyzed, but the proposed estimators are also valid and can be applied to two-stage sampling designs. It would also be interesting to study the behavior of the studied methods under uninformative complex sampling designs to analyze whether there are any differences in the estimates or variability of the weighted and unweighted methods. To answer these questions concerning the sampling designs, the simulation study carried out in Chapter 6 could be easily extended to the framework of the estimation of optimal cut-off points (further work). Secondly, it would be interesting to analyze and compare the behavior of the methods that have been studied throughout this document in different scenarios, for instance, with different probability of event values. Nevertheless, it should be noted that as the simulation study we have depicted is based on a real survey, the probability of event of the scenario we have analyzed was also described by the observed data.

In conclusion, in this work, we have implemented four of the most commonly used optimal cut-off point estimation methods, which are implemented in diverse software. In the simulation study considered, out of these four methods, in three of them, the

use of sampling weights highly improves the results, while in the fourth, the results do not differ whether you use the sampling weights or not (hence, we neither lose anything if sampling weights are considered in the estimation process). However, we have also seen that, even in this fourth method, weighted and unweighted optimal cut-off point estimates may differ in some scenarios. Therefore, our recommendation is to incorporate the sampling weights in the estimation process of optimal cut-off points when working with data derived from complex sampling designs. However, if one is interested in applying other methods different from those studied throughout this paper, it should be considered whether it is appropriate or not to use sampling weights and, if so, how to consider them in each particular case. The optimal cut-off point estimation methods that consider sampling weights proposed in this work have been incorporated into the `wROC` package available on GitHub and is explained in detail in Chapter 8.

CHAPTER 8

Software development

In the previous chapters of this dissertation, new methodological approaches have been proposed and described to improve the development of prediction models when working with complex survey data. Two R-packages have been developed, which incorporate these methodological proposals. In this chapter, we explain in detail these two R-packages: `wlasso` and `wROC`.

In particular, the R-package `wlasso` has been developed to incorporate the methodological proposal to fit LASSO regression models to complex survey data, described in Chapter 5. The purpose of this package is two-fold: on the one hand, it offers the possibility to create partially independent training and test sets of a data set by means of replicate weights methods described in Section 5.2.3; on the other hand, it allows to fit LASSO regression models considering sampling weights and select the tuning parameter that minimizes the error based on replicate weights methods, as described throughout Chapter 5, and in particular, in Sections 5.2.2 and 5.2.3. All the documentation related to this package is available in Section 8.1.

In addition, the methodological proposals described in detail in Chapters 6 and

7 to estimate the ROC curve and AUC of logistic regression models with complex survey data, as well as to estimate optimal cut-off points for individual classification, are implemented in the R-package `wROC`. This package provides the functions needed to estimate sensitivity and specificity parameters, optimal cut-off points based on these parameters by means of the methods described in Section 7.2.2, and the ROC curve and AUC of logistic regression models fitted to complex survey data considering sampling weights as described in Section 6.2.2. The functions incorporated into the package and the arguments needed are described in detail in Section 8.2.

Both packages are thought to be easy to implement for survey statisticians and professionals handling the analysis of complex survey data in their daily practice. The most well-known R-package in this context is the `survey` package (Lumley 2020), and hence, all the functions of these new packages `wlasso` and `wROC` allow (among other options) to indicate the complex survey designs with `survey.design` objects generated by means of the function `survey::svydesign()`.

At the moment of writing this dissertation, both packages are uniquely available on two different repositories of GitHub. However, both of them will be submitted to the official R-packages repository (i.e., the Comprehensive R Archive Network (CRAN)), in brief. It should also be noted that these packages are dynamic packages that are and will continue to be constantly improved, either by user requests or by the incorporation of new methodological proposals developed by the authors.

8.1 `wlasso` R-package

The main goal of the R-package `wlasso` is to provide tools to fit LASSO regression models for complex survey data. The methodological proposals implemented in this package have been described in detail in Chapter 5. This package is available on GitHub: https://github.com/aiparragirre/wlasso.

In particular, there are two ways to install this package in R. On the one hand, the whole GitHub repository can be downloaded and the package can manually be installed in R. On the other hand, we can also run the following code to install the package directly without the need to download it manually:

```
library("devtools")
install_github("aiparragirre/wlasso/wlasso")
library(wlasso)
```

Three main functions are available in this R-package. Table 8.1 contains a summary of these functions and their main purposes.

Specifically, the function `replicate.weights()` generates new training and test subsets, considering the complex sampling design followed to obtain the original sample, by means of any of the replicate weights methods described in Section 5.2.3. All the details concerning this function are described in Section 8.1.1.

The main goal of the function `wlasso()` is to fit weighted LASSO regression models that consider complex sampling designs as described in Sections 5.2.2 and 5.2.3. This function fits LASSO regression models for a grid of tuning parameters and indicates which of the parameters minimizes the error. The function `replicate.weights()` is used to define training and test sets for this purpose. This function is described in detail in Section 8.1.2.

Finally, the goal of the function `wlasso.plot()`, which is described in Section 8.1.3, is to provide a graphical summary of the results obtained by means of the models fitted using the function `wlasso()`.

Table 8.1: Summary of the functions available in the R-package `wlasso`, a brief description of them, and the corresponding references to the sections in which the implemented methodology has been described.

Function	**Description**	**Methods**
`replicate.weights()`	Define training and test sets of the original sample with replicate weights.	Section 5.2.3
`wlasso()`	Fit LASSO models for complex survey data.	Sections 5.2.2 and 5.2.3
`wlasso.plot()`	Graphical visualization of the error of the fitted LASSO models.	

In addition, a simulated data set has been incorporated into the package for illustration purposes. All the examples set to define the usage of the functions described in Sections 8.1.1, 8.1.2 and 8.1.3 use this data set so that the readers and package users can reproduce them. Users that install the R-package `wlasso` can access this data set by running the following code:

```
data(simdata_lasso_binomial)
```

The data set `simdata_lasso_binomial` contains information about a total of 50 covariates (denoted as `x.1`, ..., `x.50`), a dichotomous response variable (`y`), columns

indicating the stratum and cluster to which each unit belongs (`strata` and `cluster`, respectively), and a column indicating the sampling weight (`weights`) assigned to each of 1 720 observations in the data set.

```
names(simdata_lasso_binomial)
```

```
 [1] "x.1"     "x.2"     "x.3"     "x.4"     "x.5"     "x.6"
 [7] "x.7"     "x.8"     "x.9"     "x.10"    "x.11"    "x.12"
[13] "x.13"    "x.14"    "x.15"    "x.16"    "x.17"    "x.18"
[19] "x.19"    "x.20"    "x.21"    "x.22"    "x.23"    "x.24"
[25] "x.25"    "x.26"    "x.27"    "x.28"    "x.29"    "x.30"
[31] "x.31"    "x.32"    "x.33"    "x.34"    "x.35"    "x.36"
[37] "x.37"    "x.38"    "x.39"    "x.40"    "x.41"    "x.42"
[43] "x.43"    "x.44"    "x.45"    "x.46"    "x.47"    "x.48"
[49] "x.49"    "x.50"    "strata"  "cluster" "y"       "weights"
```

In the following sections, we proceed to describe in detail the usage of each of the functions available in the package `wlasso`.

8.1.1 `replicate.weights()` function

The function `replicate.weights()` allows defining new training and test sets by means of the replicate weights methods described in Section 5.2.3. In this way, the training and test sets properly represent the finite population considered to carry out the survey, and the complex sampling design is also represented in the way in which these subsets are generated.

The way of calling this function is depicted below and a brief description of each argument is given in Table 8.2:

```
replicate.weights(
  data,
  method = c("JKn",
             "dCV",
             "bootstrap",
             "subbootstrap",
             "BRR",
             "split",
```

```
            "extrapolation"),
  cluster = NULL,
  strata = NULL,
  weights = NULL,
  design = NULL,
  k = 10,
  R = 1,
  B = 200,
  train.prob = 0.7,
  method.split = c("dCV", "bootstrap", "subbootstrap"),
  rw.test = FALSE,
  dCV.sw.test = FALSE
)
```

Table 8.2: Summary and usage of the arguments incorporated to the function `replicate.weights()`.

Argument	Description
`data`	A data frame with information on (at least) cluster and strata indicators and sampling weights. It could be `NULL` if the sampling design is indicated in the `design` argument.
`method`	A character string indicating the replicate weights method to be applied to define training and test sets. Choose between one of these: `JKn`, `dCV`, `bootstrap`, `subbootstrap`, `BRR`, `split`, `extrapolation`.
`cluster`	A character string indicating the name of the column with cluster identifiers in the data frame indicated in `data`. It could be `NULL` if the sampling design is indicated in the `design` argument (see `design`).
`strata`	A character string indicating the name of the column with strata identifiers in the data frame indicated in `data`. It could be `NULL` if the sampling design is indicated in the `design` argument (see `design`).

`weights`	A character string indicating the name of the column with sampling weights in the data frame indicated in `data`. It could be `NULL` if the sampling design is indicated in the `design` argument (see `design`).
`design`	An object of class `survey.design` generated by `survey::svydesign()`. It could be `NULL` if information about `cluster`, `strata`, `weights` and `data` are given.
`k`	A numeric value indicating the number of folds to be defined. Default is `k=10`. Only applies for the `dCV` method.
`R`	A numeric value indicating the number of times the sample is partitioned. Default is `R=1`. Only applies for `dCV`, `split` or `extrapolation` methods.
`B`	A numeric value indicating the number of bootstrap resamples. Default is `B=200`. Only applies for `bootstrap` and `subbootstrap` methods.
`train.prob`	A numeric value between 0 and 1, indicating the proportion of clusters (for the method `split`) or strata (for the method `extrapolation`) to be set in the training sets. Default is `train.prob=0.7`. Only applies for `split` and `extrapolation` methods.
`method.split`	A character string indicating the way in which replicate weights should be defined in the `split` method. Choose one of the following: `dCV`, `bootstrap` or `subbootstrap`. Only applies for `split` method.
`rw.test`	A logical value. If `TRUE`, the function returns in the output object the replicate weights to the corresponding test sets. If `FALSE`, only the replicate weights of the training sets are returned. Default is `rw.test = FALSE`.
`dCV.sw.test`	A logical value. If `TRUE`, original sampling weights for the units in the test sets are returned instead of the replicate weights. Default is `dCV.sw.test = FALSE`. Only applies for `dCV` method. See more details below.

Some of these methods (specifically `JKn`, `bootstrap`, `subbootstrap` and `BRR`), were previously implemented in the `survey` R-package, to which we can access by means of the function `as.svrepdesign()` (the names of the methods are kept as

in `as.svrepdesign()`). Thus, the function `replicate.weights()` depends on this function to define replicate weights based on these options. In contrast, `dCV`, `split` and `extrapolation` have been expressly defined to be incorporated into this function.

As briefly explained in Table 8.2, some of the arguments are general arguments and need to be incorporated for any selected method (these arguments are `data`, `cluster`, `strata` and `weights`, or optionally `design`, in addition to `method`). Other arguments are optional and only apply to certain methods. Hence, more details on this function should be given to clarify this point, in particular, regarding the different replicate weight methods that can be applied and the arguments that each of the options needs.

Therefore, we proceed below with a brief description of the usage and output object of each replicate weights method. Selecting any of the above-mentioned methods, the object returned by the function `replicate.weights()` is a new data frame, which includes new columns into the original data set, each of them indicating replicate weights for different training (always) and test (optionally, controlled by the argument `rw.test`) subsets. In order to comment on all the information that may be obtained by means of this function, we set `rw.test = TRUE` in all the examples shown, which returns replicate weights of both training and test sets. The number of new columns and the way in which they are denoted depend on the values set for the arguments described in Table 8.2, in general, and on the replicate weights method selected, in particular. The new columns indicating training and test sets follow a similar structure for any of the selected methods. Specifically, the structure of the names of the training sets is the following: `rw_r_x_train_t` where `x=1,...,R` indicates the $\texttt{x}^{th}$ partition of the sample and `t=1,...,T` the $\texttt{t}^{th}$ training set. Similarly, the structure of the new columns indicating the test sets is the following: `rw_r_x_test_t` or `sw_r_x_test_t` (depending on the selected method, see more information below), where `x` indicates the partition and `t` the number of the test set. In addition, for some of the methods we also indicate the fold or set to which each unit in the data set has been included in each partition. This information is included as `fold_t` or `set_t`, depending on the method. See more detailed information below.

- The option `JKn` refers to the Jackknife Repeated Replication (JKn) method, the replicate weights of which were defined in eq. (5.20) for the training sets and in eq. (5.21) for the test sets. As the amount of training and test sets is determined by means of the number of clusters in the data set, no more

arguments are required to run the function `replicate.weights()` considering the method JKn. Below, we show an example of the correct usage of this function for the JKn method and the column names of the output data frame we obtain:

```
newdata <- replicate.weights(data = simdata_lasso_binomial[,51:54],
                             method = "JKn",
                             cluster = "cluster",
                             strata = "strata",
                             weights = "weights",
                             rw.test = TRUE)
names(newdata)
```

```
 [1] "strata"          "cluster"         "y"
 [4] "weights"         "rw_r_1_train_1"  "rw_r_1_train_2"
 [7] "rw_r_1_train_3"  "rw_r_1_train_4"  "rw_r_1_train_5"
[10] "rw_r_1_train_6"  "rw_r_1_train_7"  "rw_r_1_train_8"
[13] "rw_r_1_train_9"  "rw_r_1_train_10" "rw_r_1_train_11"
[16] "rw_r_1_train_12" "rw_r_1_train_13" "rw_r_1_train_14"
[19] "rw_r_1_train_15" "rw_r_1_train_16" "rw_r_1_train_17"
[22] "rw_r_1_train_18" "rw_r_1_train_19" "rw_r_1_train_20"
[25] "sw_r_1_test_1"   "sw_r_1_test_2"   "sw_r_1_test_3"
[28] "sw_r_1_test_4"   "sw_r_1_test_5"   "sw_r_1_test_6"
[31] "sw_r_1_test_7"   "sw_r_1_test_8"   "sw_r_1_test_9"
[34] "sw_r_1_test_10"  "sw_r_1_test_11"  "sw_r_1_test_12"
[37] "sw_r_1_test_13"  "sw_r_1_test_14"  "sw_r_1_test_15"
[40] "sw_r_1_test_16"  "sw_r_1_test_17"  "sw_r_1_test_18"
[43] "sw_r_1_test_19"  "sw_r_1_test_20"
```

In particular, note that a total of 40 columns has been added to the original data set, with the information of the replicate weights for the units in the 20 training sets (columns 5 to 24, defined as in eq. (5.20)) and 20 test sets (columns 25 to 44, following eq. (5.21)), given that there are $A = 20$ clusters in the data set.

- The option `dCV` refers to the design-based K-fold cross-validation (dCV) method. The number of folds to be defined should be indicated in the argument `k` (de-

fault is k=10). The number of times the original sample should be partitioned into k folds should also be indicated by means of the argument R (default is R=1). See eq. (5.26) to see the replicate weights for the training sets and eq. (5.27) for the test sets. An example of the usage and the output of the function for the method dCV are given below:

```
newdata <- replicate.weights(data = simdata_lasso_binomial[,51:54],
                             method = "dCV",
                             cluster = "cluster",
                             strata = "strata",
                             weights = "weights",
                             k = 5,
                             R = 2,
                             rw.test = TRUE)
names(newdata)
```

```
 [1] "strata"          "cluster"         "y"
 [4] "weights"         "folds_1"         "rw_r_1_train_1"
 [7] "rw_r_1_train_2"  "rw_r_1_train_3"  "rw_r_1_train_4"
[10] "rw_r_1_train_5"  "rw_r_1_test_1"   "rw_r_1_test_2"
[13] "rw_r_1_test_3"   "rw_r_1_test_4"   "rw_r_1_test_5"
[16] "folds_2"         "rw_r_2_train_1"  "rw_r_2_train_2"
[19] "rw_r_2_train_3"  "rw_r_2_train_4"  "rw_r_2_train_5"
[22] "rw_r_2_test_1"   "rw_r_2_test_2"   "rw_r_2_test_3"
[25] "rw_r_2_test_4"   "rw_r_2_test_5"
```

As R = 2 is indicated, the data has been split twice, with k = 5 folds each of them. In column 5, information on the first split of the data is given, indicating the fold to which each individual has been included. In columns 6 to 10, information on the replicate weights of the training sets (see eq. (5.26)) related to this split is given, while in columns 11 to 15, the replicate weights of the corresponding test sets (see eq. (5.27)) are returned. The same pattern is followed for the second split of the data set, starting in column 16 and following in columns 17 to 21 with replicate weights of the training sets and in columns 22 to 26 with replicate weights of the test sets.

The replicate weights obtained above for the units in the test sets are those obtained by the default option `dCV.sw.test = FALSE`, which as stated above, returns the replicate weights for the test set as described in eq. (5.27). However, we have another option to define the replicate weights for those units, by turning `dCV.sw.test = TRUE`, given that in this way, the original sampling weights are assigned to each unit in the test set (in a similar way as for the `JKn` method in eq. (5.21)) as defined in eq. (8.1):

$$w_{i,\text{dCV.pool}}^{*,\text{test}(t)} = \begin{cases} 0, & \text{if } i \in S_{\text{tr}(t)}^{\text{dCV}}, \\ w_i, & \text{if } i \in S_{\text{test}(t)}^{\text{dCV}}, \end{cases} \qquad \forall i \in S. \tag{8.1}$$

The usage of the function is indicated below:

```
newdata <- replicate.weights(data = simdata_lasso_binomial[,51:54],
                             method = "dCV",
                             cluster = "cluster",
                             strata = "strata",
                             weights = "weights",
                             k = 5,
                             R = 2,
                             rw.test = TRUE,
                             dCV.sw.test = TRUE)
names(newdata)
```

```
 [1] "strata"         "cluster"        "y"
 [4] "weights"        "folds_1"        "rw_r_1_train_1"
 [7] "rw_r_1_train_2" "rw_r_1_train_3" "rw_r_1_train_4"
[10] "rw_r_1_train_5" "folds_2"        "rw_r_2_train_1"
[13] "rw_r_2_train_2" "rw_r_2_train_3" "rw_r_2_train_4"
[16] "rw_r_2_train_5" "sw_r_1_test_1"  "sw_r_1_test_2"
[19] "sw_r_1_test_3"  "sw_r_1_test_4"  "sw_r_1_test_5"
[22] "sw_r_2_test_1"  "sw_r_2_test_2"  "sw_r_2_test_3"
[25] "sw_r_2_test_4"  "sw_r_2_test_5"
```

The structure of the output data set is similar to the option with `dCV.sw.test = FALSE`. However, note that the column names (now `sw_r_x_test_t` instead

of `rw_r_x_test_t` for `x`$\in \{$`1,...,R`$\}$ and `t`$\in \{$`1,...,T`$\}$) corresponding to the test sets (i.e., from 17 to 26 in this case obtained following eq. (8.1)) have been slightly changed in order to make the difference between the replicate weights obtained with eqs. (5.27) and (8.1) clearer (denoted now as `sw_` instead of `rw_`).

- The option `subbootstrap` refers to the Rescaling Bootstrap (Bootstrap) defined by Rao and Wu (1988) and explained in Section 5.2.3. Briefly, this method selects $a_h - 1$ clusters randomly with replacement among the a_h total clusters from each stratum h, $\forall h \in \{1, \ldots, H\}$ and the corresponding replicate weights have been defined in eq. (5.22). For the proper usage of this function, we only need to indicate the number of bootstrap resamples we aim to define by means of the argument `B` as shown below (the default option is `B=200`).

```
newdata <- replicate.weights(data = simdata_lasso_binomial[,51:54],
                             method = "subbootstrap",
                             cluster = "cluster",
                             strata = "strata",
                             weights = "weights",
                             B = 20)
names(newdata)
```

```
 [1] "strata"          "cluster"         "y"
 [4] "weights"         "rw_r_1_train_1"  "rw_r_1_train_2"
 [7] "rw_r_1_train_3"  "rw_r_1_train_4"  "rw_r_1_train_5"
[10] "rw_r_1_train_6"  "rw_r_1_train_7"  "rw_r_1_train_8"
[13] "rw_r_1_train_9"  "rw_r_1_train_10" "rw_r_1_train_11"
[16] "rw_r_1_train_12" "rw_r_1_train_13" "rw_r_1_train_14"
[19] "rw_r_1_train_15" "rw_r_1_train_16" "rw_r_1_train_17"
[22] "rw_r_1_train_18" "rw_r_1_train_19" "rw_r_1_train_20"
```

As shown above, the replicate weights of the training sets (in this case, the `B=20` bootstrap resamples) are added in columns 5 to 24. As explained in Section 5.2.3, the original sample is used as the test sets in this method, and the replicate weights corresponding to the test sets are equal to the original

sampling weights as indicated in eq. (5.23). Thus, with this method, the replicate weights corresponding to the test sets are not returned and the argument `rw.test` is always ignored.

- The option `bootstrap` refers to a slightly different Bootstrap variant proposed by Canty and Davison (1999). This method randomly selects a_h clusters with replacement, $\forall h \in \{1, \ldots, H\}$ (instead of $a_h - 1$, as the option `subbootstrap` does). Hence, the replicate weights of the training sets are calculated as defined in eq. (8.2) with this variant, $\forall t = 1, \ldots, T_{\text{Bootstrap}}$:

$$w_{i,\texttt{bootstrap}}^{*,\text{tr}(t)} = \sum_{h=1}^{H} \sum_{\dot{\alpha} \in \mathbb{A}_h} 1_{S_{h,\dot{\alpha}}}(i) \cdot w_i \cdot v_{h,\dot{\alpha}}^{(t)}, \quad \forall i \in S, \tag{8.2}$$

where now $T_{\text{Bootstrap}}$ is defined by means of the argument `B`, $1_{S_{h,\dot{\alpha}}}(i)$ indicates whether unit i is in cluster $\dot{\alpha}$ from stratum h in the sample S (1) or not (0), and $v_{h,\dot{\alpha}}^{(t)}$ indicates the number of times cluster $\dot{\alpha}$ from stratum h has been selected to be part of the t^{th} resample, $\forall \dot{\alpha} \in \mathbb{A}_h$ and $\forall h \in \{1, \ldots, H\}$.

The usage and structure of the output data frame are equal to those of the `subbootstrap` indicated above:

```
newdata <- replicate.weights(data = simdata_lasso_binomial[,51:54],
                             method = "bootstrap",
                             cluster = "cluster",
                             strata = "strata",
                             weights = "weights",
                             B = 20)
names(newdata)
```

```
 [1] "strata"          "cluster"         "y"
 [4] "weights"         "rw_r_1_train_1"  "rw_r_1_train_2"
 [7] "rw_r_1_train_3"  "rw_r_1_train_4"  "rw_r_1_train_5"
[10] "rw_r_1_train_6"  "rw_r_1_train_7"  "rw_r_1_train_8"
[13] "rw_r_1_train_9"  "rw_r_1_train_10" "rw_r_1_train_11"
[16] "rw_r_1_train_12" "rw_r_1_train_13" "rw_r_1_train_14"
[19] "rw_r_1_train_15" "rw_r_1_train_16" "rw_r_1_train_17"
[22] "rw_r_1_train_18" "rw_r_1_train_19" "rw_r_1_train_20"
```

As in the option `method = "subbootstrap"`, the replicate weights of the training sets are added in columns 5 to 24 and the argument `rw.test` is ignored.

- The option `BRR` refers to the Balanced Repeated Replication (BRR) method, for which replicate weights have been defined in eq. (5.24) for the training sets and in eq. (5.25) for the test sets. As the number of the total training and test sets is defined based on the Hadamard matrix as explained in Section 5.2.3, none of the rest of the arguments are required to apply this method. The usage of the function for this option is shown below:

```
newdata <- replicate.weights(data = simdata_lasso_binomial[,51:54],
                             method = "BRR",
                             cluster = "cluster",
                             strata = "strata",
                             weights = "weights",
                             rw.test = TRUE)
names(newdata)
```

```
 [1] "strata"          "cluster"         "y"
 [4] "weights"         "rw_r_1_train_1"  "rw_r_1_train_2"
 [7] "rw_r_1_train_3"  "rw_r_1_train_4"  "rw_r_1_train_5"
[10] "rw_r_1_train_6"  "rw_r_1_train_7"  "rw_r_1_train_8"
[13] "rw_r_1_train_9"  "rw_r_1_train_10" "rw_r_1_train_11"
[16] "rw_r_1_train_12" "rw_r_1_test_1"   "rw_r_1_test_2"
[19] "rw_r_1_test_3"   "rw_r_1_test_4"   "rw_r_1_test_5"
[22] "rw_r_1_test_6"   "rw_r_1_test_7"   "rw_r_1_test_8"
[25] "rw_r_1_test_9"   "rw_r_1_test_10"  "rw_r_1_test_11"
[28] "rw_r_1_test_12"
```

The replicate weights of the training sets are included in columns 5 to 16, and the replicate weights of the test sets in columns 17 to 28.

- The option `split` refers to the Split-sample Repeated Replication (split) method. As explained in Section 5.2.3, this method refers to the way in which training and test sets should be defined, but the way in which replicate weights should be calculated needs to be indicated by means of the argument `method.split`. This function provides three different options to do so:

- `method.split = "dCV"` defines replicate weights based on the dCV method as defined in eqs. (5.29) and (5.30).
- `method.split = "subbootstrap"` defines replicate weights based on the Rescaling Bootstrap method as defined in eqs. (5.31) and (5.32).
- `method.split = "bootstrap"` defines replicate weights by means of the Bootstrap variant proposed by Canty and Davison (1999), as defined in eqs. (8.3) and (8.4):

$$w^{*,\text{tr}(t)}_{i,\texttt{split-bootstrap}} = \sum_{h=1}^{H} \sum_{\dot{\alpha} \in \mathbb{A}_h} 1_{S_{h,\dot{\alpha}}}(i) \cdot w_i \cdot v^{(t)}_{h,\dot{\alpha}}, \quad \forall i \in S, \tag{8.3}$$

$$w^{*,\text{test}(t)}_{i,\texttt{split-bootstrap}} = \sum_{h=1}^{H} \sum_{\dot{\alpha} \in \mathbb{A}_h} 1_{S_{h,\dot{\alpha}}}(i) \cdot w_i \cdot \tilde{v}^{(t)}_{h,\dot{\alpha}}, \quad \forall i \in S, \tag{8.4}$$

  where $v^{(t)}_{h,\dot{\alpha}}$ indicates the number of times that $S_{h,\dot{\alpha}}$ is selected to be part of the t^{th} training set $S^{\texttt{split-bootstrap}}_{\text{train}(t)}$, $\forall \dot{\alpha} \in \mathbb{A}_h$ and $\forall h \in \{1, \ldots, H\}$. Note that if $S_{h,\dot{\alpha}}$ is set to the test set, then, $v^{(t)}_{h,\dot{\alpha}} = 0$. Similarly, $\tilde{v}^{(t)}_{h,\dot{\alpha}}$ indicates the number of times that $S_{h,\dot{\alpha}}$ is selected to be part of the t^{th} test set $S^{\texttt{split-bootstrap}}_{\text{test}(t)}$. If $S_{h,\dot{\alpha}}$ is set to the training set, then, $\tilde{v}^{(t)}_{h,\dot{\alpha}} = 0$.

In addition, we also need to indicate the proportion of primary sampling units (i.e., clusters in two-stage stratified cluster samplings and individuals in one-stage stratified samplings) to be set into the training set, which is controlled by the argument `train.prob` (default is `train.prob = 0.7`), as well as, the number of times we would like to split the original sample into training and test sets (with the argument `R`, being the default option `R=1`). An example of the usage of the function for this option is given below:

```
newdata <- replicate.weights(data = simdata_lasso_binomial[,51:54],
                             method = "split",
                             cluster = "cluster",
                             strata = "strata",
                             weights = "weights",
                             R=5,
                             train.prob = 0.5,
                             method.split = "dCV",
```

```
                                rw.test = TRUE)
names(newdata)
```

```
 [1] "strata"       "cluster"      "y"            "weights"
 [5] "set_1"        "rw_r_1_test"  "rw_r_1_train" "set_2"
 [9] "rw_r_2_train" "rw_r_2_test"  "set_3"        "rw_r_3_test"
[13] "rw_r_3_train" "set_4"        "rw_r_4_test"  "rw_r_4_train"
[17] "set_5"        "rw_r_5_train" "rw_r_5_test"
```

The data set has been split a total of `R = 5` times. In column 5, we can find information on whether each unit in the data set has been included in the training or in the test set in the first split of the data set, being the corresponding replicate weights of the training sets indicated in column 6 and the ones corresponding to the test sets in column 7. The same structure is followed in columns 9 to 19 for each of the other four splits of the data set.

- The option `extrapolation` refers to the Extrapolation (extrap) method, for which replicate weights have been defined in eq. (5.33). We need to define the proportion of strata we aim to set to the training set by means of the `train.prob` argument (default is `train.prob = 0.7`) and the number of times the original data set should be partitioned with the argument `R` (being the default `R=1`). An example of the usage considering this method is given below:

```
newdata <- replicate.weights(data = simdata_lasso_binomial[,51:54],
                             method = "extrapolation",
                             cluster = "cluster",
                             strata = "strata",
                             weights = "weights",
                             R=5,
                             train.prob = 0.5,
                             rw.test = TRUE)

names(newdata)
```

```
 [1] "strata"       "cluster"      "y"            "weights"
 [5] "set_1"        "rw_r_1_train" "rw_r_1_test"  "set_2"
 [9] "rw_r_2_train" "rw_r_2_test"  "set_3"        "rw_r_3_train"
[13] "rw_r_3_test"  "set_4"        "rw_r_4_train" "rw_r_4_test"
[17] "set_5"        "rw_r_5_train" "rw_r_5_test"
```

The output of this option is quite similar to the `method = "split"`, and information of the data partition for each of the `R = 5` splits (indicating whether each unit is set into the training or test set), and the corresponding replicate weights of the training and test sets are indicated every three columns, starting in column 5 and until column 19.

All the commands indicated above can also be run by including the information of the design defined by the function `svydesign()` from the `survey` R-package (Lumley 2020) instead of inserting the data and the column names indicating the clusters, strata and sampling weights. For example, for the dCV method, we could also run the following code:

```
mydesign <- survey::svydesign(ids=~cluster,
                              strata = ~strata,
                              weights = ~weights,
                              nest = TRUE,
                              data = simdata_lasso_binomial)
newdata <- replicate.weights(method = "dCV",
                             design = mydesign,
                             k = 5, R = 2,
                             rw.test = TRUE)
```

8.1.2 `wlasso()` function

The function `wlasso()` allows fitting either linear or logistic LASSO regression models to complex survey data, considering sampling weights in the estimation process. This function also indicates the value of the tuning parameter that minimizes the error. For this purpose, it uses the `replicate.weights()` function, which has been detailed in Section 8.1.1, to define training and test sets. All the methods implemented in this function have been proposed and described in detail mathematically in Chapter 5.

The usage of this function is shown below. The summary and explanation of the arguments can be found in Table 8.3.

```
wlasso(
  data = NULL,
  col.y = NULL,
  col.x = NULL,
  cluster = NULL,
  strata = NULL,
  weights = NULL,
  design = NULL,
  family = c("gaussian", "binomial"),
  lambda.grid = NULL,
  method = c("dCV",
             "JKn",
             "bootstrap",
             "subbootstrap",
             "BRR",
             "split",
             "extrapolation"),
  k = 10,
  R = 1,
  B = 200,
  dCV.sw.test = FALSE,
  train.prob = 0.7,
  method.split = c("dCV", "bootstrap", "subbootstrap"),
  print.rw = FALSE
)
```

Table 8.3: Summary and usage of the arguments incorporated to the function `wlasso()`.

Argument	**Description**
`data`	A data frame with information about the response variable and covariates, as well as sampling weights and strata and cluster indicators. It could be `NULL` if the sampling design is indicated in the `design` argument.
`col.y`	A numeric value indicating the number of the column in which information on the response variable can be found or a character string indicating the name of that column.
`col.x`	A numeric vector indicating the numbers of the columns in which information on the covariates can be found or a vector of character strings indicating the names of these columns.
`cluster`	A character string indicating the name of the column with cluster identifiers. It could be `NULL` if the sampling design is indicated in the `design` argument.
`strata`	A character string indicating the name of the column with strata identifiers. It could be `NULL` if the sampling design is indicated in the `design` argument.
`weights`	A character string indicating the name of the column with sampling weights. It could be `NULL` if the sampling design is indicated in the `design` argument.
`design`	An object of class `survey.design` generated by `survey::svydesign()`. It could be `NULL` if information about `cluster`, `strata`, `weights` and `data` are given.
`family`	A character string indicating the family to fit LASSO models. Choose between `gaussian` (to fit linear models) or `binomial` (for logistic models).
`lambda.grid`	A numeric vector indicating a grid for penalization parameters. The default option is `lambda.grid = NULL`, which considers the default grid selected by the function `glmnet::glmnet()` (Friedman et al. 2010).
`method`	A character string indicating the method to be applied to define replicate weights. Choose between one of these: `JKn`, `dCV`, `bootstrap`, `subbootstrap`, `BRR`, `split`, `extrapolation`.

`k`	A numeric value indicating the number of folds to be defined. Default is `k=10`. Only applies for the `dCV` method.
`R`	A numeric value indicating the number of times the sample is partitioned. Default is `R=1`. Only applies for `dCV`, `split` or `extrapolation` methods.
`B`	A numeric value indicating the number of bootstrap resamples. Default is `B=200`. Only applies for `bootstrap` and `subbootstrap` methods.
`dCV.sw.test`	A logical value indicating the method for estimating the error for `dCV` method. `FALSE`, (the default option) estimates the error for each test set and defines the cross-validated error as the average of all those errors as described in eqs. (5.47) and (5.48). Option `TRUE` estimates the cross-validated error based on the pooling strategy (Airola et al. 2011). See more information below.
`train.prob`	A numeric value between 0 and 1, indicating the proportion of clusters (for the method `split`) or strata (for the method `extrapolation`) to be set in the training sets. Default is `train.prob = 0.7`. Only applies for `split` and `extrapolation` methods.
`method.split`	A character string indicating the way in which replicate weights should be defined in the `split` method. Choose one of the following: `dCV`, `bootstrap` or `subbootstrap`. Only applies for `split` method.
`print.rw`	A logical value. If `TRUE`, the data set with the replicate weights is saved in the output object. Default `print.rw=FALSE`.

Let us explain in more detail the performance of this function. This function fits (either linear (`family="gaussian"`) or logistic (`family="binomial"`)) LASSO regression models for a grid of tuning parameters, considering sampling weights. It analyzes the error produced by each of the tuning parameters by generating training and test sets based on the function `replicate.weights()` and indicates which of the tuning parameters we should select in order to minimize the error of our LASSO model. Hence, some of the arguments employed in this function are related to the replicate weights method selected to fit LASSO regression models and have already been explained in detail in Section 8.1.1. Specifically, the arguments

`method`, `k`, `R`, `B`, `dCV.sw.test`, `train.prob` and `method.split` have been previously defined in Section 8.1.1. However, another issue that should also be clarified is the implications of selecting the option `TRUE` or `FALSE` in the argument `dCV.sw.test` in terms of error estimation. If the default option `dCV.sw.test = FALSE` is selected, the cross-validated error for a given value of the tuning parameter $\lambda_l, \forall l \in \{1, \ldots, L\}$ is calculated as indicated in eq. (5.48), that is,

$$\widehat{Err}_{\text{test}}^{\text{dCV}}(\lambda_l) = \frac{1}{T_{\text{dCV}}} \sum_{t=1}^{T_{\text{dCV}}} \widehat{Err}_{\text{test}}^{\text{dCV},t}(\lambda_l), \tag{8.5}$$

where,

$$\widehat{Err}_{\text{test}}^{\text{dCV},t}(\lambda_l) = \frac{1}{\sum_{i \in S_{\text{test}(t)}^{\text{dCV}}} w_{i,\text{dCV}}^{*,\text{test}(t)}} \sum_{i \in S_{\text{test}(t)}^{\text{dCV}}} w_{i,\text{dCV}}^{*,\text{test}(t)} \mathcal{L}(y_i, \hat{f}_{\text{tr}(t)}^{l,\text{dCV}}(\boldsymbol{x}_i)), \tag{8.6}$$

where the replicate weights $w_{i,\text{dCV}}^{*,\text{test}(t)}$ have been defined in eq. (5.27). In contrast, if we turn to `dCV.sw.test = TRUE`, then, the error is estimated as defined in eq. (5.49) for the JKn method, that is,

$$\widehat{Err}_{\text{test}}^{\text{dCV.pool}}(\lambda_l) = \frac{\sum_{t=1}^{T_{\text{dCV}}} \sum_{i \in S_{\text{test}(t)}^{\text{dCV}}} w_{i,\text{dCV.pool}}^{*,\text{test}(t)} \cdot \mathcal{L}(y_i, \hat{f}_{\text{tr}(t)}^{l,\text{dCV}}(\boldsymbol{x}_i))}{\sum_{t=1}^{T_{\text{dCV}}} \sum_{i \in S_{\text{test}(t)}^{\text{dCV}}} w_{i,\text{dCV.pool}}^{*,\text{test}(t)}}, \tag{8.7}$$

where $w_{i,\text{dCV.pool}}^{*,\text{test}(t)}$ have been defined in eq. (8.1). This strategy is usually known in the literature as *pooling* cross-validation (see, e.g., Airola et al. (2011), Swets and Pickett (1982)).

In the following lines, we show some examples of the usage of the function `wlasso()`. We can run the `wlasso()` function in two different ways. On the one hand, we can indicate the name of the data set, and the names of the columns corresponding to the cluster indicator, strata indicator and sampling weights as character strings:

```
set.seed(1)
mdCV <- wlasso(data = simdata_lasso_binomial,
               col.y = "y", col.x = 1:50,
               family = "binomial",
               cluster = "cluster",
               strata = "strata",
```

```
                weights = "weights",
                method = "dCV", k=10, R=1)
```

On the other hand, we can also incorporate the information of the survey design by means of the `svydesign` of the `survey` package equivalently, i.e.,

```
mydesign <- survey::svydesign(ids= ~cluster,
                              strata = ~strata,
                              weights = ~weights,
                              nest = TRUE,
                              data = simdata_lasso_binomial)
set.seed(1)
mdCV <- wlasso(col.y = "y", col.x = 1:50, design = mydesign,
                family = "binomial",
                method = "dCV", k=10, R=1)
```

The output object of the function `wlasso()` is an object of class `wlasso`. This object is a list containing 4 or 5 elements, depending on the value set to the argument `print.rw`. Below we describe the contents of these elements:

- `lambda`: A list containing information of two elements:
  - `grid`: A numeric vector indicating all the values considered for the tuning parameter.
  - `min`: A numeric value indicating the value of the tuning parameter that minimizes the average error (i.e., selected optimal tuning parameter).
- `error`: A list containing information of two elements:
  - `average`: A numeric vector indicating the average error corresponding to each tuning parameter.
  - `all`: A numeric matrix indicating the error of each test set for each tuning parameter.
- `model`: A list containing information of two elements in relation to the fitted models. Note that all these models are fitted considering the whole data set (and not uniquely the training sets).

  - `grid`: A list with the information about the models fitted for each of the tuning parameters considered (i.e., all the values in the `lambda$grid` object):
    * `a0`: a numeric vector of model intercepts across the whole grid of tuning parameters (hence, of the same length as `lambda$grid`).
    * `beta`: a matrix of regression coefficients corresponding to all the considered covariates across the whole grid of tuning parameters (the number of rows is equal to the number of covariates considered and the number of columns to the length of `lambda$grid`).
    * `df`: a numeric vector of the degrees of freedom (i.e., the number of coefficients different from zero) across the whole grid of tuning parameters (hence, of the same length as `lambda$grid`).
  - `min`: A list with the information about the model fitted considering uniquely the tuning parameter that minimizes the error in the training models (i.e., the optimal tuning parameter selected between the elements in `lambda$grid`):
    * `a0`: a numeric value indicating the intercept value of the selected model.
    * `beta`: a matrix of regression coefficients corresponding to all the considered covariates for the selected tuning parameters (the number of rows is equal to the number of covariates considered and the number of columns is one).
    * `df`: a numeric value indicating the degrees of freedom (i.e., the number of coefficients different from zero) of the selected model.

- `data.rw`: A data frame containing the original data set and the replicate weights added to define training and test sets. Only included in the output object if `print.rw=TRUE`.

- `call`: an object containing the information about the way in which the function has been run.

As an example, let us show the output obtained in the above-example and saved into the object `mdCV`. The tuning parameters' grid is available in the `lambda$grid` object.

```
mdCV$lambda$grid
```

```
 [1] 0.0990628869 0.0902624131 0.0822437491 0.0749374411 0.0682802053
 [6] 0.0622143801 0.0566874260 0.0516514713 0.0470628970 0.0428819589
[11] 0.0390724439 0.0356013557 0.0324386293 0.0295568709 0.0269311200
[16] 0.0245386335 0.0223586889 0.0203724046 0.0185625763 0.0169135282
[21] 0.0154109770 0.0140419083 0.0127944638 0.0116578389 0.0106221887
[26] 0.0096785427 0.0088187276 0.0080352962 0.0073214626 0.0066710441
[31] 0.0060784069 0.0055384180 0.0050464002 0.0045980919 0.0041896101
[36] 0.0038174167 0.0034782879 0.0031692864 0.0028877357 0.0026311972
[41] 0.0023974489 0.0021844662 0.0019904042 0.0018135822 0.0016524685
[46] 0.0015056678 0.0013719084 0.0012500319 0.0011389825 0.0010377984
[51] 0.0009456033 0.0008615985 0.0007850564 0.0007153142 0.0006517676
[56] 0.0005938664 0.0005411090 0.0004930383 0.0004492382 0.0004093291
[61] 0.0003729654 0.0003398322 0.0003096424 0.0002821347 0.0002570706
[66] 0.0002342331
```

The tuning parameter value that minimizes the error in the test sets can easily be observed in the object `lambda$min`:

```
mdCV$lambda$min
```

```
[1] 0.01856258
```

The average error for every tuning parameter considered can be checked in the object `error$average`:

```
mdCV$error$average
```

```
 [1] 0.5819602 0.5796554 0.5768222 0.5736486 0.5707556 0.5661586
 [7] 0.5561894 0.5450417 0.5351562 0.5257853 0.5172037 0.5104355
[13] 0.5040830 0.4990725 0.4939163 0.4894817 0.4859550 0.4833799
[19] 0.4822768 0.4823669 0.4832621 0.4845715 0.4863010 0.4883728
[25] 0.4900422 0.4921848 0.4950723 0.4980396 0.5008377 0.5037935
[31] 0.5069181 0.5100185 0.5131271 0.5160820 0.5189973 0.5219854
```

```
[37] 0.5248796 0.5275667 0.5300389 0.5324649 0.5347991 0.5370334
[43] 0.5391630 0.5411648 0.5430402 0.5448048 0.5464661 0.5480274
[49] 0.5494846 0.5508856 0.5522167 0.5534747 0.5546379 0.5557256
[55] 0.5567348 0.5576666 0.5585061 0.5592891 0.5600148 0.5606996
[61] 0.5613302 0.5618975 0.5624285 0.5628735 0.5632485 0.5636692
```

In the same line, the error of each training model in the corresponding test set is also available in the object `error$all`. We reduce the printed information about all the errors of each test set for the reduction of used space by printing uniquely the errors corresponding to the first, optimal and last values of the tuning parameter in the selected grid and the optimum value that minimizes the error:

```
mdCV$error$all
```

```
                 [,1]       ...      [,19]       ...      [,66]
dCV_r_1_k_1  0.5390746       ...  0.5090328       ...  0.7527343
dCV_r_1_k_2  0.6366243       ...  0.4848907       ...  0.5121032
dCV_r_1_k_3  0.6182985       ...  0.5435209       ...  0.6622000
dCV_r_1_k_4  0.5624325       ...  0.4206534       ...  0.3967353
dCV_r_1_k_5  0.4566277       ...  0.3835127       ...  0.3734332
dCV_r_1_k_6  0.4793457       ...  0.4143847       ...  0.5151107
dCV_r_1_k_7  0.6332941       ...  0.5228030       ...  0.5300957
dCV_r_1_k_8  0.6645730       ...  0.6706335       ...  0.8423438
dCV_r_1_k_9  0.5630955       ...  0.4181648       ...  0.4299665
dCV_r_1_k_10 0.6662359       ...  0.4551716       ...  0.6219693
```

Finally, we can find information about all the models fitted to the data set with all the considered tuning parameter values (in the object `model$grid`), as well as the optimum model that minimizes the error in the test sets (object `model$min`). For example, if we want to observe the number of covariates that end up in the model for each tuning parameter, we can find it in the following object:

```
mdCV$model$grid$df
```

```
 [1]  0  3  3  3  5  8 10 10 12 13 13 14 16 20 21 23 23 23 23 25 27 29
[23] 32 34 35 38 39 40 40 40 41 42 42 43 43 43 45 47 48 48 48 48 49 50
[45] 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50
```

In case we want to print the total number of covariates that end up in the final model considering the optimal tuning parameter, then we can run the following code:

```
mdCV$model$min$df
```

```
[1] 23
```

In the same way, if we want to know the model coefficients estimated considering the optimal tuning parameter, we continue as follows:

```
mdCV$model$min$beta
```

```
50 x 1 sparse Matrix of class "dgCMatrix"
              s0
x.1   .
x.2   .
x.3   0.06834372
x.4   0.57938665
x.5   0.20271896
x.6   .
x.7  -0.08561328
x.8   .
x.9  -0.15569722
x.10  .
x.11  .
x.12  0.23198427
x.13  0.03891243
x.14  .
x.15  .
x.16  .
x.17  0.12863789
x.18  .
x.19  .
x.20  .
x.21  .
x.22  .
```

```
x.23 -0.18228043
x.24 -0.23874770
x.25 -0.10372963
x.26 -0.17349581
x.27 -0.19391779
x.28 -0.26335366
x.29 -0.22867425
x.30  .
x.31  .
x.32  .
x.33  .
x.34  .
x.35  .
x.36  .
x.37  .
x.38  .
x.39  0.05542878
x.40 -0.05654776
x.41 -0.31603315
x.42  .
x.43  .
x.44 -0.07576487
x.45 -0.08442320
x.46 -0.28023181
x.47 -0.14774265
x.48 -0.16958293
x.49  .
x.50  .
```

Dots indicate that the corresponding variable does not end up in the final model (i.e., its model coefficient takes the value 0 in the final model).

Note that this function only provides estimates for model coefficients. In order to obtain the variance estimation of those coefficients, we should fit the corresponding regression model considering uniquely the subset of those variables that end up in the final model (i.e., that have coefficient values different to 0) by means of the `survey::svyglm()` function.

8.1.3 `wlasso.plot()` function

By means of the function `wlasso.plot()`, we can summarize graphically the information of an object of class `wlasso`, obtained by the function `wlasso()` as explained in Section 8.1.2.

```
wlasso.plot(x)
```

Table 8.4: Summary and usage of the unique argument incorporated to the function `wlasso.plot()`.

Argument	Description
`x`	An object of class `wlasso`.

The output object of this function is a graph. It depicts the average error of the training models for each tuning parameter. The tuning parameters are depicted in the logarithmic scale. The minimum value of the error is indicated and the corresponding degrees of freedom of the corresponding model (i.e., the number of covariates that end up in the model with the tuning parameter that minimizes the average error of the training sets) is also specified.

For example, if we plot the object `mdCV` previously obtained by means of the `wlasso()` function in Section 8.1.2, the resulting graph can be observed in Figure 8.1.

```
wlasso.plot(mdCV)
```

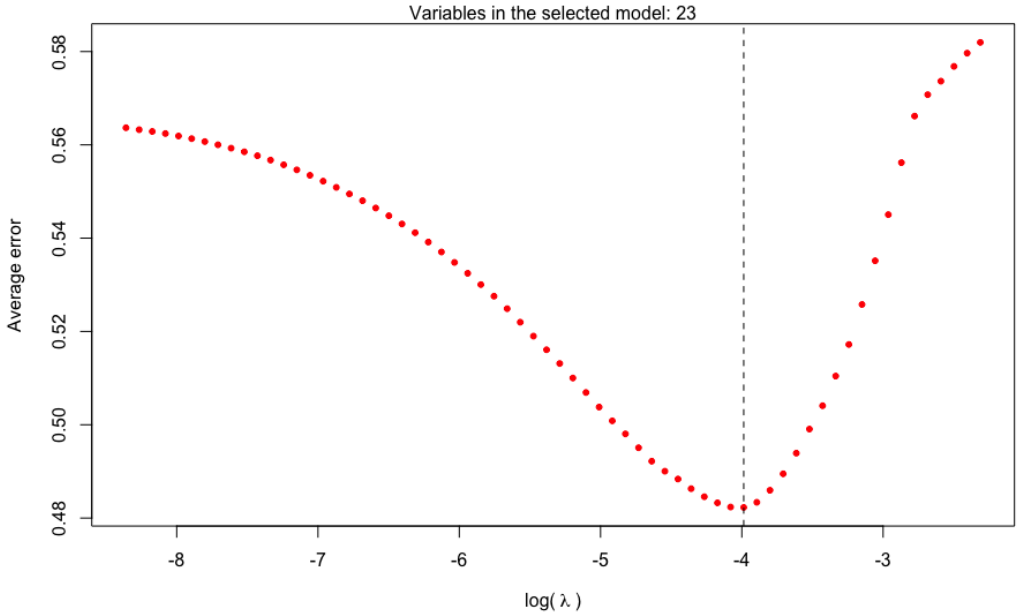


Figure 8.1: An example of the output graph obtained by means of the function `wlasso.plot()`.

8.2 `wROC` R-package

The main goal of the R-package `wROC` is to estimate the ROC curve, AUC and optimal cut-off points for individual classification for logistic regression models fitted to complex survey data. The methods incorporated into this package have been proposed and described in detail in Chapters 6 and 7. This package is available on GitHub: https://github.com/aiparragirre/wROC.

In particular, there are two ways to install this package in R. On the one hand, the whole GitHub repository can be downloaded and the package can manually be installed in R. On the other hand, we can also run the following code to install the package directly without the need to download it manually:

```
library("devtools")
install_github("aiparragirre/wROC/wROC")
library(wROC)
```

Six main functions are available in this R-package. Table 8.5 summarizes the principal purposes of these functions. In addition, the particular sections, in which the methodological details concerning each function can be found, are indicated.

Table 8.5: Summary of the functions available in the R-package `wROC`, a brief description of them, and the corresponding references to the sections in which the implemented methodology has been described.

Function	**Description**	**Methods**
`wsp()`	Estimation of the specificity parameter considering sampling weights.	Sections 6.2.2 and 7.2.2.
`wse()`	Estimation of the specificity parameter considering sampling weights.	Sections 6.2.2 and 7.2.2.
`wocp()`	Estimation of optimal cut-off points considering sampling weights.	Section 7.2.2.
`wauc()`	Estimation of the AUC considering sampling weights.	Section 6.2.2.
`wroc()`	Estimation of the ROC curve considering sampling weights.	Section 6.2.2.
`wroc.plot()`	Plot the ROC curve.	

In addition, a data set has been incorporated into the package for illustration purposes. We can access this data set, once the package has been installed, as follows:

```
data(example_data_wroc)
names(example_data_wroc)
```

```
"y"       "weights" "phat"
```

This data set contains information about 740 simulated observations. We have information on the response variable (`"y"`), sampling weights (`"weights"`), and predicted probabilities (`"phat"`) estimated based on a particular logistic regression model for each observation.

We have not incorporated more information about the sampling design other than the sampling weights, given that this information is enough for the current purposes of this package. However, note that all the functions of this package are

also prepared for working with objects of class `survey.design` obtained by the function `survey::svydesign()` in the same way as explained in Section 8.1.

It should also be noted that traditional unweighted estimates of the sensitivity and specificity parameters, ROC curve, AUC, and optimal cut-off points can be obtained with the functions of this package by setting all the sampling weights to 1 (or any other positive value different to 1, but the same for all the units).

In the following sections, each function is explained in detail.

8.2.1 `wsp()` function

The function `wsp()` allows estimating the specificity parameter given a cut-off point and considering sampling weights. The methods implemented in this function have first been presented in Section 6.2.2 and, afterward, again in Section 7.2.2.

The usage of this function is specified in the following lines, and its arguments are defined in Table 8.6.

```
wsp(
  response.var,
  phat.var,
  weights.var = NULL,
  tag.nonevent = NULL,
  cutoff.value,
  data = NULL,
  design = NULL
)
```

Table 8.6: Summary and usage of the arguments incorporated to the function `wsp()`.

Argument	Description
`response.var`	A character string with the name of the column indicating the response variable in the data set or a vector (either numeric or character string) with information of the response variable for all the units.
`phat.var`	A character string with the name of the column indicating the estimated probabilities in the data set or a numeric vector containing estimated probabilities for all the units.

`weights.var`	A character string indicating the name of the column with sampling weights or a numeric vector containing information of the sampling weights. It could be `NULL` if the sampling design is indicated in the `design` argument.
`tag.nonevent`	A character string indicating the label used for non-event in `response.var`. The default option is `tag.nonevent = NULL`, which selects the class with the greatest number of units as non-event.
`cutoff.value`	A numeric value indicating the cut-off point to be used. No default value is set for this argument, and a numeric value must be indicated necessarily.
`data`	A data frame which, at least, must contain information on the columns `response.var`, `phat.var` and `weights.var`. If `data=NULL`, then specific numerical vectors must be included in `response.var`, `phat.var` and `weights.var` or the sampling design should be indicated in the argument `design`.
`design`	An object of class `survey.design` generated by `survey::svydesign()` indicating the complex sampling design of the data. If `design=NULL`, information on the data set (argument `data`) and/or sampling weights (argument `weights.var`) must be included.

In the following lines, we show a couple of examples of the correct usage of the function `wsp()`. For example, we can indicate the data set and the column names indicating the response variable, predicted probabilities and sampling weights for each unit, as follows:

```
sp.obj <- wsp(response.var = "y",
              phat.var = "phat",
              weights.var = "weights",
              tag.nonevent = 0,
              cutoff.value = 0.5,
              data = example_data_wroc)
```

Equivalently, we can also include the numeric vectors indicating the values for the response variable, predicted probabilities and sampling weights for all the units in the sample:

```
sp.obj <- wsp(response.var = example_data_wroc$y,
              phat.var = example_data_wroc$phat,
              weights.var = example_data_wroc$weights,
              tag.nonevent = 0,
              cutoff.value = 0.5)
```

The output of this function is a list of 4 elements containing the following information:

- `Spw`: a numeric value indicating the weighted estimate of the specificity parameter.
- `tags`: a list containing one element with the following information:
  - `tag.nonevent`: a character string indicating the label used for non-events.
- `basics`: a list containing information of the following 6 elements:
  - `n`: a numeric value indicating the number of units in the data set.
  - `n.nonevent`: a numeric value indicating the number of units in the data set without the event of interest.
  - `n.nonevent.class`: a numeric value indicating the number of units in the data set without the event of interest that are correctly classified as non-events based on the selected cut-off point.
  - `hatN`: a numeric value indicating the number of units in the population that are represented by means of the units in the data set, i.e., the sum of the sampling weights of all the units in the data set.
  - `hatN.nonevent`: a numeric value indicating the number of non-event units in the population represented by means of the non-event units in the data set, i.e., the sum of the sampling weights of the non-event units in the data set.
  - `hatN.nonevent.class`: number of non-event units represented in the population by the non-event units in the data set that have been correctly classified as non-events based on the selected cut-off point, i.e., the sum of the sampling weights of the correctly classified non-event units in the data set.

- `call`: an object saving the information about the way in which the function has been run.

For example, the object `sp.obj` obtained above contains the following information:

```
sp.obj
```

```
$Spw
[1] 0.9060005

$tags
$tags$tag.nonevent
[1] "0"


$basics
$basics$n
[1] 740

$basics$n.nonevent
[1] 540

$basics$n.nonevent.class
[1] 464

$basics$hatN
[1] 10000

$basics$hatN.nonevent
[1] 7305

$basics$hatN.nonevent.class
[1] 6618.333
```

```
$call
wsp(response.var = "y",
    phat.var = "phat",
    weights.var = "weights",
    tag.nonevent = 0,
    cutoff.value = 0.5,
    data = example_data_wroc)
```

8.2.2 `wse()` function

The function `wse()` can be used to estimate the sensitivity parameter for a given cut-off point based on its estimated probability of event and considering sampling weights. The methods implemented in this function have first been presented in Section 6.2.2 and, afterward, again in Section 7.2.2.

In the following lines, the usage of this function is described and the summary of the arguments is given in Table 8.7.

```
wse(
  response.var,
  phat.var,
  weights.var = NULL,
  tag.event = NULL,
  cutoff.value,
  data = NULL,
  design = NULL
)
```

Table 8.7: Summary and usage of the arguments incorporated to the function `wse()`.

Argument	Description
`response.var`	A character string with the name of the column indicating the response variable in the data set or a vector (either numeric or character string) with information of the response variable for all the units.
`phat.var`	A character string with the name of the column indicating the estimated probabilities in the data set or a numeric vector containing estimated probabilities for all the units.

`weights.var`	A character string indicating the name of the column with sampling weights or a numeric vector containing information of the sampling weights. It could be `NULL` if the sampling design is indicated in the `design` argument.
`tag.event`	A character string indicating the label used to indicate the event of interest in `response.var`. The default option is `tag.event = NULL`, which selects the class with the lowest number of units as event.
`cutoff.value`	A numeric value indicating the cut-off point to be used. No default value is set for this argument, and a numeric value must be indicated necessarily.
`data`	Data frame which, at least, must contain information on the columns `response.var`, `phat.var` and `weights.var`. If `data=NULL`, then specific numerical vectors must be included in `response.var`, `phat.var` and `weights.var` or the sampling design should be indicated in the argument `design`.
`design`	An object of class `survey.design` generated by `survey::svydesign()` indicating the complex sampling design of the data. If `design=NULL`, information on the data set (argument `data`) and/or sampling weights (argument `weights.var`) must be included.

In the following lines, we show a couple of examples on how the function `wse()` can be used:

```
se.obj <- wse(response.var = "y",
              phat.var = "phat",
              weights.var = "weights",
              tag.event = 1,
              cutoff.value = 0.5,
              data = example_data_wroc)
```

Or equivalently, the function can also be used in the following way, obtaining the same result:

```
se.obj <- wse(response.var = example_data_wroc$y,
              phat.var = example_data_wroc$phat,
              weights.var = example_data_wroc$weights,
              tag.event = 1,
              cutoff.value = 0.5)
```

The output of this function is a list of 4 elements containing the following information:

- `Sew`: a numeric value indicating the weighted estimate of the sensitivity parameter.
- `tags`: list containing one element with the following information:
  - `tag.event`: a character string indicating the label used to indicate event of interest.
- `basics`: a list containing information of the following 6 elements:
  - `n`: a numeric value indicating the number of units in the data set.
  - `n.event`: a numeric value indicating the number of units in the data set with the event of interest.
  - `n.event.class`: a numeric value indicating the number of units in the data set with the event of interest that are correctly classified as events based on the selected cut-off point.
  - `hatN`: number of units in the population, represented by all the units in the data set, i.e., the sum of the sampling weights of the units in the data set.
  - `hatN.event`: number of units with the event of interest represented in the population by all the event units in the data set, i.e., the sum of the sampling weights of the units with the event of interest in the data set.
  - `hatN.event.class`: number of event units represented in the population by the event units in the data set that have been correctly classified as events based on the selected cut-off point, i.e., the sum of the sampling weights of the correctly classified event units in the data set.
- `call`: an object saving the information about the way in which the function has been run.

For example, the object `se.obj` contains the following information:

```
se.obj
```

```
$Sew
[1] 0.5207174

$tags
$tags$tag.event
[1] "1"


$basics
$basics$n
[1] 740

$basics$n.event
[1] 200

$basics$n.event.class
[1] 116

$basics$hatN
[1] 10000

$basics$hatN.event
[1] 2695

$basics$hatN.event.class
[1] 1403.333


$call
wse(response.var = "y",
    phat.var = "phat",
    weights.var = "weights",
```

```
    tag.event = 1,
    cutoff.value = 0.5,
    data = example_data_wroc)
```

8.2.3 `wocp()` function

Optimal cut-off points for individual classification can be calculated by means of the function `wocp()` in the context of complex survey data. This function is based on the package `OptimalCutpoints` (López-Ratón et al. 2014), which has been modified in order to consider sampling weights in the estimation process. The methods implemented in this function have been described in Section 7.2.2.

The usage of the function is described below and a summary of the arguments can be found in Table 8.8.

```
wocp(
  response.var,
  phat.var,
  weights.var = NULL,
  tag.event = NULL,
  tag.nonevent = NULL,
  method = c("Youden", "MaxProdSpSe", "ROC01", "MaxEfficiency"),
  data = NULL,
  design = NULL
)
```

Table 8.8: Summary and usage of the arguments incorporated to the function `wocp()`.

Argument	Description
`response.var`	A character string with the name of the column indicating the response variable in the data set or a vector (either numeric or character string) with information of the response variable for all the units.
`phat.var`	A character string with the name of the column indicating the estimated probabilities in the data set or a numeric vector containing estimated probabilities for all the units.

`weights.var`	A character string indicating the name of the column with sampling weights or a numeric vector containing information of the sampling weights. It could be `NULL` if the sampling design is indicated in the `design` argument.
`tag.event`	A character string indicating the label used for the event of interest in `response.var`. The default option is `tag.event = NULL`, which selects the class with the lowest number of units as event.
`tag.nonevent`	A character string indicating the label used for non-event in `response.var`. The default option is `tag.nonevent = NULL`, which selects the class with the greatest number of units as non-event.
`method`	A character string indicating the method to be used to select the optimal cut-off point. Choose one of the following methods (López-Ratón et al. 2014): `MaxProdSpSe, ROC01, Youden, MaxEfficiency`.
`data`	A data frame which, at least, must contain information on the columns `response.var`, `phat.var` and `weights.var`. If `data=NULL`, then specific numerical vectors must be included in `response.var`, `phat.var` and `weights.var` or the sampling design should be indicated in the argument `design`.
`design`	An object of class `survey.design` generated by `survey::svydesign()` indicating the complex sampling design of the data. If `design=NULL`, information on the data set (argument `data`) and/or sampling weights (argument `weights.var`) must be included.

A couple of examples of the usage of the function `wocp()` are given below and the resulting object is shown:

```
myocp <- wocp(response.var = "y",
              phat.var = "phat",
              weights.var = "weights",
              tag.event = 1,
              tag.nonevent = 0,
              method = "Youden",
```

```
              data = example_data_wroc)
```

Or equivalently:

```
myocp <- wocp(example_data_wroc$y,
              example_data_wroc$phat,
              example_data_wroc$weights,
              tag.event = 1,
              tag.nonevent = 0,
              method = "Youden")
```

The output of this function is an object of class `wocp`. This object is a list that contains information about the following 4 elements:

- `tags`: a list containing two elements with the following information:
  - `tag.event`: a character string indicating the event of interest.
  - `tag.nonevent`: a character string indicating the non-event.
- `basics`: a list containing information of the following 4 elements:
  - `n.event`: number of units with the event of interest in the data set.
  - `n.nonevent`: number of units without the event of interest in the data set.
  - `hatN.event`: number of units with the event of interest represented in the population by all the event units in the data set, i.e., the sum of the sampling weights of the units with the event of interest in the data set.
  - `hatN.nonevent`: a numeric value indicating the number of non-event units in the population represented by means of the non-event units in the data set, i.e., the sum of the sampling weights of the non-event units in the data set.
- `optimal.cutoff`: this object is a list of three elements containing the information described below:
  - `method`: a character string indicating the method implemented to select the optimal cut-off point.
  - `optimal`: a list containing information of the following four elements:

        * `cutoff`: a numeric vector indicating the optimal cut-off point(s) that optimize(s) the selected criterion.
        * `Sew`: a numeric vector indicating the estimated sensitivity parameter(s) corresponding to the optimal cut-off point(s) that optimize(s) the selected criterion.
        * `Spw`: a numeric vector indicating the estimated specificity parameter(s) corresponding to the optimal cut-off point(s) that optimize(s) the selected criterion.
        * `criterion`: a numeric value indicating the criterion value optimized by means of the selected optimal cut-off point(s).
    - `all`: a list containing information on the following four elements:
        * `cutoff`: a numeric vector indicating all the cut-off points considered.
        * `Sew`: a numeric vector indicating the estimated sensitivity parameters corresponding to all the considered cut-off points.
        * `Spw`: a numeric vector indicating the estimated sensitivity parameters corresponding to all the considered cut-off points.
        * `criterion`: a numeric vector indicating the values of the selected criterion corresponding to all the considered cut-off points.

- `call`: an object saving the information about the way in which the function has been run.

For example, the object `myocp` obtained above, contains the following information. In the object `$tags` we can check the labels with which the event of interest and the non-event are indicated:

```
myocp$tags
```

```
$tag.event
[1] "1"

$tag.nonevent
[1] "0"
```

Some basic information on the number of events and non-events (both, considering and not the sampling weights) is available in the object `$basics`:

```
myocp$basics
```

```
$n.event
[1] 200

$n.nonevent
[1] 540

$hatN.event
[1] 2695

$hatN.nonevent
[1] 7305
```

The object `$optimal.cutoff` contains the following information. First, the method implemented to obtain the optimal cut-off point is saved in the object `method`, i.e.,

```
myocp$optimal.cutoff$method
```

```
[1] "Youden"
```

The object `optimal` contains information on the optimal cut-off point, which can be accessed as follows:

```
myocp$optimal.cutoff$optimal
```

```
$cutoff
[1] 0.3271605

$Sew
[1] 0.737786

$Spw
[1] 0.8384668
```

```
$criterion
[1] 0.5762528
```

Finally, the information of all the cut-off points considered is in the object `all`, which is printed below (due to the large sizes of the vectors, only the first and last elements are printed in this document):

```
myocp$optimal.cutoff$all
```

```
$cutoff
  [1] 0.0003479797 0.0005718813 0.0006068810 0.0006551310
      ...
[737] 0.9769322384 0.9782241987 0.9830640592 0.9873385525

$Sew
  [1] 1.000000000 1.000000000 1.000000000 1.000000000 1.000000000
      ...
[736] 0.018552876 0.011131725 0.008658009 0.004947434 0.002473717

$Spw
  [1] 0.0000000000 0.0009126169 0.0054757016 0.0063883185
      ...
[737] 1.0000000000 1.0000000000 1.0000000000 1.0000000000

$criterion
  [1] 0.0000000000 0.0009126169 0.0054757016 0.0063883185
      ...
[737] 0.0111317254 0.0086580087 0.0049474335 0.0024737168
```

8.2.4 `wauc()` function

The function `wauc()` can be used to calculate the AUC of a logistic regression model considering sampling weights with complex survey data. This function uses the Mann-Whitney U-statistic with marginal sampling weights to calculate the AUC as defined in eq. (6.27) in Section 6.2.2.

The function can be used in the following way, which arguments are described in Table 8.9.

```
wauc(
  response.var,
  phat.var,
  weights.var = NULL,
  tag.event = NULL,
  tag.nonevent = NULL,
  data = NULL,
  design = NULL
)
```

Table 8.9: Summary and usage of the arguments incorporated to the function `wauc()`.

Argument	**Description**
`response.var`	A character string with the name of the column indicating the response variable in the data set or a vector (either numeric or character string) with information of the response variable for all the units.
`phat.var`	A character string with the name of the column indicating the estimated probabilities in the data set or a numeric vector containing estimated probabilities for all the units.
`weights.var`	A character string indicating the name of the column with sampling weights or a numeric vector containing information of the sampling weights. It could be `NULL` if the sampling design is indicated in the `design` argument.
`tag.event`	A character string indicating the label used for the event of interest in `response.var`. The default option is `tag.event = NULL`, which selects the class with the lowest number of units as event.
`tag.nonevent`	A character string indicating the label used for non-event in `response.var`. The default option is `tag.nonevent = NULL`, which selects the class with the greatest number of units as non-event.

`data`	A data frame which must incorporate information on the columns `response.var`, `phat.var` and `weights.var`. If `data=NULL`, then specific numerical vectors must be included in `response.var`, `phat.var` and `weights.var` or the sampling design should be indicated in the argument `design`.
`design`	An object of class `survey.design` generated by `survey::svydesign` indicating the complex sampling design of the data. If `NULL` information on the data and weights must be included in the argument `data` or as a vector in the argument `weights.var`.

For example, one possible usage of the function is the following one:

```
auc.obj <- wauc(response.var = "y",
                phat.var = "phat",
                weights.var = "weights",
                tag.event = 1,
                tag.nonevent = 0,
                data = example_data_wroc)
```

Or equivalently, the function can also be used in the following way, obtaining the same result:

```
auc.obj <- wauc(response.var = example_data_wroc$y,
                phat.var = example_data_wroc$phat,
                weights.var = example_data_wroc$weights,
                tag.event = 1,
                tag.nonevent = 0)
```

The output object of this function is a list of 4 elements containing the following information:

- `AUCw`: the weighted estimate of the AUC.
- `tags`: a list containing two elements with the following information:
  - `tag.event`: a character string indicating the event of interest.
  - `tag.nonevent`: a character string indicating the non-event.

- `basics`: a list containing information of the following 4 elements:
  - `n.event`: number of units with the event of interest in the data set.
  - `n.nonevent`: number of units without the event of interest in the data set.
  - `hatN.event`: number of units with the event of interest represented in the population by all the event units in the data set, i.e., the sum of the sampling weights of the units with the event of interest in the data set.
  - `hatN.nonevent`: a numeric value indicating the number of non-event units in the population represented by means of the non-event units in the data set, i.e., the sum of the sampling weights of the non-event units in the data set.
- `call`: an object saving the information about the way in which the function has been run.

For example, the object `auc.obj` obtained above contains the following information:

```
auc.obj
```

```
$AUCw
[1] 0.863269

$tags
$tags$tag.event
[1] "1"

$tags$tag.nonevent
[1] "0"


$basics
$basics$n.event
[1] 200

$basics$n.nonevent
```

```
[1] 540

$basics$hatN.event
[1] 2695

$basics$hatN.nonevent
[1] 7305


$call
wauc(response.var = "y",
     phat.var = "phat",
     weights.var = "weights",
     tag.event = 1,
     tag.nonevent = 0,
     data = example_data_wroc)
```

8.2.5 `wroc()` function

The function `wROC()` allows calculating all the information about the ROC curve of a logistic regression model considering sampling weights with complex survey data, which has been defined in eq. (6.18) in Section 6.2.2. Some basic information on the optimal cut-off points (obtained by means of the function `wocp()` described in detail in Section 8.2.3) can also be included to the output object.

In the following lines, we indicate the usage of the function and the information on the arguments is summarized in Table 8.10.

```
wroc(
  response.var,
  phat.var,
  weights.var = NULL,
  tag.event = NULL,
  tag.nonevent = NULL,
  data = NULL,
  design = NULL,
  cutoff.method = NULL
```

```
)
```

Table 8.10: Summary and usage of the arguments incorporated to the function `wroc()`.

Argument	**Description**
`response.var`	A character string with the name of the column indicating the response variable in the data set or a vector (either numeric or character string) with information of the response variable for all the units.
`phat.var`	A character string with the name of the column indicating the estimated probabilities in the data set or a numeric vector containing estimated probabilities for all the units.
`weights.var`	A character string indicating the name of the column with sampling weights or a numeric vector containing information of the sampling weights. It could be `NULL` if the sampling design is indicated in the `design` argument.
`tag.event`	A character string indicating the label used for the event of interest in `response.var`. The default option is `tag.event = NULL`, which selects the class with the lowest number of units as event.
`tag.nonevent`	A character string indicating the label used for non-event in `response.var`. The default option is `tag.nonevent = NULL`, which selects the class with the greatest number of units as non-event.
`data`	Data frame which must incorporate information on the columns `response.var`, `phat.var` and `weights.var`. If `data=NULL`, then specific numerical vectors must be included in `response.var`, `phat.var` and `weights.var` or the sampling design should be indicated in the argument `design`.
`design`	An object of class `survey.design` generated by `survey::svydesign` indicating the complex sampling design of the data. If `NULL` information on the data and weights must be included in the argument `data` or as a vector in the argument `weights.var`.

`cutoff.method`	A character string indicating the method to be used to select the optimal cut-off point. If `cutoff.method = NULL`, then no optimal cut-off point is drawn. If an optimal cut-off point is to be drawn, one of the following methods needs to be selected: `Youden`, `MaxProdSpSe`, `ROC01`, `MaxEfficiency`.

In the following lines, we give a couple of examples of how the function `wroc()` can be used. For example, we can set the information of the data set and indicate the column names for the response variable, predicted probabilities and sampling weights:

```
mycurve <- wroc(response.var = "y",
                phat.var = "phat",
                weights.var = "weights",
                data = example_data_wroc,
                tag.event = 1,
                tag.nonevent = 0,
                cutoff.method = "Youden")
```

Or equivalently, we can also use numeric vectors for these values, i.e.,

```
mycurve <- wroc(response.var = example_data_wroc$y,
                phat.var = example_data_wroc$phat,
                weights.var = example_data_wroc$weights,
                tag.event = 1,
                tag.nonevent = 0,
                cutoff.method = "Youden")
```

The output object of this function is a list of class `wroc`, which contains information about the weighted ROC curve of a logistic regression model and some of its components. In particular, this list contains a total of 5 or 6 elements (depending on the selected arguments) with the following information:

- `wroc.curve`: this element is a list that contains three numerical vectors. Specifically,
  - `Sew.values`: a vector of all the different values for the weighted estimate of the sensitivity across all the possible cut-off points.

  - `Spw.values`: a vector of all the different values for the weighted estimate of the specificity across all the possible cut-off points.
  - `cutoffs`: this vector contains all the cut-off points that have been considered to estimate sensitivity and specificity parameters.

- `wauc`: a numeric value indicating the area under the weighted estimate of the ROC curve. Actually, this value is calculated following eq. (6.26).

- `optimal.cutoff`: if the argument `cutoff.method != NULL`, this object is a list containing the 4 elements described below:
  - `method`: character string indicating the method implemented to calculate the optimal cut-off point as explained in Section 8.2.3.
  - `cutoff.value`: the optimal cut-off point value.
  - `Spw`: the weighted estimate of the specificity for the optimal cut-off point value (indicated in `cutoff.value`).
  - `Sew`: the weighted estimate of the sensitivity for the optimal cut-off point value (indicated in `cutoff.value`).

- `tags`: a list containing two elements with the following information:
  - `tag.event`: a character string indicating the event of interest.
  - `tag.nonevent`: a character string indicating the non-event.

- `basics`: a list containing information of the following 4 elements:
  - `n.event`: number of units with the event of interest in the data set.
  - `n.nonevent`: number of units without the event of interest in the data set.
  - `hatN.event`: number of units with the event of interest represented in the population by all the event units in the data set, i.e., the sum of the sampling weights of the units with the event of interest in the data set.
  - `hatN.nonevent`: a numeric value indicating the number of non-event units in the population represented by means of the non-event units in the data set, i.e., the sum of the sampling weights of the non-event units in the data set.

- `call`: an object saving the information about the way in which the function has been run.

For example, let us show one by one each element of the object `mycurve` obtained above. First, we can access to the information about the ROC curve as follows (due to the large size of the vectors, we only print information on the first and last elements):

```
mycurve$wroc.curve
```

```
$Sew.values
  [1] 0.000000000 0.002473717 0.004947434 0.008658009 0.011131725
      ...
[741] 1.000000000

$Spw.values
  [1] 1.0000000000 1.0000000000 1.0000000000 1.0000000000
      ...
[741] 0.0000000000

$cutoffs
  [1] 1.0000000000 0.9852013059 0.9806441289 0.9775782185
      ...
[741] 0.0000000000
```

The value of the area under the curve can be seen as follows:

```
mycurve$wauc
```

```
[1] 0.863269
```

Information related to the optimal cut-off point is shown below:

```
mycurve$optimal.cutoff
```

```
$method
[1] "Youden"

$cutoff.value
[1] 0.3271605

$Spw
[1] 0.8384668

$Sew
[1] 0.737786
```

Finally, the rest of the information saved in the object, which is mostly related to the data set, is given in the following lines:

```
$tags
$tags$tag.event
[1] "1"

$tags$tag.nonevent
[1] "0"


$basics
$basics$n.event
[1] 200

$basics$n.nonevent
[1] 540

$basics$hatN.event
[1] 2695

$basics$hatN.nonevent
[1] 7305
```

```
$call
wroc(response.var = "y", phat.var = "phat", weights.var = "weights",
    tag.event = 1, tag.nonevent = 0, data = example_data_wroc,
    cutoff.method = "Youden")

attr(,"class")
[1] "wroc"
```

8.2.6 `wroc.plot()` function

The function `wroc.plot()` allows to graph the information on an object of class `wroc`, obtained by means of the function `wroc()` as explained in Section 8.2.5.

Below, we show the usage of this function and the arguments are described in Table 8.11.

```
wroc.plot(
  x,
  print.auc = TRUE,
  print.cutoff = FALSE,
  col.cutoff = "red",
  cex.text = 0.75,
  round.digits = 4
)
```

Table 8.11: Summary and usage of the arguments incorporated to the function `wroc.plot()`.

Argument	Description
`x`	An object of class `wroc` obtained by means of the function `wroc`.
`print.auc`	A logical value. If `TRUE`, the value of the area under the ROCw curve (AUCw) is printed (default `print.auc = TRUE`).
`print.cutoff`	A logical value. If `TRUE`, the value of the optimal cut-off point, and the corresponding weighted estimates of the sensitivity and specificity parameters are printed (default `print.cutoff = TRUE`).
`col.cutoff`	A character string indicating the color in which the cut-off point is depicted. The default option is `col.cutoff = "red"`.

`cex.text`	A numerical value indicating the size with which the information of the AUCw and the optimal cut-off point is printed. The default option is `cex.text = 0.75`.
`round.digits`	A numeric value indicating the number of digits that will be employed when printing the information about the AUCw and optimal cut-off point. The default option is `round.digits = 4`.

The output value of this function is a plot. For example, if we take the object `mycurve`, obtained in Section 8.2.5, and we run the following function, we obtain the graph depicted in Figure 8.2.

```
wroc.plot(x = mycurve,
          print.auc = TRUE,
          print.cutoff = TRUE)
```

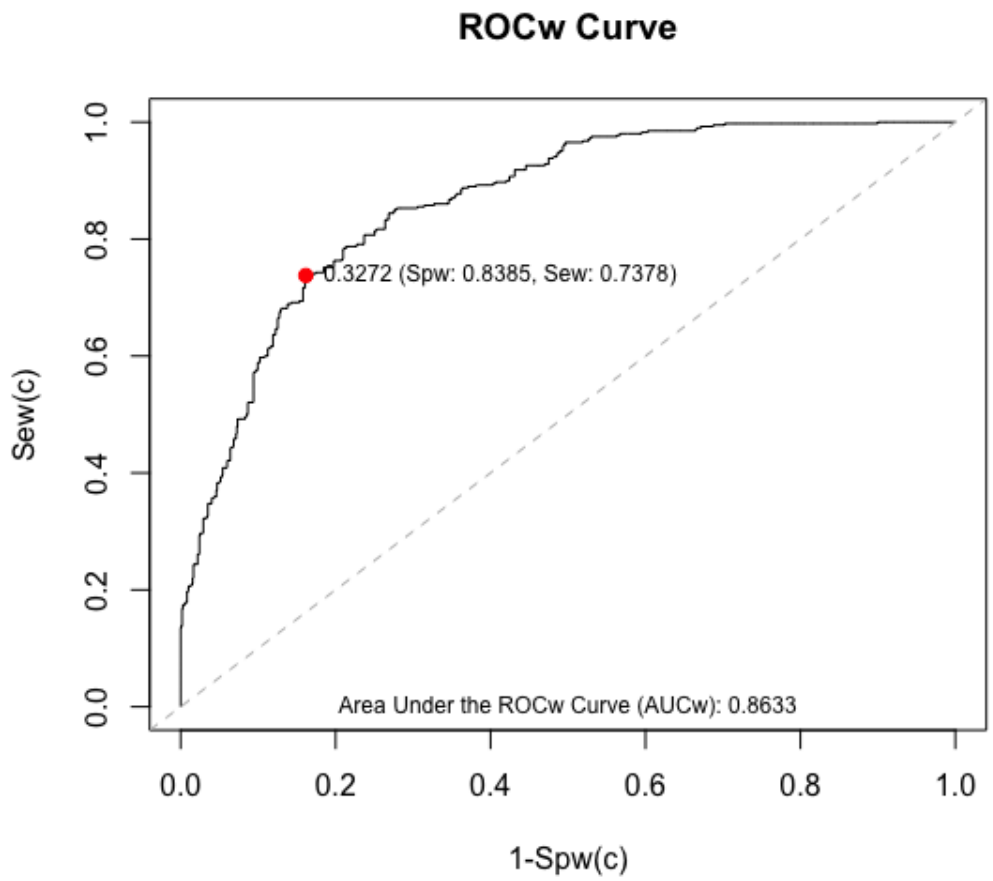


Figure 8.2: An example of the output graph obtained by means of the function `wroc.plot()`.

CHAPTER 9

Discussion

In this chapter, we describe the general conclusions and the main limitations of this Ph.D. thesis (Section 9.1). In addition, we present the research questions we aim to address in the future, which are summarized in Section 9.2. Finally, in Section 9.3, we briefly present the main contributions that emerged from this research work.

9.1 General conclusions and limitations

The main objective of this Ph.D. dissertation has been to make new proposals for the development and validation of prediction models for complex survey data. The different goals addressed in this dissertation were inspired by real problems we came across when analyzing real-life survey data provided by EUSTAT. The Ph.D. candidate and advisors are really grateful to EUSTAT for providing them with these data sets, letting them work with this data and for being the major source of inspiration to carry out this thesis.

We feel it is worth mentioning that this thesis has contributed to three impor-

tant aspects of statistical research: statistical proposal, software development, and application with social impact. From a statistical point of view, four contributions have been made and published in four scientific papers from high-impact journals in the category of "Statistics & Probability", each of them related to a different step of the development process of prediction models. Specifically, we contributed to: (1) model estimation process by means of a comparative simulation study by analyzing the performance of different existing estimation methods to estimate logistic regression model parameters with complex survey data (described in detail in Chapter 4); (2) variable selection by proposing a new design-based linear and logistic LASSO regression model estimation (Chapter 5); (3) discrimination ability of logistic regression models with design-based proposals for estimating the ROC curve and AUC (Chapter 6); and, (4) individual classification with new proposals that consider sampling weights for estimating optimal cut-off points (Chapter 7). In relation to the software development, all the above-mentioned statistical proposals have been incorporated into two R-packages that are freely available (`wlasso` and `wROC`, described in detail in Chapter 8), so that researchers and data analysts that deal with complex survey data in their daily practice can easily apply these advances in their analysis. Finally, as a significant contribution of this doctoral thesis at a practical level, we believe it is worth mentioning the social impact of this thesis, given that EUSTAT is currently publishing the results obtained by means of prediction models developed considering the proposals described throughout this dissertation on their official website[11,12].

Since the objective of this doctoral thesis has been to address the questions that emerged when developing prediction models for the data provided by EUSTAT, this dissertation has focused on complex survey data that meet certain characteristics that are worth mentioning. We describe these limitations in the following lines.

First, throughout this dissertation, we have considered sufficiently large sample sizes, which is a common situation when working with data related to official statistics. However, an important issue related to sample size is the difficulty of obtaining good estimates for those areas where not much data has been collected. This type of problem is known as Small Area Estimation (SAE) in the literature, and other techniques different from the ones considered in this dissertation have been developed in this field (see, e.g., Molina and Rao (2010), Rao and Molina (2015)). The problem

[11] https://www.eustat.eus/estadisticas/tema_150/opt_1/tipo_1/ti_encuesta-sobre-la-sociedad-de-la-informacion/temas.html

[12] https://www.eustat.eus/estadisticas/tema_37/opt_0/tipo_1/ti_poblacion-en-relacion-con-la-actividad-pra/temas.html

with some of these techniques is that they do not consider the sampling weights in the estimation of the models. It would be interesting to make a comparative study between the estimates obtained by prediction models that have been considered in this thesis and the traditional SAE techniques for this purpose in future work, particularly in the context of informative sampling designs.

Moreover, we have not worked with missing data since, in practice, we have not had to deal with them either when working with real data sets provided by EUSTAT. Therefore, this dissertation and related research have not considered the effect that missing values may have in this context. Nevertheless, this is an important issue in the context of complex surveys and should be analyzed further (see, e.g., Kalpourtzi et al. (2023)).

In this dissertation, we have focused exclusively on probability samples. There also exist other types of samples, known as non-probability samples in the literature (Vehovar et al. 2016). The main difference between probability and non-probability samples is that, in the latter, participants are not randomly selected, and hence, sampling probabilities cannot be calculated for those units. An increasingly popular example of non-probability samples is voluntary online surveys (Andrade 2020, Callegaro et al. 2015). Techniques for blending probability and non-probability samples are also emerging in the literature (see, e.g., Robbins et al. (2021), Rueda et al. (2023)). It would be interesting to try to develop techniques for fitting prediction models with non-probability samples in the future.

Besides, among the different types of probability samples, only two of them have been considered throughout this dissertation: the one-stage stratified sampling design and the two-stage stratified cluster sampling design. The sampling can be carried out in more than two sampling stages, which are usually known in the literature as multistage sampling designs (see, e.g., Kish (1965), Särndal et al. (2003)). Two-phase sampling designs are also common in the literature (Breslow et al. 2009, Neyman 1938, Saegusa and Wellner 2013, Särndal et al. 2003). In this kind of design, the sample obtained following a complex sampling design in phase one is sampled for the second time in phase two, which leads to smaller but more efficient samples given that information on more variables may be obtained for a reduced sample (see, e.g., Rivera-Rodriguez et al. (2019)). However, these kinds of designs have not been considered in this dissertation, and the results obtained are limited to the scenarios that have been drawn.

Similarly, it is worth noting that although two types of sampling designs have been used in this dissertation, we have not considered both of them in all chapters.

For example, in the simulation study carried out in Chapter 5 for variable selection, we only considered simulated data derived from two-stage stratified cluster sampling designs. However, the application of the methodology proposed is straightforward for data derived from one-stage stratified sampling designs. The package `wlasso` can also handle one-stage stratified samples. We expect the performance of different replicate weights methods to be similar to the scenario in which no cluster variables were considered, but a new simulation study should be performed in order to confirm this assumption.

In contrast, the two real surveys provided by EUSTAT that we started working with (ESIE described in Section 3.1 and PRA in Section 3.2) follow a one-stage stratified sampling design, so the research proposals in which we worked first in time (Chapter 4 and 7) only considered this sampling design. In the following paragraphs, we detail the limitations of the works described in these two chapters, focusing on the limitations of the considered sampling design and data sets.

In Chapter 4, a simulation study based on real survey data derived from one-stage stratified sampling designs was conducted to analyze the performance of three estimation methods to estimate the logistic regression parameters. The three methods that we considered in the simulation study are the unweighted logistic regression model, the weighted logistic regression model, and the unweighted mixed model. However, we believe it would be interesting to extend this simulation study to two-stage stratified cluster sampling designs, and we think that results may slightly differ from the ones shown in Chapter 4. For example, in line with the results shown in previous studies (see, e.g., Lumley and Scott (2017)), we believe that the weighted estimates may have a larger variance than the one we have observed. In addition, the mixed model has shown poor performance in the simulation study conducted in Chapter 4. However, the mixed model is more commonly used in the context of two-stage stratified cluster samples than in one-stage stratified samples. Thus, we believe it would be a fairer comparison for the mixed model to analyze its performance in two-stage samples.

Similarly, in the simulation study conducted in Chapter 7 for the estimation of optimal cut-off points, we only considered one-stage stratified samples. Nevertheless, the proposals made in this chapter can also be applied to two-stage stratified cluster samples, and the package `wROC` can also handle these sorts of samples. The performance of proposed methods in the context of two-stage stratified cluster sampling designs can be analyzed by replicating the simulation study carried out in Chapter 6, which considers both one-stage and two-stage sampling designs. The Ph.D. can-

didate and her advisors do not expect large differences between the results obtained from the two different sampling designs.

Lastly, even in the studies described in Chapters 4, 6 and 7, in which real surveys were considered, the selection of the data set used in each study differ. Let us explain the reason for these differences in the choice of the data used in each study.

In the simulation study carried out in Chapter 4, both surveys (ESIE and PRA) were considered in order to point out the differences in the performance of analyzed estimation methods in one-stage stratified samples. However, regarding the results obtained in this study, we concluded that the ESIE survey is more interesting so as to highlight the problems we may come across when working with complex survey data. Therefore, we only considered the ESIE survey in the studies carried out in Chapters 6 and 7.

Furthermore, even when selecting the ESIE survey, there are differences in the data sets we considered in different studies. In the studies described in Chapters 4 and 6, all the establishments of the BC were considered, which is in line with the models we fitted in practice for EUSTAT. In contrast, only the establishments with at least ten employees were considered in the study carried out in Chapter 7 related to the estimation of optimal cut-off points. The reason for conducting the study in this way is also related to a decision we made in practice. EUSTAT technicians suggested that we should obtain different cut-off points for those establishments with at least ten employees and small establishments with less than ten employees. The reason is that the behavior of these two types of establishments is different, and we could obtain better results by doing this analysis separately for both groups. It should be noted that, in this case, it makes sense to do the analysis separately for both groups since the number of employees is one of the variables used in the sampling design, so this subsample properly represents the corresponding population. However, the problems that may arise when obtaining the cut-off points separately according to a variable that has not been previously used for stratification should be further analyzed.

9.2 Further Research

Even though the initial objective of this dissertation has been fulfilled, we believe that there is still plenty of room for improvement in the development and validation of prediction models for complex survey data. Below, we summarize the open issues and new research lines that have emerged from this dissertation and the collaboration

with EUSTAT, which we aim to address in the future. In order to differentiate the different research lines more clearly, we separate the different future objectives according to the step of the development of prediction models to which they belong. Table 9.1 is displayed as a summary of these research ideas.

Some of the further research ideas described below are related to the transference of the proposed methods to real-life problems. For example, the Health Survey carried out by the Healthy Department of the Basque Country[13] is one of the surveys to which we would like to apply the methods developed in this dissertation. In addition, the Ph.D. candidate and her advisors have access to the population of COVID-19-positive patients in the Basque Country, which was collected during the pandemic (Portuondo-Jiménez et al. 2023). Having a real population data set allows us to study some properties based on a real population rather than in simulated data. In addition, we will be able to sample it following any sampling design we are interested in.

Estimation

We set below a list of issues related to the model estimation to be addressed in the future:

E1. Two-stage stratified cluster samples: a simulation study.
As explained in Section 9.1, we believe it would be interesting to replicate the simulation study carried out in Chapter 4 for real survey data derived from a two-stage stratified cluster sampling design and compare the performance of the methods analyzed in Chapter 4 in this context. For this purpose, we will use the above-mentioned population of COVID-19-positive patients in BC.

E2. Comparison to Small Area Estimation (SAE) techniques.
As discussed in Section 9.1, obtaining good estimates for areas in which small data is collected is a big challenge in the context of complex surveys. To compare the performance of different SAE techniques to the design-based prediction models considered in this dissertation is part of further research.

E3. Patient Reported Outcomes (PRO): Beta-binomial regression.
Nowadays, personalized healthcare is becoming increasingly relevant in clinical research, and the interest in patient-reported outcome (PRO) measurements is

[13] https://www.euskadi.eus/encuesta-salud/inicio/

growing. The beta-binomial regression models have been proposed in the literature for analyzing PROs (see, e.g., Arostegui et al. (2007), Najera-Zuloaga et al. (2018)). As far as we know, there is a lack of proposals to consider complex sampling designs in the estimation process of beta-binomial regression models. Therefore, another challenge that the Ph.D. candidate and her advisors aim to face in the future is to study the effect of complex sampling designs in the context of beta-binomial regression.

Variable selection

Several new research lines have emerged related to the variable selection proposal described in Chapter 5 after discussing the research done with other colleagues in several conferences at the national and international levels. Some of those proposals have been previously mentioned in that chapter and are now summarized below:

V1. Application and simulation study based on a real data set.
We aim to apply the methodological proposal described in Chapter 5 to a real data set. In this study, we aim to focus on the variables that end up in the final model and analyze whether the variables selected by our proposal make sense in daily practice or whether variables of high scientific interest are discarded from the model instead. This study will be carried out with the above-mentioned COVID-19-positive database, which will be sampled following different complex sampling designs. LASSO regression models have been developed to select relevant covariates in practice, so the models obtained with the methodology proposed in Chapter 5 will be compared to those that have shown clinical relevance in practice (Portuondo-Jiménez et al. 2023).

V2. Ridge regression and elastic nets.
As mentioned in Chapter 5, we aim to extend the methodology proposed to fit LASSO regression models to other types of models beyond LASSO, such as ridge regression (Hoerl and Kennard 1970, Kidwell and Brown 1982) and elastic nets (Zou and Hastie 2005). The validity of the proposals to fit those models will be analyzed by means of a simulation study.

V3. Statistical Boosting with complex survey data.
Nowadays, more advanced statistical techniques beyond LASSO can be used for variable selection. In particular, Statistical Boosting algorithms are being increasingly used in the last years in different modeling contexts (Mayr and

Hofner 2018, Mayr et al. 2017). As further research, we aim to study the adequacy of using Statistical Boosting with complex survey data and extend the proposals made in Chapter 5 to this context.

V4. Analysis of the design-effect: dCV vs w-SRSCV.
One of the issues we have pointed out in Chapter 5 is the differences observed between dCV and w-SRSCV, depending on the scenario. Specifically, when including cluster-level variables in the analysis, the performance of both methods differs. We believe that the inclusion of cluster-level variables leads to an increase in the effect of the sampling design, which leads to different performances of both methods. This would be in line with the results obtained by Lumley and Scott (2015), in which the effect of the sampling design has shown an important role in the model selection. In particular, Lumley and Scott (2015) propose to account for the design-effect by means of the trace of the variance-covariance matrix of the model coefficients to estimate the design-based AIC and BIC parameters. Given the similarities of both studies, we believe that the differences between dCV and w-SRSCV methods may be quantified by means of that trace. Thus, the magnitude of the relationship between the trace and the differences between the methods will be analyzed as further research.

Discrimination ability

In the context of the estimation of the discrimination ability of logistic regression, we aim to continue making proposals in the following directions (some of which have been previously mentioned in Chapter 6):

D1. Simulation study based on a real data set.
The simulation study conducted in Chapter 6 was based on artificial data. We believe that it would also be interesting to conduct a simulation study based on real survey data in order to analyze the performance of the proposed estimators in a more realistic scenario. This study will be carried out by sampling the COVID-19-positive population following one-stage and two-stage sampling designs.

D2. Optimism correction.
In the simulation study of Chapter 6, we have observed that the proposed weighted AUC estimator slightly overestimates the true population parameter.

As pointed out in Chapter 6, we believed that the optimism observed in the weighted estimates is due to the "overfitting" as the same data is used to fit the model and estimate the AUC (see, e.g., Steyerberg (2008)). We believe it would be interesting to analyze the performance of different replicate weights methods to correct for the optimism of the proposed AUC estimator in the context of complex survey data.

D3. Variance estimation.
Another interesting research line that emerged from Chapter 6 is the estimation of the variance of the proposed ROC curve and AUC estimators. On the one hand, we aim to analyze the validity of replicate weights methods such as the Jackknife Repeated Replication (JKn) and the Balanced Repeated Replication (BRR) (proposed by Yao et al. (2015)) as well as the Rescaling Bootstrap (used in the application of Chapter 6) to estimate the variance of those estimators. On the other hand, it would be interesting to obtain analytical expressions for the variance of the proposed AUC and ROC curve estimators. Comparison of the performance of different replicate weights methods to estimate the variance of the proposed estimators by means of a simulation study, as well as the analytical expression for the variance of those parameters, will be developed as future work.

D4. Optimal sampling design for the maximization of the AUC.
So far, we have focused on the development of prediction models for a given sample obtained based on some particular sampling design. However, another interesting question is how to define an optimal sampling design (see, e.g., Chen and Lumley (2022)). As a new research line, we aim to focus on the definition of optimal sampling designs for the optimization of some parameters, such as the maximization of the discrimination ability in terms of the AUC of the models to be fitted. Although we have worked on this slightly during the last few years (we have made one contribution to an international conference), we would like to analyze this further.

Estimation of optimal cut-off points

In the following lines, we summarize the further ideas we aim to address in the future regarding the estimation of optimal cut-off points:

C1. Two-stage stratified cluster samples: a simulation study.
The simulation study carried out in Chapter 6 will be extended to analyze the performance of the proposed methods for selecting the optimal cut-off points for individual classification under two-stage sampling designs.

C2. Optimal cut-off points based on a non-design categorical variable.
As stated in Section 9.1, in Chapter 7, we obtained optimal cut-off points for a subset of the whole population based on a categorical variable that was previously used as a stratification variable. However, the validity of obtaining different cut-off points for different categories of a variable that has not been used in the stratification process could be problematic due to the fact that the units in the sample subset may not properly represent the corresponding population. This issue will be analyzed as further research.

C3. Optimal categorization of continuous covariates.
However, beyond the individual classification, another context in which the estimation of optimal cut-off points plays an important role is the categorization of continuous covariates to be included in prediction models (Barrio et al. 2017). As further work, we would like to extend this methodology so that it can be applied to complex survey data.

Table 9.1: Summary and classification of further research lines (Task) based on the development process step to which they belong (Step), the type of contribution expected from each work (Contribution), and the level of challenge they suppose (Challenge).

Context	**Task**	**Contribution**				**Challenge**		
		Meth.	**Sim.**	**Appl.**	**Soft.**	**E**	**M**	**H**
Estimation	E1. Two-stage stratified cluster samples		✓			■		
	E2. Comparison to SAE techniques		✓	✓			■	
	E3. PRO - Beta-binomial regression	✓	✓	✓	✓			■
Variable selection	V1. Application to a real data set		✓	✓		■		
	V2. Ridge regression and elastic nets	✓	✓		✓	■		
	V3. Boosting	✓	✓		✓			■
	V4. Analysis of the design-effect	✓	✓					■
Discrimination ability	D1. Simulation study with real data		✓	✓		■		
	D2. Optimism correction	✓	✓		✓		■	
	D3. Variance estimation	✓	✓		✓			■
	D4. Maximization of the AUC	✓	✓					■
Cut-off points	C1. Two-stage stratified cluster samples		✓	✓		■		
	C2. Cut-offs with non-design variables	✓	✓	✓	✓		■	
	C3. Categorization of covariates	✓	✓	✓	✓		■	

Meth.: methodological proposal, **Sim.**: simulation study, **Appl.**: application, **Soft.**: software development, **E**: easy, **M**: medium, **H**: high.

9.3 Main Contributions

In this section, we summarize the information about the main contributions raised from this Ph.D. thesis, including scientific papers, developed software, the most important contributions to prestigious national and international conferences, and the most remarkable grants and awards received by the Ph.D. candidate directly related to this dissertation.

Research articles

Iparragirre, A., Barrio, I., Aramendi, J. & Arostegui, I. (2022). Estimation of cut-off points under complex-sampling design data. *SORT-Statistics and Operations Research Transactions*, 46(1), 137–158.

Iparragirre, A., Lumley, T., Barrio, I., & Arostegui, I. (2023). Variable selection with LASSO regression for complex survey data. *Stat*, 12(1), e578.

Iparragirre, A., Barrio, I., & Arostegui, I. (2023). Estimation of the ROC curve and the area under it with complex survey data. *Stat*, 12(1), e635.

Iparragirre, A., Barrio, I., Aramendi, J., & Arostegui, I. (2024) Estimation of logistic regression parameters for complex survey data: simulation study based on real survey data. *SORT - Statistics and Operations Research Transactions*, (in press).

Software

1. `wlasso` R-package: https://github.com/aiparragirre/wlasso
2. `wROC` R-package: https://github.com/aiparragirre/wROC

Invitied contributions

1. On the development of prediction models for complex survey data. Research Meeting, Department of Statistics, University of Auckland. Auckland, September 2022.

2. Dealing with sampling weights on the development of prediction models for complex survey data. Grup de Recerca en Bioestadística i Bioinformatica (GRBIO). Barcelona, June 2021.

3. *10th Conference of the Eastern Mediterranean Region of the International Biometric Society (EMR-IBS)*. Development and validation of prediction models with complex survey data: from the outset to new proposals. Iparragirre A, Barrio I, Arostegui I. Jerusalem, December 2018.

Contributions

1. *16th International Conference of the ERCIM WG on Computational and Methodological Statistics*. On Lasso regression for complex survey data. A new replicate weights cross-validation proposal. Iparragirre A, Lumley T, Barrio I, Arostegui I. Berlin, December 2023.

2. *XIX Conferencia Española y VIII Encuentro Iberoamericano de Biometría*. Variable selection with LASSO regression for complex survey data. Iparragirre A, Lumley T, Barrio I, Arostegui I. Vigo, June 2023.

3. *XXXI International Biometric Conference*. Optimal sampling to Maximize the Predictive Performance of Logistic Regression Models for Complex Survey Data. Iparragirre A, Barrio I, Gómez-Melis G. Riga, July 2022.

4. *XXXIX Congreso Nacional de Estadística e Investigación Operativa y XIII Jornadas de Estadística Pública*. AUC estimation proposal under complex survey data. Iparragirre A, Barrio I., Aramendi J, Arostegui I. Granada, June 2022.

5. *XVIII Congreso de Biometría*. Estimation of the area under the ROC curve with complex survey data. Iparragirre A, Barrio I, Arostegui I. Madrid, May 2022.

6. *V Congreso de Jóvenes investigadores en Diseño de Experimentos y Bioestadística*. Estimation of logistic regression model coefficients for complex survey data: real data based simulation study. Iparragirre A, Barrio I, Arostegui I. Almería, November 2021.

7. *30th International Biometric Conference*. Estimating Logistic Regression Parameters for Complex Survey Data: a Comparative Study. Iparragirre A,

Barrio I, Arostegui I. Online, November 2020.

8. *XVII Conferencia Española y VII Encuentro Iberoamericano de Biometría (CEB-EIB 2019).* Dealing with missing predictor variables in logistic regression models with complex survey data. Iparragirre A, Barrio I, Aramendi J, Arostegui I. Valencia, June 2019.

9. *XXXVII Congreso Nacional de Estadística e Investigación Operativa y XI Jornadas de Estadística Pública.* Modeling probabilities for complex survey data. Iparragirre A, Barrio I, Arostegui I. Oviedo, June 2018.

Research stays

1. Department of Statistics, The University of Auckland. Auckland, July 2022-October 2022.

2. Department of Statistics and Operations Research, Universitat Politecnica de Catalunya. Barcelona, November 2021 - January 2022.

3. Department of Statistics and Operations Research, Universitat Politecnica de Catalunya. Barcelona, June 2021.

Grants and awards

1. Award for the best work presented by a young researcher for the work entitled: *Estimation of the area under the ROC curve with complex survey data.*. XVIII Congreso de Biometría. Madrid, May 2022.

2. Grant from BIOSTATNET for a two-weeks research stay in the Universitat Politecnica de Catalunya. Barcelona, June 2021.

3. Predoctoral grant (PIF18/213). University of the Basque Country UPV/EHU. June 2019.

References

Airola, A., Pahikkala, T., Waegeman, W., De Baets, B., and Salakoski, T. (2011). An experimental comparison of cross-validation techniques for estimating the area under the ROC curve. *Computational Statistics & Data Analysis*, 55(4):1828–1844.

Andrade, C. (2020). The Limitations of Online Surveys. *Indian Journal of Psychological Medicine*, 42(6):575–576.

Archer, K. J., Lemeshow, S., and Hosmer, D. W. (2007). Goodness-of-fit tests for logistic regression models when data are collected using a complex sampling design. *Computational Statistics & Data Analysis*, 51(9):4450–4464.

Arostegui, I., Legarreta, M. J., Barrio, I., Esteban, C., Garcia-Gutierrez, S., Aguirre, U., and Quintana, J. M. (2019). A Computer Application to Predict Adverse Events in the Short-Term Evolution of Patients With Exacerbation of Chronic Obstructive Pulmonary Disease. *JMIR Medical Informatics*, 7(2):e10773.

Arostegui, I., Núñez-Antón, V., and Quintana, J. M. (2007). Analysis of the short form-36 (SF-36): the beta-binomial distribution approach. *Statistics in Medicine*, 26(6):1318–1342.

Austin, P. C. and Steyerberg, E. W. (2017). Events per variable (EPV) and the relative performance of different strategies for estimating the out-of-sample validity of logistic regression models. *Statistical Methods in Medical Research*, 26(2):796–808.

Baker, T. and Gerdin, M. (2017). The clinical usefulness of prognostic prediction models in critical illness. *European Journal of Internal Medicine*, 45:37–40.

Bamber, D. (1975). The area above the ordinal dominance graph and the area below the receiver operating characteristic graph. *Journal of Mathematical Psychology*, 12(4):387–415.

Barrio, I., Arostegui, I., Rodríguez-Álvarez, M.-X., and Quintana, J.-M. (2017). A new approach to categorising continuous variables in prediction models: Proposal and validation. *Statistical Methods in Medical Research*, 26(6):2586–2602.

Bates, D., Mächler, M., Bolker, B., and Walker, S. (2015). Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67(1):1–48.

Binder, D. A. (1981). On the variances of asymptotically normal estimators from complex surveys. *Survey Methodology*, 7(3):157–170.

Binder, D. A. (1983). On the variances of asymptotically normal estimators from complex surveys. *International Statistical Review*, 51(3):279–292.

Binder, D. A. and Roberts, G. (2009). Design- and Model-Based Inference for Model Parameters. In *Handbook of Statistics*, volume 29, pages 33–54. Elsevier, Amsterdam.

Breslow, N. E., Lumley, T., Ballantyne, C. M., Chambless, L. E., and Kulich, M. (2009). Improved Horvitz–Thompson Estimation of Model Parameters from Two-phase Stratified Samples: Applications in Epidemiology. *Statistics in Biosciences*, 1(1):32–49.

Brewer, K. R. W. and Mellor, R. W. (1973). The effect of sample structure on analytical surveys. *Australian Journal of Statistics*, 15(3):145–152.

Callegaro, M., Manfreda, K. L., and Vehovar, V. (2015). *Web Survey Methodology*. SAGE Publications Ltd, London.

Canty, A. J. and Davison, A. C. (1999). Resampling-based Variance Estimation for Labour Force Surveys. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 48(3):379–391.

Chambers, R. L. and Skinner, C. J. (2003). *Analysis of Survey Data*. John Wiley & Sons, New York.

Chambless, L. E. and Boyle, K. E. (1985). Maximum likelihood methods for complex sample data: logistic regression and discrete proportional hazards models. *Communications in Statistics-Theory and Methods*, 14(6):1377–1392.

Chen, J.-Y., Feng, J., Wang, X.-Q., Cai, S.-W., Dong, J.-H., and Chen, Y.-L. (2015). Risk scoring system and predictor for clinically relevant pancreatic fistula after pancreaticoduodenectomy. *World Journal of Gastroenterology*, 21(19):5926–5933.

Chen, T. and Lumley, T. (2022). Optimal sampling for design-based estimators of regression models. *Statistics in Medicine*, 41(8):1482–1497.

Cochran, W. G. (1977). *Sampling Techniques*. John Wiley & Sons, New York.

Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46.

Copas, J. B. (2002). Overestimation of the receiver operating characteristic curve for logistic regression. *Biometrika*, 89(2):315–331.

Cox, D. and Snell, E. J. (1991). *Analysis of Binary Data*. Chapman & Hall/CRC, London, 2 edition.

DeMets, D. and Halperin, M. (1977). Estimation of a Simple Regression Coefficient in Samples Arising from a Sub-Sampling Procedure. *Biometrics*, 33(1):47–56.

Diggle, P., Liang, K.-Y., and Zeger, S. L. (2002). *Analysis of Longitudinal Data*. Oxford University Press, Oxford, 2 edition.

Efron, B. and Tibshirani, R. (1994). *An Introduction to the Bootstrap*. Chapman and Hall/CRC, New York.

Fisher, R. A. (1992). Statistical Methods for Research Workers. In *Breakthroughs in Statistics: Methodology and Distribution*, pages 66–70. Springer, New York.

Fisher, S., Bennett, C., Hennessy, D., Robertson, T., Leyland, A., Taljaard, M., Sanmartin, C., Jha, P., Frank, J., Tu, J. V., Rosella, L. C., Wang, J., Tait, C., and Manuel, D. G. (2020). International population-based health surveys linked to outcome data: A new resource for public health and epidemiology. *Health Reports*, 31(7):12–23.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, 33(1):1–22.

Fuller, W. A. (1975). Regression analysis for sample survey. *Sankhya*, 37(3):117–132.

Gelman, A. (2007). Struggles with survey weighting and regression modeling. *Statistical Science*, 22(2):153–164.

Green, D. M. and Swets, J. A. (1966). *Signal Detection Theory and Psychophysics*. John Wiley & Sons, New York.

Greiner, M. (1995). Two-graph receiver operating characteristic (TG-ROC): a Microsoft-EXCEL template for the selection of cut-off values in diagnostic tests. *Journal of Immunological Methods*, 185(1):145–146.

Greiner, M. (1996). Two-graph receiver operating characteristic (TG-ROC): update version supports optimisation of cut-off values that minimise overall misclassification costs. *Journal of Immunological Methods*, 191(1):93–94.

Greiner, M., Pfeiffer, D., and Smith, R. (2000). Principles and practical application of the receiver-operating characteristic analysis for diagnostic tests. *Preventive Veterinary Medicine*, 45(1-2):23–41.

Guisan, A., Tingley, R., Baumgartner, J. B., Naujokaitis-Lewis, I., Sutcliffe, P. R., Tulloch, A. I. T., Regan, T. J., Brotons, L., McDonald-Madden, E., Mantyka-Pringle, C., Martin, T. G., Rhodes, J. R., Maggini, R., Setterfield, S. A., Elith, J., Schwartz, M. W., Wintle, B. A., Broennimann, O., Austin, M., Ferrier, S., Kearney, M. R., Possingham, H. P., and Buckley, Y. M. (2013). Predicting species distributions for conservation decisions. *Ecology Letters*, 16(12):1424–1435.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer, New York, 2 edition.

Hausman, J. A. and Wise, D. A. (1981). Stratification on endogenous variables and estimation: The Gary income maintenance experiment. In *Structural Analysis of Discrete Data with Econometric Applications*, pages 365–391. MIT Press, Cambridge.

Heeringa, S. G., West, B. T., and Berglund, P. A. (2017). *Applied Survey Data Analysis*. Chapman and Hall/CRC, Boca Raton.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1):55.

Holt, D., Smith, T. M. F., and Winter, P. D. (1980). Regression analysis of data from complex surveys. *Journal of the Royal Statistical Society: Series A (General)*, 143(4):474–487.

Horvitz, D. G. and Thompson, D. J. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47(260):663–685.

Hosmer, D. W. and Lemesbow, S. (1980). Goodness of fit tests for the multiple logistic regression model. *Communications in Statistics - Theory and Methods*, 9(10):1043–1069.

Hosmer, D. W. and Lemeshow, S. (2000). *Applied Logistic Regression*. John Wiley & Sons, Hoboken.

Iparragirre, A., Barrio, I., and Rodríguez-Álvarez, M. X. (2019). On the optimism correction of the area under the receiver operating characteristic curve in logistic prediction models. *SORT - Statistics and Operations Research Transactions*, 43(1):145–162.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning*. Springer, New York.

Kaier, A. (1895). Observations et experiences concernant des dénombrements representatives. *Bulletin of the International Statistical Institute*, 9:176–183.

Kalpourtzi, N., Carpenter, J. R., and Touloumi, G. (2023). Handling Missing Values in Surveys With Complex Study Design: A Simulation Study. *Journal of Survey Statistics and Methodology*, smac039.

Kalton, G. (1983). *Introduction to Survey Sampling*. SAGE Publications Ltd, Thousand Oaks.

Kaufman, S., Rosset, S., and Perlich, C. (2011). Leakage in Data Mining: Formulation, Detection, and Avoidance. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 556–563, New York. Association for Computing Machinery.

Kidwell, J. S. and Brown, L. H. (1982). Ridge Regression as a Technique for Analyzing Models with Multicollinearity. *Journal of Marriage and the Family*, 44(2):287–299.

Kish, L. (1965). *Survey Sampling*. John Wiley & Sons, New York.

Korn, E. L. and Graubard, B. I. (1995). Examples of differing weighted and unweighted estimates from a sample survey. *The American Statistician*, 49(3):291–295.

Kott, P. S. (2012). Why one should incorporate the design weights when adjusting for unit nonresponse using response homogeneity groups. *Survey Methodology*, 38(1):95–99.

Kshirsagar, V., Wieczorek, J., Ramanathan, S., and Wells, R. (2017). Household poverty classification in data-scarce environments: a machine learning approach. In *31st Conference on Neural Information Processing Systems*, Long Beach. arXiv:1711.06813.

Kuhn, M. and Johnson, K. (2019). *Feature Engineering and Selection: A Practical Approach for Predictive Models*. Chapman and Hall/CRC, Boca Raton.

Lee, Y. and Nelder, J. A. (2001). Hierarchical generalised linear models: A synthesis of generalised linear models, random-effect models and structured dispersions. *Biometrika*, 88(4):987–1006.

Lee, Y. and Nelder, J. A. (2004). Conditional and Marginal Models: Another View. *Statistical Science*, 19(2):219–238.

Lewis, J. D., Chuai, S., Nessel, L., Lichtenstein, G. R., Aberra, F. N., and Ellenberg, J. H. (2008). Use of the noninvasive components of the mayo score to assess clinical response in Ulcerative Colitis. *Inflammatory Bowel Diseases*, 14(12):1660–1666.

Lewis, T. H. (2016). *Complex Survey Data Analysis with SAS*. Chapman & Hall/CRC, New York.

Li, X., Ergu, D., Zhang, D., Qiu, D., Cai, Y., and Ma, B. (2022). Prediction of loan default based on multi-model fusion. *Procedia Computer Science*, 199:757–764.

Li, Y., Sun, M., Zhang, C., Zhang, Y., Xu, B., Ren, Y., and Chen, Y. (2020). Evaluating fisheries conservation strategies in the socio-ecological system: A grid-based dynamic model to link spatial conservation prioritization tools with tactical fisheries management. *PLOS ONE*, 15(4):e0230946.

Lie, E. (2002). The rise and fall of sampling surveys in Norway, 1875-1906. *Science in Context*, 15(3):385–409.

Little, R. and Vartivarian, S. (2003). On weighting the rates in non-response weights. *Statistics in Medicine*, 22(9):1589–1599.

López-Ratón, M., Rodríguez-Álvarez, M. X., Suárez, C. C., and Sampedro, F. G. (2014). OptimalCutpoints: An R Package for Selecting Optimal Cutpoints in Diagnostic Tests. *Journal of Statistical Software*, 61(8):1–36.

Lumley, T. (2010). *Complex Surveys: A Guide to Analysis Using R*. John Wiley & Sons, Hoboken.

Lumley, T. (2017). Pseudo-R2 statistics under complex sampling. *Australian & New Zealand Journal of Statistics*, 59(2):187–194.

Lumley, T. (2020). survey: analysis of complex survey samples.

Lumley, T. and Huang, X. (2023a). Linear mixed models for complex survey data: implementing and evaluating pairwise likelihood. *arXiv:2307.04944*, pages 1–13.

Lumley, T. and Huang, X. (2023b). Weighted composite likelihood for linear mixed models in complex samples. *arXiv:2307.04944*, pages 1–29.

Lumley, T. and Scott, A. (2015). AIC and BIC for modeling with complex survey data. *Journal of Survey Statistics and Methodology*, 3(1):1–18.

Lumley, T. and Scott, A. (2017). Fitting regression models to survey data. *Statistical Science*, 32(2):265–278.

Magder, L. (2003). Optimal choice of a cut point for a quantitative diagnostic test performed for research purposes. *Journal of Clinical Epidemiology*, 56(10):956–962.

Manel, S., Williams, H. C., and Ormerod, S. (2001). Evaluating presence-absence models in ecology: the need to account for prevalence. *Journal of Applied Ecology*, 38(5):921–931.

Masood, M., Newton, T., and Reidpath, D. (2016). Comparison of four analytic strategies for complex survey data: a case-study of Spanish data. *Epidemiology, Biostatistics and Public Health*, 13(1):1–7.

Mayr, A. and Hofner, B. (2018). Boosting for statistical modelling-A non-technical introduction. *Statistical Modelling*, 18(3-4):365–384.

Mayr, A., Hofner, B., Waldmann, E., Hepp, T., Meyer, S., and Gefeller, O. (2017). An Update on Statistical Boosting in Biomedicine. *Computational and Mathematical Methods in Medicine*, 2017:1–12.

McCarthy, P. J. (1966). Replication: an approach to the analysis of data from complex surveys. *Vital and Health Statistics. Series 2, Data Evaluation and Methods Research*, (14):1–38.

McConville, K. S., Breidt, F. J., Lee, T., and Moisen, G. G. (2017). Model-assisted survey regression estimation with the lasso. *Journal of Survey Statistics and Methodology*, 5(2):131–158.

McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models*. Chapman & Hall/CRC, London, 2 edition.

Merkle, E. C., Furr, D., and Rabe-Hesketh, S. (2019). Bayesian Comparison of Latent Variable Models: Conditional Versus Marginal Likelihoods. *Psychometrika*, 84(3):802–829.

Metz, C. E. (1978). Basic principles of ROC analysis. *Seminars in Nuclear Medicine*, 8(4):283–298.

Molina, I. and Rao, J. N. K. (2010). Small area estimation of poverty indicators. *The Canadian Journal of Statistics*, 38(3):369–385.

Nagelkerke, N. J. D. (1991). A note on a general definition of the coefficient of determination. *Biometrika*, 78(3):691–692.

Najera-Zuloaga, J., Lee, D.-J., and Arostegui, I. (2018). Comparison of beta-binomial regression model approaches to analyze health-related quality of life data. *Statistical Methods in Medical Research*, 27(10):2989–3009.

Nathan, G. and Holt, D. (1980). The effect of survey design on regression analysis. *Journal of the Royal Statistical Society: Series B (Methodological)*, 42(3):377–386.

Neyman, J. (1934). On the Two Different Aspects of the Representative Method: The Method of Stratified Sampling and the Method of Purposive Selection. *Journal of the Royal Statistical Society*, 97(4):558–625.

Neyman, J. (1938). Contribution to the Theory of Sampling Human Populations. *Journal of the American Statistical Association*, 33(201):101–116.

Pauker, S. G. and Kassirer, J. P. (1980). The threshold approach to clinical decision making. *New England Journal of Medicine*, 302(20):1109–1117.

Pepe, M. S. (2003). *The Statistical Evaluation of Medical Tests for Classification and Prediction*. Oxford University Press, Oxford.

Pfeffermann, D. (1993). The role of sampling weights when modeling survey data. *International Statistical Review/Revue Internationale de Statistique*, 61(2):317–337.

Pfeffermann, D. and Sverchkov, M. (2009). Inference under informative sampling. In *Handbook of Statistics*, volume 29, pages 455–487. Elsevier, Amsterdam.

Portuondo-Jiménez, J., Barrio, I., España, P. P., García, J., Villanueva, A., Gascón, M., Rodríguez, L., Larrea, N., García-Gutierrez, S., and Quintana, J. M. (2023). Clinical prediction rules for adverse evolution in patients with COVID-19 by the Omicron variant. *International Journal of Medical Informatics*, 173:105039.

R Core Team (2022). R: A Language and Environment for Statistical Computing.

Rao, J. and Molina, I. (2015). *Small Area Estimation*. John Wiley & Sons, Hoboken, 2 edition.

Rao, J. N. K. and Wu, C. F. J. (1988). Resampling Inference With Complex Survey Data. *Journal of the American Statistical Association*, 83(401):231–241.

Reiter, J. P., Zanutto, E. L., and Hunter, L. W. (2005). Analytical modeling in complex surveys of work practices. *Industrial and Labor Relations Review*, 59(1):82–100.

Rivera-Rodriguez, C., Spiegelman, D., and Haneuse, S. (2019). On the analysis of two-phase designs in cluster-correlated data settings. *Statistics in Medicine*, 38(23):4611–4624.

Robbins, M. W., Ghosh-Dastidar, B., and Ramchand, R. (2021). Blending Probability and Nonprobability Samples with Applications to a Survey of Military Caregivers. *Journal of Survey Statistics and Methodology*, 9(5):1114–1145.

Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J.-C., and Müller, M. (2011). pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinformatics*, 12:77.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3):581–592.

Rueda, M. M., Pasadas-del-Amo, S., Rodríguez, B. C., Castro-Martín, L., and Ferri-García, R. (2023). Enhancing estimation methods for integrating probability and nonprobability survey samples with machine-learning techniques. An application to a Survey on the impact of the COVID-19 pandemic in Spain. *Biometrical Journal*, 65:2200035.

Saegusa, T. and Wellner, J. A. (2013). Weighted likelihood estimation under two-phase sampling. *The Annals of Statistics*, 41(1):269–295.

Sanchez, G. and Marzban, E. (2020). *All Models Are Wrong: Concepts of Statistical Learning*.

Särndal, C.-E., Swensson, B., and Wretman, J. (2003). *Model Assisted Survey Sampling*. Springer, New York.

Scott, A. J. (1977). On the problem of randomization in survey sampling. *Sankhya*, 39:1–9.

Scott, A. J. and Wild, C. J. (1986). Fitting logistic models under case-control or choice based sampling. *Journal of the Royal Statistical Society: Series B (Methodological)*, 48(2):170–182.

Scott, A. J. and Wild, C. J. (2002). On the robustness of weighted methods for fitting models to case–control data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(2):207–219.

Skinner, C. J., Holt, D., and Smith, T. M. F. (1989). *Analysis of Complex Surveys*. John Wiley & Sons, New York.

Smith, T. M. F. (1981). Regression analysis for complex surveys. In *Current Topics in Survey Sampling*, pages 267–292. Academic Press, New York.

Smith, T. M. F. (1988). To weight or not to weight, that is the question. *Bayesian Statistics*, 3:437–451.

Spence, R. T., Chang, D. C., Kaafarani, H. M. A., Panieri, E., Anderson, G. A., and Hutter, M. M. (2018). Derivation, Validation and Application of a Pragmatic Risk Prediction Index for Benchmarking of Surgical Outcomes. *World Journal of Surgery*, 42(2):533–540.

Steyerberg, E. W. (2008). *Clinical Prediction Models: A Practical Approach to Development, Validation, and Updating*. Springer, New York.

Steyerberg, E. W., Harrell, F. E., Borsboom, G. J., Eijkemans, M., Vergouwe, Y., and Habbema, J. F. (2001). Internal validation of predictive models. *Journal of Clinical Epidemiology*, 54(8):774–781.

Steyerberg, E. W., Marshall, P. B., Keizer, H. J., and Habbema, J. D. (1999). Resection of small, residual retroperitoneal masses after chemotherapy for nonseminomatous testicular cancer: a decision analysis. *Cancer*, 85(6):1331–1341.

Steyerberg, E. W. and Vergouwe, Y. (2014). Towards better clinical prediction models: seven steps for development and an ABCD for validation. *European Heart Journal*, 35(29):1925–1931.

Sugden, R. A. and Smith, T. M. F. (1984). Ignorable and informative designs in survey sampling inference. *Biometrika*, 71(3):495–506.

Swets, J. A. (1992). The science of choosing the right decision threshold in high-stakes diagnostics. *American Psychologist*, 47(4):522–532.

Swets, J. A. and Pickett, R. M. (1982). *Evaluation of Diagnostic Systems: Methods from Signal Detection Theory*. Academic Press, New York.

Tellez-Plaza, M., Briongos-Figuero, L., Pichler, G., Dominguez-Lucas, A., Simal-Blanco, F., Mena-Martin, F. J., Bellido-Casado, J., Arzua-Mouronte, D., Chaves, F. J., Redon, J., and Martin-Escudero, J. C. (2019). Cohort profile: the Hortega Study for the evaluation of non-traditional risk factors of cardiometabolic and other chronic diseases in a general population from Spain. *BMJ Open*, 9(6):e024073.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Tsuruta, H. and Bax, L. (2006). Polychotomization of continuous variables in regression models based on the overall C index. *BMC Medical Informatics and Decision Making*, 6(1):41.

Vehovar, V., Toepoel, V., and Steinmetz, S. (2016). Non-probability Sampling. In *The SAGE Handbook of Survey Methodology*, chapter 22, pages 329–346. SAGE Publications Ltd, London.

Vermont, J., Bosson, J., François, P., Robert, C., Rueff, A., and Demongeot, J. (1991). Strategies for graphical threshold determination. *Computer Methods and Programs in Biomedicine*, 35(2):141–150.

Wieczorek, J., Guerin, C., and McMahon, T. (2022). K-fold cross-validation for complex sample surveys. *Stat*, 11(1):e454.

Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, 4 edition.

Wolter, K. (2007). *Introduction to Variance Estimation*. Springer, New York, 2 edition.

Wynants, L., van Smeden, M., McLernon, D. J., Timmerman, D., Steyerberg, E. W., Van Calster, B., and others (2019). Three myths about risk thresholds for prediction models. *BMC Medicine*, 17(192).

Yao, W., Li, Z., and Graubard, B. I. (2015). Estimation of ROC curve with complex survey data. *Statistics in Medicine*, 34(8):1293–1303.

Youden, W. J. (1950). Index for rating diagnostic tests. *Cancer*, 3(1):32–35.

Zou, H. and Hastie, T. (2005). Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 67(2):301–320.