Treatment of unit nonresponse in surveys through machine learning methods : an empirical comparison.

Khaled Larbi (*), David Haziza (**), Mehdi Dagdoug (***)

(*) Insee / Ensae
(**) University of Ottawa
(***) Université de Bourgogne France-Comté

khaled.larbi@insee.fr
dhaziza@uottawa.ca
mohamed_mehdi.dagdoug@univ-fcomte.fr

Mots-clés. Non réponse totale, apprentissage automatique, calage, simulation, échantillonnage.

Domaines. Non réponse, Machine Learning / apprentissage automatique, classification.

Résumé

Ces dernières années, l'apprentissage automatique a suscité un intérêt considérable dans les offices nationaux de statistique. Grâce à leur flexibilité, ces méthodes peuvent s'avérer utiles au stade du traitement de la non-réponse totale. Dans cet article, nous menons une étude par simulation afin de comparer plusieurs procédures d'apprentissage automatique en termes de biais et d'efficacité. En plus des approches classiques d'apprentissage automatique, nous évaluons la performance de certaines approches d'aggrégation qui utilisent difféntes procédures d'apprentissage automatique pour produire un ensemble de poids ajusté pour la non-réponse.

Abstract

In recent years, there has been a significant interest in machine learning in national statistical offices. Thanks to their flexibility, these methods may prove useful at the nonresponse treatment stage. In this article, we conduct an empirical investigation in order to compare several machine learning procedures in terms of bias and efficiency. In addition to the classical machine learning procedure, we assess the performance of ensemble approaches that make use of different machine learning procedures to produce a set of weights adjusted for nonresponse.

Introduction

Most surveys conducted by national statistical offices collect information on many survey variables and the aim is to estimate many population parameters : such surveys are often referred to as multipurpose surveys. Response rates have been declining over time. Thus, there is an increased concern for the potential of nonresponse bias. Unit nonresponse, which is characterized by the absence of information for all the survey variables, is usually treated by some form of weight adjustment procedure. The main idea behind a weighting adjustment consists of inflating the weight of the respondents to compensate for the nonrespondents. The inflation factor is defined as the inverse of the estimated response probability to the survey. The treatment of unit nonresponse starts with postulating a nonresponse model describing the relationship between the response indicators (equal to 1 for respondents and 0 for nonrespondents) and a vector of explanatory variables. Determining a suitable model is thus crucial. This modeling exercise consists of two steps : (i) select a vector of explanatory variables that are predictive of the response indicators and that are related to the survey variables ; (ii) Determine a suitable model for the relationship between the response indicator and the selected explanatory variables ; see Haziza and Beaumont (2017).

In recent years, there has been a substantial interest in machine learning methods in national statistical offices. Machine learning procedures provide flexible approaches able to adapt to complex non-linear and non-additive relationships between a response variable and a set of predictors and may prove useful in the context of big data sets. Although these procedures can prove useful in the context of unit nonresponse, one should exercise some caution. Indeed, many machine learning procedures are known to have very good predictive performances. However, in the context of unit nonresponse, one face an estimation problem rather than a prediction problem. Our goal is to estimate a finite population parameter (e.g., a population total) and the most predictive nonresponse model may not necessarily lead to the best estimator (in terms of mean square error) of a population total. This will be illustrated in Section 3. Our problem here is different from what is encountered in the context of imputation for imputing item nonresponse. In that context, the most predictive model is expected to perform well in terms of bias and efficiency.

In this paper, we conduct an extensive simulation study to compare several machine learning procedures in terms of bias and efficiency. Other empirical investigation on the use of machine learning in the context of unit nonresponse is surveys can be found in Lohr et al. (2015), Gelein (2017) and Kern et al. (2019).

1 The setup

Consider a finite population U of size N ; i.e., $\mathscr{U} = \{1, \ldots, k, \ldots, N\}$. In this paper, the aim is to estimate the population total of a survey variable y, $t_y := \sum_{k \in \mathscr{U}} y_k$. To that end, we select a sample S, of size n, according to a sampling design, $P(S \mid \mathbf{Z})$, with first-order inclusion probabilities $\pi_k, k \in U$, where $\mathbf{Z}$ denotes the matrix of design information. In the absence of nonsampling errors, a design-unbiased estimator of t_y is the well known Narain–Horvitz–Thompson estimator

$$\widehat{t}_{y\pi} := \sum_{k \in \mathscr{S}} d_k y_k, \tag{1}$$

where $d_k = 1/\pi_k$ denotes the design weight attached to unit k.

In the presence of unit nonresponse, the survey variable y is collected for a subset $\mathscr{S}_r \subset \mathscr{S}$. Let R_k be a response indicator attached to unit k such that $R_k = 1$ if unit k responds to the survey and $R_k = 0$, otherwise. Let p_k denote the response probability associated with unit k. In our empirical study, we make the following assumptions : (i) The response indicators R_k are mutually independent ; (ii) The response indicators R_K are independent from the sample selection indicators I_k, where $I_k = 1$ if $k \in \mathscr{S}$ and $I_k = 0$, otherwise. This assumption implies that the response probability of a unit y is essentially determined by fixed respondent characteristics. This assumption may be violated in the context of adaptative collection designs (e.g., Groves and Heeringa, 2006). (iii) the positivity assumption is satisfied ; i.e., $\pi_k > 0$ for all k and $p_k > 0$ for all k.

A naive estimator of t_y is given by

$$\widehat{t}_{y,naive} = N\frac{\sum_{k\in\mathcal{S}} d_k R_k y_k}{\sum_{k\in\mathcal{S}} d_k R_k}. \tag{2}$$

Alternatively, the population size N in (2) may be replaced by the estimated population size $\widehat{N}_\pi = \sum_{k\in\mathcal{S}} d_k$. Unless the data are Missing Completely At Random (MCAR), the estimator $\widehat{t}_{y,naive}$ is biased. The bias may be significant if the nonresponse rate is high and/or the responding units and the nonresponding units exhibit a different behavior with respect ot the survey variable y.

If the response probabilities p_k were known, a design-unbiased estimator of t_y is the so-called double expansion estimator (Sarndal et al., 1992) :

$$\hat{t}_{y,DE} := \sum_{k\in\mathcal{S}} d_k R_k \frac{y_k}{p_k}. \tag{3}$$

In practice, the p_k's are unknown and are replaced by estimated response probabilities $\widehat{p}_k$. It is common practice to postulate a nonresponse model, which is a set of assumptions about the unknown nonresponse mechanism. More specifically, we postulate the following model :

$$\mathbb{E}(R_k \mid y_k, \mathbf{x}_k) = m(\mathbf{x}_k), \tag{4}$$

where $m(\cdot)$ is either a predetermined function in the case of a parametric model or is left unspecified in the case of a nonparametric model, and $\mathbf{x}_k$ is a vector of fully observed variables (i.e., available for both the responding and the nonresponding units). The resulting estimator, often referred to as the propensity score adjusted estimator, is given by

$$\widehat{t}_{y,PSA} := \sum_{k\in\mathcal{S}} d_k R_k \frac{y_k}{\widehat{p}_k}. \tag{5}$$

An alternative estimator of t_y is the Hajek estimator

$$\widehat{t}_{y,H} := \frac{N}{\widehat{N}} \sum_{k\in\mathcal{S}} d_k R_k \frac{y_k}{\widehat{p}_k}. \tag{6}$$

Although both $\widehat{t}_{y,PSA}$ and $\widehat{t}_{y,H}$ exhibit the same asymptotic bias, they may differ significantly from one another in terms of variance. If the nonresponse model (4) is correctly specified, both $\widehat{t}_{y,PSA}$ and $\widehat{t}_{y,H}$ will be nearly unbiased. The weights adjusted for nonresponse are defined as $w_k^* = d_k/\widehat{p}_k$.

2 Estimation vs. prediction

In this section, we illustrate empirically that the best predictive model does not necessarily yield the best estimator of t_y in terms of mean square error. Indeed, including predictors that are highly predictive of R_k may lead to very small estimated response probabilities $\widehat{p}_k$, which may result in extreme adjusted weights w_k^*. In this case, both (5) and (6) may be inefficient. How then to choose the $\mathbf{x}_k$ variables to incorporate in the model ? A common recommendation is to include the variables $\mathbf{x}_k$ that are related to both the indicator variable R_k and the variable of interest y ; e.g., Little and Vartivarian (2005), Beaumont (2005). Indeed, if an x-variable is strongly related to R_k but not to the survey variable y, it is not desirable to include it in the nonresponse model, since it will not help reduce the nonresponse bias but may contribute to increasing the variance of the point estimator.

As an illustration, we generated a finite population of size $N = 10,000$ with seven variables : one survey variable y and six auxiliary variables x_1 to x_6. We first generated the x-variables according to the following distributions : $x_1 \sim \text{Gamma}(5,1)$; $x_2 \sim \text{Gamma}(1,5)$; $x_3 \sim \text{Gamma}(1,6)$; $x_4 \sim \text{Gamma}(1,10)$; $x_5 \sim \text{Gamma}(1,20)$; $x_6 \sim \text{Gamma}(0.5,50)$. Given x_1-x_6, we generated a y-variable according to the linear model

$$y_k = 2 - 2x_{1k} + 4x_{2k} + \epsilon_k,$$

where the errors ϵ_k were generated from a normal distribution of mean equal to zero and variance equal to 1.

From the population, we drew $10,000$ samples, of size $n = 1,000$, according to simple random design without replacement. In each sample, each unit was assigned a response probability p_k using the logistic function :

$$p_k = 0.05 + 0.95\left\{1 + \exp\left(-0.05x_{1k} + 0.05x_{2k} - 0.05x_{3k} + 0.05x_{4k} - 0.05x_{5k} + 0.02x_{6k}\right)\right\}^{-1}.$$

This led to a response rate of about 50% in each sample. In each sample, the indicator variables R_k were generated according to a Bernoulli distribution with probability p_k. Our goal is to estimate the population total, $t_y = \sum_{k\in U} y_k$. In our experiment, the variables x_1-x_6 are fully observed and only the y-variable is prone to missing values.

In each sample, we computed two estimators of t_y :

(i) The naive estimator given by (2).

(ii) The propensity score-adjusted estimator, $\widehat{t}_{y,PSA}$ given by (5), where $\widehat{p}_k$ was obtained using the score method (described below) based on different subsets of the variables x_1-x_6.

The score method(Little, 1986, Eltinge and Yansaneh, 1997 ; Haziza and Beaumont, 2007) may described as follows :

— *Step 1* : Obtain preliminary estimated response probabilities, $\widehat{p}_k^{LR}$, $k \in S$, from a logistic regression.

— *Step 2* : Form 20 classes based on the estimated response probabilities, $\widehat{p}_k^{LR}$, using either an equal quantile method.

— *Step 3* : Perform weight adjustment within each class (i.e, divide the design weight d_k of the kth respondents in a given class by the response rate observed within the same class).

We computed the Monte Carlo percent relative bias of each estimator

$$RB_{MC}(\widehat{t}) = \frac{1}{10,000}\sum_{b=1}^{10,000}\frac{(\widehat{t}_{(b)} - t_y)}{t_y} \times 100,$$

as well as the Monte Carlo mean square error

$$MSE_{MC}(\widehat{t}) = \frac{1}{10,000}\sum_{b=1}^{10,000}\left(\widehat{t}_{(b)} - t_y\right)^2,$$

where $\widehat{t}_{(b)}$ denotes the $\widehat{t}$ estimator in the b-th sample, $b = 1, \ldots, 10000$. To ease readability, we computed the relative efficiency of the point estimators, defined as

$$RE_{MC}(\widehat{t}) = 100 \times \frac{MSE_{MC}(\widehat{t})}{MSE_{MC}(\widehat{t}_{y,\pi})}$$

where $\widehat{t}_{y,\pi}$ is the complete data estimator given by (1). In addition, in each sample, we computed the Monte Carlo percent coefficient of variation of the adjusted weights w_k^* defined as

$$CV_{MC} = \frac{100}{B}\sum_{b=1}^{B}\frac{s_{w^*(b)}}{\overline{w}^*_{(b)}},$$

where

$$s_{w^*} = \frac{1}{n_r - 1} \sum_{k \in S_r} (w_k^* - \overline{w}^*)^2$$

with $\overline{w}^* = n_r^{-1} \sum_{k \in S_r} w_k^*$.

Finally, we computed the Monte Carlo mean square error of the predictions defined as

$$MSE = \frac{100}{B} \sum_{b=1}^{B} \frac{1}{n_r} \sum_{k \in S_r} \left(\widehat{p}_k^b - p_k \right)^2 .$$

The results are displayed in Table 1.

Estimator	$\widehat{t}_{y,naive}$	$\widehat{t}_{y,PSA}$ x_1	$\widehat{t}_{y,PSA}$ x_1-x_2	$\widehat{t}_{y,PSA}$ x_1-x_3	$\widehat{t}_{y,PSA}$ x_1-x_4	$\widehat{t}_{y,PSA}$ x_1-x_5	$\widehat{t}_{y,PSA}$ x_1-x_6
RB_{MC} in (%)	-14.1	-13.0	-1.7	-1.8	-0.7	-1.1	-0.8
RE_{MC}	540	480	112	118	117	149	218
$CV(w*)$ in (%)	0	17.4	19.6	21.7	30.1	46.7	64.2
MSE	4.8	5.4	5.3	5.1	4.6	1.7	0.9

TABLE 1 – Monte percent relative bias and mean square error of several estimator of t_y

The results in the table 1 can be summarized as follows :

— As expected, the naive estimator was biased with a relative bias of -14.1%. This result is not surprising because the naive estimator does not take into account the variables x_1 and x_2 which are related to both R_k and y.

— The propensity score estimator $\widehat{t}_{y,PSA}$ based on the variable x_1 exhibited a smaller bias than the unadjusted estimator, which can be explained by the fact that it incorporates the variable x_1 which is related to both the probability of response and the variable of interest y. The bias is explained by the fact that the variable x_2 was not included.

— The propensity score estimator $\widehat{t}_{y,PSA}$ based on the variable x_1 and x_2 was nearly unbiased bias because it included both x_1 and x_2 in the nonresponse model. In terms of relative efficiency, this estimator was the best with a value of RE equal 112. It is worth noting that the other propensity score estimators were nearly unbiased but were less efficient than $\widehat{t}_{y,PSA}$ based on x_1 and x_2. In other words, incorporating x_3 to x_6 into the model contributed in increasing the variance.

— The most predictive model of R_k included all the x-variables x_1-x_6. However, except for $\widehat{t}_{y,PSA}$, based on x_1, the propensity score estimators based on x_1-x_6 was the worst in terms of relative efficiency with a value of RE equal to 218. In comparison with $\widehat{t}_{y,PSA}$, based on x_1 and x_2, this corresponds to a significant increase of 194%. This result shows that the most predictive model does not necessarily translate into the best estimator of the total t_y. This is supported by the values of the mean square of the predictions : 5.4 for $\widehat{t}_{y,PSA}$ based on x_1 and x_2 and only 0.9 for for $\widehat{t}_{y,PSA}$ based on x_1-x_6.

— A large dispersion of the adjusted weights w_k^* led to estimators with a large variance. This is why, in practice, it is desirable to limit the dispersion of weights.

3 Ensemble methods

In addition to commonly encountered machine learning procedures (see Section 4.?), we tested the performance of three ensemble methods. The rationale behind an ensemble method is to obtain estimated response probability using several (machine learning or non machine learning) procedures and combining these probabilities in some way to obtain a set of weights adjusted for nonresponse. Why use an ensemble method ? As we illustrate in Section 4, there is no machine learning procedures that outperforms all the other competitors in all the scenarios. Indeed, a machine learning procedures may do well in a particular scenario but not as well in another scenario. However, one cannot tell in advance which procedure will perform well. An ensemble method that combines several machine learning procedures, may outperform a single procedure, which is an attractive feature.

Below, we describe three methods for combining the machine learning procedures : the first is based on a calibration procedure similar to a model calibration procedure (Wu and Sitter, 2001) ; the second is based on refitting (Duan and Yin, 2017, Chen and Haziza, 2019) ; The third uses both refitting and calibration.

Let $\widehat{\mathbf{p}}_k = (\widehat{p}_k^{(1)}, \ldots, \widehat{p}_k^{(M)})$ be a M-vector of estimated response probabilities associated with unit k. The component $\widehat{p}_k^{(m)}$ in $\widehat{\mathbf{p}}_k$ corresponds to an estimated response probability based on the mth machine learning procedure, $m = 1, \ldots, M$.

The three ensemble methods are described below.

(1) *Calibration* Combining through calibration proceeds as follows : we seek calibrated weight w_k such that

$$\sum_{k \in S_r} \frac{G(w_k, d_k)}{q_k} \tag{7}$$

subject to

$$\sum_{k \in S_r} w_k = \sum_{k \in S} d_k$$

and

$$\sum_{k \in S_r} w_k \mathscr{L}(\widehat{\mathbf{p}}_k) = \sum_{k \in S} d_k \mathscr{L}(\widehat{\mathbf{p}}_k),$$

where $\mathscr{L}(\cdot)$ is the inverse of the calibration function $F(\cdot)$. The resulting weights w_k may be viewed as a scalar summary of the information contained in the M-vector $\widehat{\mathbf{p}}_k$. The resulting estimator of t_y is given by

$$\widehat{t}_y^{\text{cal}} = \sum_{k \in \mathcal{S}} w_k R_k y_k.$$

(2) *Refitting* Refitting consists of compressing the information contained in $\widehat{\mathbf{p}}_k$ by fitting a linear regression model with the response indicator R_k as the dependent variable and $\widehat{\mathbf{p}}_k$ as the vector of explanatory variables :

$$R_k = \sum_{m=1}^{M} \beta^{(m)} \widehat{p}_k^{(m)} + \varepsilon_k. \tag{8}$$

.

Let $\widehat{\boldsymbol{\beta}} = (\widehat{\beta}^{(1)}, \ldots, \widehat{\beta}^{(M)})^\top$ be the least squares estimator of $\boldsymbol{\beta} = (\beta^{(1)}, \ldots, \beta^{(M)})^\top$. We define the $\tilde{\boldsymbol{\beta}} := (\tilde{\beta}^1, \ldots, \tilde{\beta}^M) = \frac{1}{<\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\beta}}>}((\hat{\beta}^1)^2, \ldots, (\hat{\beta}^m)^2)$. where $< .,. >$ denotes the customary dot product in $\mathbb{R}^M$. Note that this standardization ensures that $\tilde{\boldsymbol{\beta}} \in [0;1]^M$ and $\sum_{j=1}^{M} \tilde{\boldsymbol{\beta}}^j = 1$. The vector of compressed scores $\widehat{p}_k^{\text{com}}$ is defined as $p_k^{\text{com}} =< \widetilde{\boldsymbol{\beta}}, \widehat{\mathbf{p}}_k >$. Because of the standardization, the components, the compressed score $\widehat{p}_k^{\text{com}}$ lies between 0 and 1. An estimator of t_y through refitting is given by

$$\widehat{t}_y^{\text{com}} = \sum_{k \in \mathcal{S}} d_k R_k \frac{y_k}{\widehat{p}_k^{\text{com}}}.$$

(3) *Refitting followed by calibration* We start by obtaining the compressed scores $\widehat{p}_k^{\text{com}}$ as above. Then, we seek calibrated weights w_k such that

$$\sum_{k\in S_r} \frac{G(w_k, d_k)}{q_k} \tag{9}$$

subject to

$$\sum_{k\in S_r} w_k = \sum_{k\in S} d_k$$

and

$$\sum_{k\in S_r} w_k \mathscr{L}(\widehat{p}_k^{\text{comp}}) = \sum_{k\in S} d_k \mathscr{L}(\widehat{p}_k^{\text{comp}}).$$

Unlike in (1) where there are $M+1$ calibration constraints, we only have two calibration constraints when calibration is performed after refitting. The resulting estimator of t_y is given by

$$\widehat{t}_y^{\text{com-cal}} = \sum_{k\in\mathcal{S}} w_k R_k y_k.$$

4 Simulation study

We conducted an extensive simulation study to assess the performance of several machine learning procedures (see Section 4.2) in terms of bias and efficiency.

4.1 Setup

We generated several finite populations of size $N = 50,000$. Each population consisted of a survey variable Y and 6 auxiliary variables drawn independently, three of which were continuous and the remaining being discrete. First, the continuous auxiliary variables were generated as follows : $X^{(s)} \sim \text{Gamma}(3,2)$, $X^{(c_1)} \sim \mathscr{N}(0,1)$; $X^{(c_2)} \sim \text{Gamma}(3,2)$ and $X^{(c_3)} \sim \text{Gamma}(3,2)$. The discrete auxiliary variables were generated as follows : $X^{(d_1)} \sim \mathscr{MN}(N, 0.5, 0.05, 0.05, 0.1, 0.3)$; $X^{(d_2)} \sim \text{Ber}(0.5)$ and $X^{(d_3)} \sim \text{UD}(1;5)$. We used two configurations for these predictors : (i) They were independently generated; (ii) Correlation between them was introduced through Gaussian copulas.

Given the values of the auxiliary variables, we have generated several y-variables according to the following models :

$$\begin{aligned} y_k = \gamma_0 + \gamma_1^{(s)} X_{1k}^{(s)} + \gamma_1^{(c)} X_{1k}^{(c)} + \gamma_2^{(c)} X_{2k}^{(c)} + \gamma_3^{(c)} X_{3k}^{(c)} + \sum_{j=2}^{5} \gamma_{1j}^{(d)} (1_{\{X_{1k}^{(d)}=j\}}) \\ + \gamma_2^{(d)} X_{2k}^{(d)} + \sum_{k=2}^{5} \gamma_{3j}^{(d)} (1_{\{X_{3k}^{(d)}=j\}}) + \varepsilon_k \end{aligned} \tag{10}$$

and

$$y_k = \delta_1 X_{2k}^{(c)} + \delta_2 (X_{2k}^{(c)})^2 (1 - 1_{\{X_{3k}^{(d)}=2\}\cup\{X_{3k}^{(d)}=3\}}) + \log(1+\delta_3 X_{2k}^{(c)})(1_{\{X_{3k}^{(d)}=2\}\cup\{X_{3k}^{(d)}=3\}}) + \varepsilon_k, \tag{11}$$

where $\varepsilon \sim \mathscr{N}(0, \sigma_\varepsilon^2)$. Note that the model (10) is linear in the coefficients, whereas Model (11) corresponds to a nonlinear relationship between the response variable and the predictors. The values of the model parameters are displayed in Table [**?**]. For the linear model we used both an non-informative sampling design and an informative sampling design with a correlation between the y-variable and the design weights d_k equal to ? ? ?. For the non-informative sampling design, the vector of coefficients $\left(\beta_0, \beta^{(s)}, \beta_1^{(c)}, \beta_2^{(c)}, \beta_3^{(c)}, \beta_{12}^{(d)}, \beta_{13}^{(d)}, \beta_{14}^{(d)}, \beta_{15}^{(d)}, \beta_{22}^{(d)}, \beta_{32}^{(d)}, \beta_{33}^{(d)}, \beta_{34}^{(d)}, \beta_{35}^{(d)}\right)$ was set to $(-0.2, 5.0, 5.0, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5)$. For the informative sampling design, this vector was set to $(-10, 5.0, 5.0, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5)$. Finally, for the non-linear model, the vector of coefficients $(\delta_0, \delta_1, \delta_2, \delta_3)$ was set to $(4, 4, 4, 4)$. This led to six different finite populations.

Each population was partitioned into ten strata on the basis of the auxiliary variable $X^{(s)}$ using an equal quantile method. From each population, we selected $B = 1,000$ samples according to stratified simple random sampling without replacement of size $n = 1,000$ based on Neyman's allocation.

In each sample, nonresponse the survey variable Y was generated according to six nonresponse mechanisms. For each $k \in \mathcal{S}$, we assigned a response probability p_k according to the following six functions :

1. $p_k^{(1)} = \text{logit}^{-1}(-0.8 - 0.05X_{1k}^{(s)} + 0.2X_{1k}^{(c)} + 0.5X_{2k}^{(c)} - 0.05X_{3k}^{(c)} + \sum_{k=2}^{5} 0.2(1_{\{X_{1k}^{(c)}=k\}}) + 0.2X_{2k}^{(d)} + \sum_{k=2}^{5} 0.3(1_{\{X_{3k}^{(d)}=k\}}))$.
2. $p_k^{(2)} = 0.1 + 0.9\,\text{logit}^{-1}(0.5 + 0.3X_{1k}^{(s)} - 1.1X_{1k}^{(c)} - 1.1X_{2k}^{(c)} - 1.1X_{3k}^{(c)} + \sum_{k=2}^{5} 0.8(1_{\{X_{1k}^{(c)}=k\}}) + 0.8X_{2k}^{(d)} + \sum_{k=2}^{5} 0.8(1_{\{X_{3k}^{(d)}=k\}}))$.
3. $p_k^{(3)} = 0.1 + 0.9\,\text{logit}^{-1}\left(-1 + \text{sgn}\left(X_{1k}^{c}\right)\left(X_{1k}^{c}\right)^2 + 3 \times 1_{\left\{X_{1k}^{(d)}<4\right\}\cap\left\{X_{2k}^{(d)}=1\right\}}\right)$.
4. $p_k^{(4)} = 0.55 + 0.45\tanh(0.05y_k - 0.5)$.
5. $p_k^{(5)} = 0.1 + 0.9\,\text{logit}^{-1}(0.2y_k - 1.2)$.
6. $p_k^{(6)} = 0.1 + 0.6\,\text{logit}^{-1}(0.85X_{1k}^{(s)} + 0.85X_{2k}^{(c)} - 0.85X_{3k}^{(c)} - \sum_{k=2}^{5} 0.2(1_{\{X_{1k}^{(c)}=k\}}) + 0.2X_{2k}^{(d)} - \sum_{k=2}^{5} 0.3(1_{\{X_{3k}^{(d)}=k\}}))$.

The parameters in each nonresponse model were set so as to obtain a response rate approximately equal to 50%. The response indicators $R_k^{(j)}$ were generated from a Bernoulli distribution with probability $p_k^{(j)}$, $j = 1, \ldots, 6$.. Note that the nonresponse mechanism (1)-(3) and (6) are ignorable, whereas the nonresponse mechanism (4) and (5) are nonignorable.

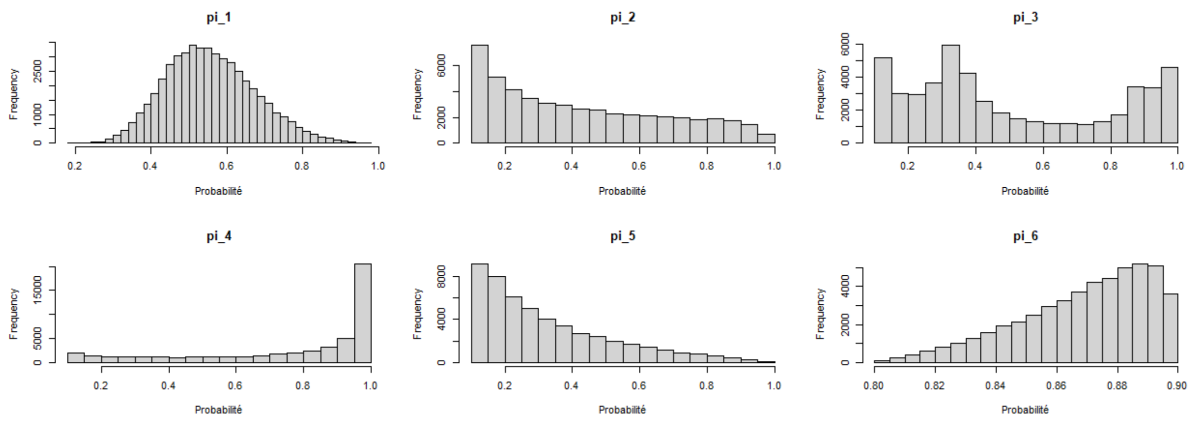

FIGURE 1 – Distribution of response probabilities in the population $\mathcal{U}$

Since we used six survey variables and six nonresponse mechanisms, we ended up with 36 scenarios, each scenario corresponding to a given survey variable and a given nonresponse mechanism.

To estimate the response probabilities p_k, we used the following procedures using $X^{(s)}$, $X_1^{(c)}$, $X_2^{(c)}$, $X_2^{(c)}$, $X_1^{(d)}$, $X_2^{(d)}$ and $X_3^{(d)}$ as the set of explanatory variables :

(a) Logistic regression ;
— `logit`.

(b) Logistic regression with variable selection based on LASSO ; e.g., see [Hastie et al., 2001].
— `logit_lasso` : the amont of penalization λ is obtained using a 10-fold cross validation.

(c) Classification and regression trees ; e.g, see [Breiman et al., 1983].
— `cart1` : Pruned trees, at least 10 observations in each leaf.
— `cart2` : Pruned trees, at least 20 observations in each leaf.
— `cart3` : Pruned trees, at least 30 observations in each leaf.
— `cart4` : Unpruned trees, at least 20 observations in each leaf.

(d) Random forests ; e.g., see [Breiman, 2004].
— `rf1` : Probabilities estimation trees, at least 10 observations in each leaf, 100 trees.
— `rf2` : Probabilities estimation trees, at least 10 observations in each leaf, 500 trees.
— `rf3` : Probabilities estimation trees, at least 30 observations in each leaf, 100 trees.
— `rf4` : Probabilities estimation trees, at least 30 observations in each leaf, 500 trees.
— `rf5` : Probabilities estimation trees, at least 30 observations in each leaf, 500 trees, variable used for the allocation is always drawn.

(e) k-nearest neighbors ;
— knn : k determined by 10-fold cross validation with $k \in \{3, 12\}$.
— knn_reg : k determined by 10-fold cross validation with $k \in \{3, 30\}$.

(f) Bayesian additive regression tree ; e.g., [Chipman et al., 2010].
— : `bart` Bart as a classification method with parameters described in the original paper for all priors.
— : `bart_reg` : Bart as a regression method with parameters described in the original paper for all priors.

(g) Extreme Gradient Boosting (XGBoost) ; see [Chen and Guestrin, 2016].
— `xb1` : 500 trees, $\Gamma = 10$, proportion for subsets : 75 %, learning rate : 0.5, max depth : 2.
— `xgb2` : 2000 trees, $\Gamma = 2$, proportion for subsets : 100 %, learning rate : 0.5, max depth : 2.
— `xgb3` : 1000 trees, $\Gamma = 2$, proportion for subsets : 75 %, learning rate : 0.01, max depth : 1.
— `xgb4` : 500 trees, $\Gamma = 10$, proportion for subsets : 75 %, learning rate : 0.05, max depth : 3.

(h) Support vector machine ;
— `svm1` : $\nu-$SVM with a Gaussian kernel, $\nu = 0.7$, $\gamma = 0.025$.
— `svm2` : $\nu-$SVM with a linear kernel, $\nu = 0.7$.

(i) Cubist algorithm ; [Quinlan, 1992] [Quinlan, 1993].
— `cb1` : Unbiaised, 100 rules, with extrapolation, 10 commitees.
— `cb2` : Unbiaised, 100 rules, without extrapolation, 10 commitees.
— `cb3` : Biaised, 100 rules, with extrapolation, 10 commitees.
— `cb4` : Unbiaised, 100 rules, with extrapolation, 50 commitees.
— `cb5` : Unbiaised, 100 rules, with extrapolation, 100 commitees.

(j) Model-based recursive partitioning ; [Zeileis et al., 2008].
— `mob` : logit model fitted, $X^{(s)}$ for stratification.

(k) Ensemble method based on calibration ; see Section 3 ;

(l) Ensemble method based on refitting ; see Section 3 ;

(m) Ensemble method based on calibration followed by refitting ; see Section 3.

In each sample, we computed two estimators : (i) the propensity score adjusted estimator, $\widehat{t}_{y,PSA}$ given by (5) and (ii) The Hajek estimator, $\widehat{t}_{y,H}$ given by (6).

As a measure of bias of an estimator $\widehat{t_y}$, we computed its Monte Carlo percent relative bias :

$$\mathbb{B}_{MC}(\widehat{t_y}) = \frac{100}{B}\sum_{k=1}^{B}\frac{\left(\widehat{t}_{y,k}-t_y\right)}{t_y}. \tag{12}$$

We also computed the Monte Carlo relative efficiency, using the complete data estimator $\widehat{t}_{y,\pi}$:

$$\mathrm{RE}_{MC}(\widehat{t_y}) = 100 \times \frac{\mathrm{MSE}_{MC}(\widehat{t_y})}{\mathrm{MSE}_{MC}(\widehat{t}_{y,\pi})}, \tag{13}$$

where

$$\mathrm{MSE}_{MC}(\hat{t_y}) = \frac{1}{B}\sum_{k=1}^{B}\left(\widehat{t}_{y,k}-t_y\right)^2 \tag{14}$$

and $\mathrm{MSE}_{MC}(\widehat{t}_{y,\pi})$ is defined similarly.

4.2 Results

4.2.1 Comparison of machine learning methods

In this section, the efficiency of the PSA estimator and the Hàjek estimator will be studied. For each algorithm of machine learning, an estimate of the efficiency of the estimators is available for 42 configurations (6 non-response mechanisms and 7 general scenarios).

Algorithm	Min	Q1	Med	Q3	Max	Mean
xgb1	155	225	324	1 124	12 551	1 677
COMPRESS_CAL	139	208	328	798	7 772	908
xgb4	148	221	330	1 139	12 111	1 589
xgb3	143	239	344	928	11 581	1 394
cart3	175	259	345	1 506	9 627	1 393
cart2	175	256	348	1 464	9 472	1 376
COMPRESS	137	199	348	906	10 382	1 317
CART_reg	162	269	350	1 367	9 522	1 293
cart1	172	259	351	1 448	9 373	1 370
xgb2	148	215	368	1 016	11 479	1 405
cart4	145	262	369	1 382	8 881	1 231
bart	129	199	384	852	10 595	1 314
knn	172	282	392	921	11 513	1 621
logit and score	134	216	392	1 252	9 998	1 359
svm1	129	280	407	780	12 482	1 639
knn_reg	144	261	413	1 020	12 398	1 745
rf4	188	235	417	1 133	9 341	1 413
cb4	197	263	456	1 592	16 376	1 948
cb5	199	267	466	2 406	17 395	2 249
calibration	222	318	472	875	7 475	1 031
rf2	199	278	487	1 470	9 717	1 482
rf5	200	269	508	2 847	25 181	2 408
rf3	192	264	522	1 419	9 215	1 488
cb1	194	270	524	1 814	14 125	2 002
bart_reg	143	208	571	2 479	*	*
cb2	181	241	598	3 239	23 578	3 385
logit_lasso	141	331	636	1 739	15 895	2 520
rf_reg	225	343	821	1 989	19 596	2 203
logit	123	215	962	5 786	*	84 503
rf1	228	345	1 147	2 152	10 973	2 208
mob	121	833	10 846	106 423	*	*
cb3	304	53 745	890 538	*	*	*
svm2	297	32 212	*	*	*	*

Algorithm	Min	Q1	Med	Q3	Max	Mean
xgb1	171	220	295	1 751	12 305	1 864
COMPRESS	158	196	296	1 470	10 144	1 443
xgb4	170	219	296	1 741	11 783	1 778
bart	159	202	306	1 417	10 201	1 457
xgb3	147	201	307	1 508	10 815	1 560
logit and score	135	217	308	1 267	9 984	1 377
xgb2	148	206	315	1 520	10 817	1 567
COMPRESS_CAL	139	208	328	798	7 772	908
CART_reg	163	252	344	1 733	9 515	1 382
cb4	165	224	345	1 389	12 223	1 675
cb5	163	224	346	1 398	12 255	1 680
cart4	145	229	362	1 413	8 879	1 255
cb1	182	228	363	1 365	12 281	1 680
cb2	138	211	419	1 291	10 922	1 367
cart1	173	248	421	1 807	9 369	1 485
cart2	174	240	422	1 807	9 472	1 487
cart3	174	243	430	1 844	9 627	1 510
rf4	156	195	437	1 555	9 721	1 406
knn	198	253	449	2 219	10 875	1 826
calibration	222	318	472	875	7 475	1 031
rf2	156	202	477	1 512	9 397	1 348
knn_reg	187	251	485	2 352	11 932	1 998
rf3	159	198	489	1 529	9 607	1 401
rf5	153	202	493	1 275	9 890	1 327
svm1	187	279	516	2 691	12 231	2 069
rf_reg	151	212	547	1 458	9 159	1 345
rf1	150	219	572	1 598	9 149	1 382
logit	123	218	671	2 202	27 493	2 770
logit_lasso	193	305	679	2 759	15 670	2 833
bart_reg	173	217	1 401	5 545	*	*
svm2	237	452	1 491	3 723	23 959	3 966
cb3	223	1 605	3 246	8 241	60 590	7 404
mob	122	976	8 259	374 131	*	*

TABLE 3 – Relative Monte-Carlo efficiency for PSA estimator (left) and Hàjek estimator (right).

Algorithm	Min	Q1	Med	Q3	Max	Mean
xgb1	15	9	1	10	21	19
COMPRESS_CAL	7	4	2	2	2	1
xgb4	13	8	3	12	18	16
xgb3	9	11	4	7	17	11
cart3	19	14	5	20	9	10
cart2	20	13	6	18	7	9
COMPRESS	6	1	7	5	12	6
CART_reg	16	21	8	14	8	4
cart1	18	15	9	17	6	8
xgb2	14	6	10	8	15	12
cart4	12	17	11	15	3	3
bart	3	2	12	3	13	5
knn	17	26	13	6	16	17
logit and score	5	7	14	13	11	7
svm1	4	25	15	1	20	18
knn_reg	11	16	16	9	19	20
rf4	22	10	17	11	5	13
cb4	25	18	18	21	24	21
cb5	26	20	19	26	25	25
calibration	29	27	20	4	1	2
rf2	27	24	21	19	10	14
rf5	28	22	22	28	28	26
rf3	23	19	23	16	4	15
cb1	24	23	24	23	22	22
bart_reg	10	3	25	27	33	33
cb2	21	12	26	29	27	28
logit_lasso	8	28	27	22	23	27
rf_reg	30	29	28	24	26	23
logit	2	5	29	30	33	29
rf1	31	30	30	25	14	24
mob	1	31	31	31	33	33
cb3	33	33	32	33	33	33
svm2	32	32	33	33	33	33

Algorithm	Min	Q1	Med	Q3	Max	Mean
xgb1	21	17	1	21	27	25
COMPRESS	14	2	2	12	15	13
xgb4	20	15	3	20	21	23
bart	16	5	4	10	16	14
xgb3	7	4	5	13	17	18
logit and score	3	12	6	3	14	8
xgb2	8	8	7	15	18	19
COMPRESS_CAL	5	9	8	1	2	1
CART_reg	17	26	9	19	9	9
cb4	19	18	10	7	23	20
cb5	18	19	11	8	25	21
cart4	6	21	12	9	3	3
cb1	26	20	13	6	26	22
cb2	4	10	14	5	20	7
cart1	23	24	15	22	6	15
cart2	25	22	16	23	8	16
cart3	24	23	17	24	11	17
rf4	13	1	18	17	12	12
knn	30	27	19	26	19	24
calibration	31	30	20	2	1	2
rf2	12	6	21	14	7	6
knn_reg	28	25	22	27	22	26
rf3	15	3	23	16	10	11
rf5	11	7	24	4	13	4
svm1	27	28	25	28	24	27
rf_reg	10	11	26	11	5	5
rf1	9	16	27	18	4	10
logit	2	14	28	25	30	28
logit_lasso	29	29	29	29	28	29
bart_reg	22	13	30	31	33	33
svm2	33	31	31	30	29	30
cb3	32	33	32	32	31	31
mob	1	32	33	33	33	33

TABLE 2 – Rank efficiency of PSA estimator (left) and Hàjek estimator (right).
The case **(cb3, Min) = 32** means that cb3 has the 32^{th} better minimum efficiency. The minimum efficiency for an algorithm is the minimum MSE we get among all the simulations done with this algorithm. We define in the same way the first quartile efficiency (Q1), the median efficiency (Med), the third quartile efficiency (Q3), the max efficiency (Max) and the mean efficiency (Mean).

The case **(xgb4,Med) = 397** means that the median of all the relative Monte-Carlo efficiencies (as defined in equation 13) using xgb4 as algorithm.

4.3 Comparaison of bias and efficiency for a specific scenario

In this section, the results of the simulations for one particular scenario are examined. The scenario corresponds to a linear y variable with dependent x variables and an informative design.

For each of the six nonresponse mechanisms, each method is represented with a point : on the x-axis, the absolute value of the relative bias is described and on the y-axis, the relative efficiency.

The red triangle corresponds to the calibration method, the red circle to the COMPRESS method and the blue square to the COMPRESS + calibration method.

Each method is described using two points : a blue point based on the PSA estimator without the score method and a red point using the PSA estimator with the 10-class score method.

By seeing this graph, it comes that the Xgboost methods allow to obtain more efficient results in several cases. However, there are strong biases in some situations.

The last two graphs correspond to MAR mechanisms : the calibration method allows to obtain an efficiency similar to the other methods but with a lower bias.

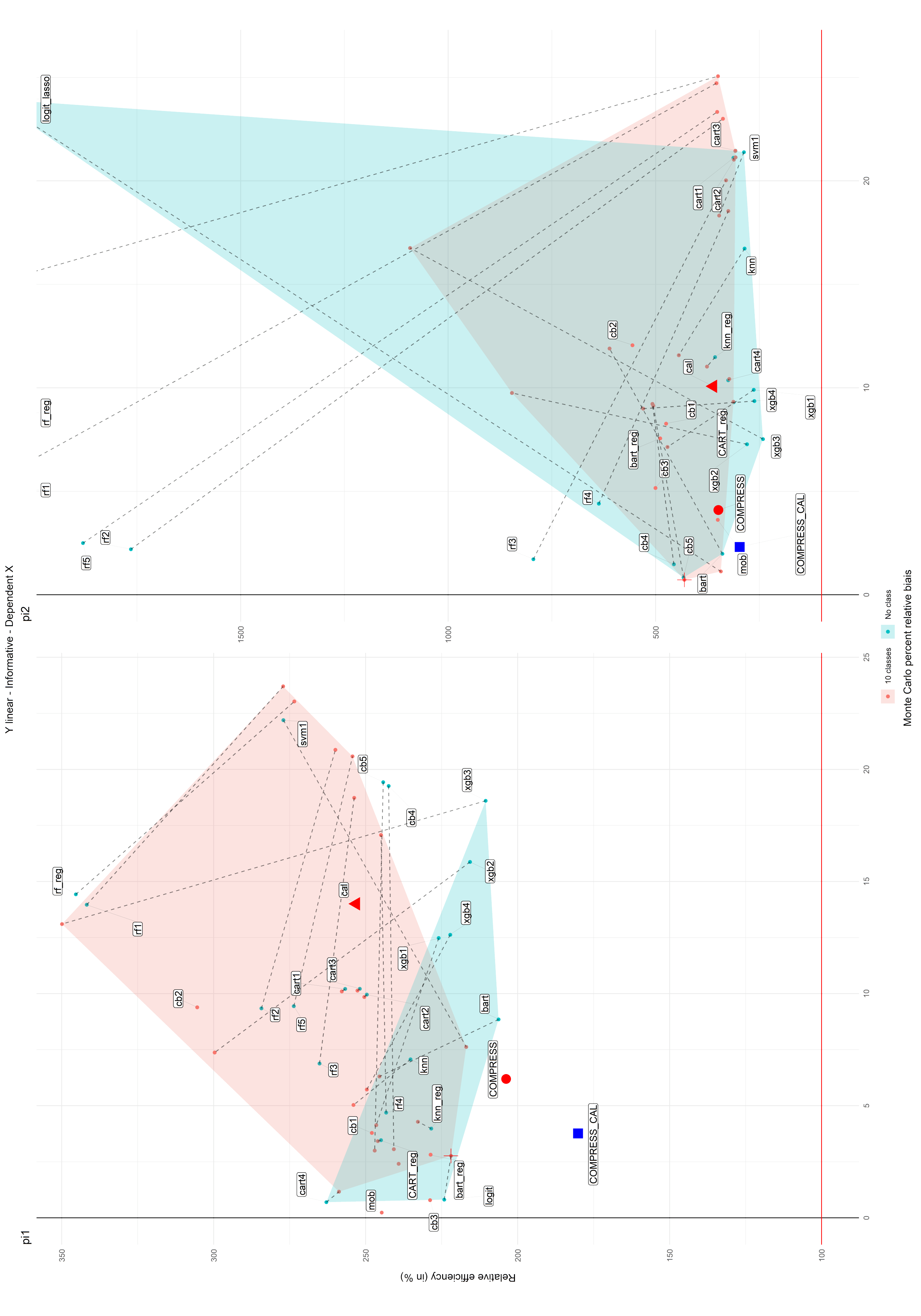

FIGURE 2 – Results for p1 and p2 when y is linear, X dependent and the design is informative.

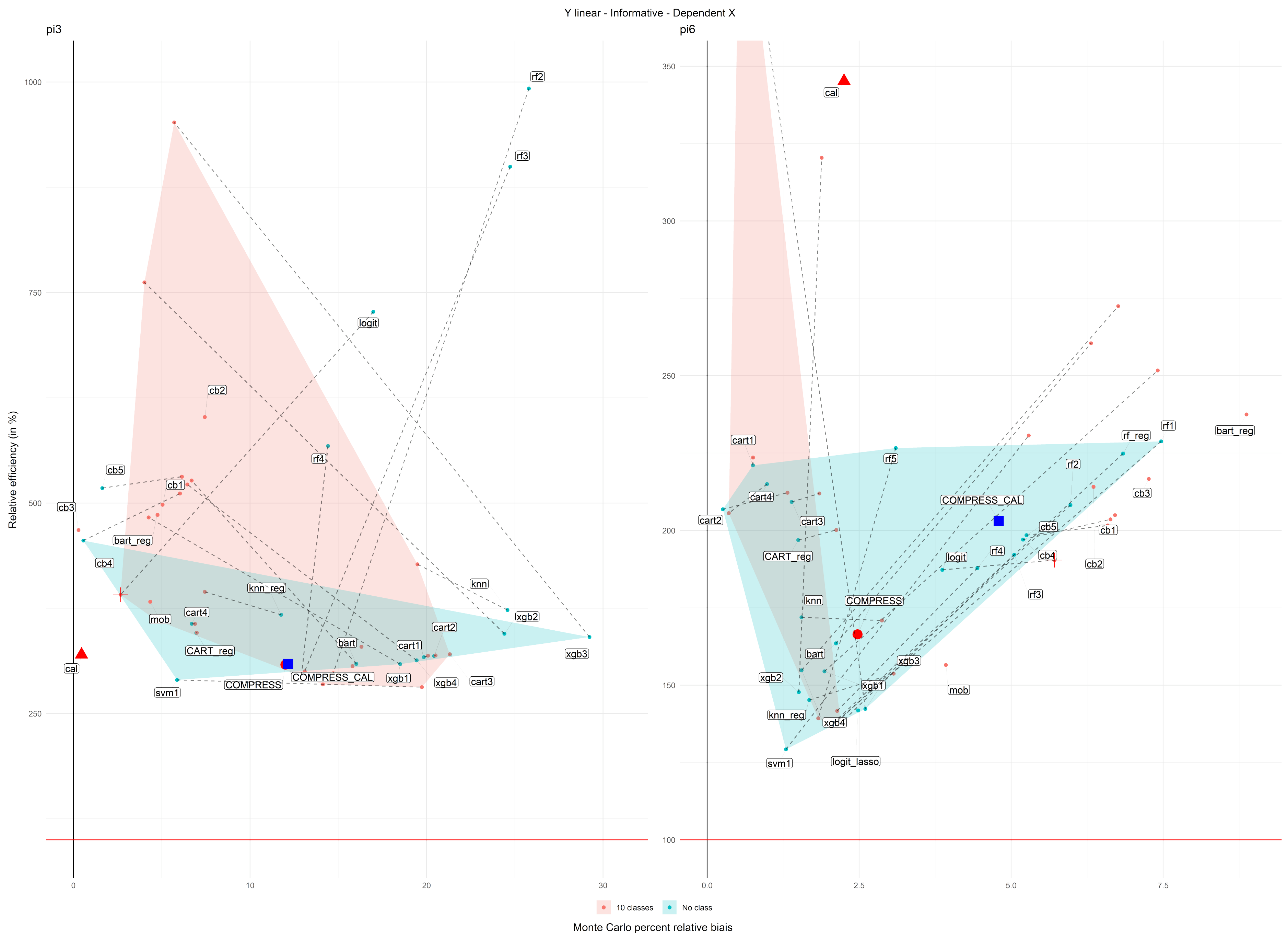

FIGURE 3 – Results for p3 and p6 when y is linear, X dependent and the design is informative.

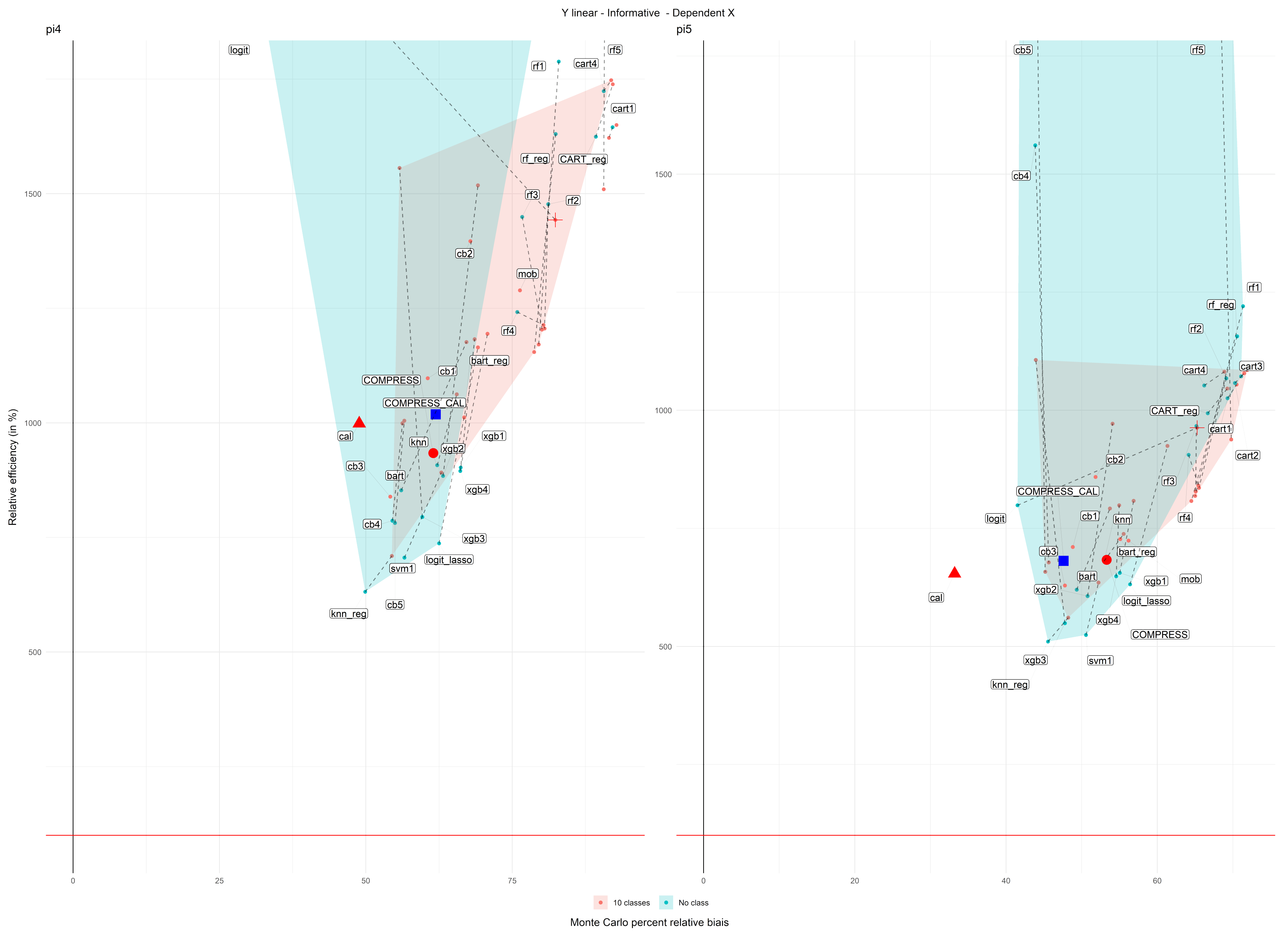

FIGURE 4 – Results for p4 and p5 when y is linear, X dependent and the design is informative.

5 Discussion

Bibliography

Références

[Boser et al., 1992] Boser, B. E., Guyon, I., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *COLT '92.*

[Breiman, 2004] Breiman, L. (2004). Random forests. *Machine Learning*, 45 :5–32.

[Breiman et al., 1983] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1983). Classification and regression trees.

[Brigitte Gelein and Causeur, 2018] Brigitte Gelein, D. H. and Causeur, D. (2018). PondÃ©ration pour correction de la non-rÃ©ponse totale et machine learning. *Acte des JMS 2018*, 1.

[Chen and Guestrin, 2016] Chen, T. and Guestrin, C. (2016). Xgboost : A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

[Chipman et al., 2010] Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). Bart : Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1).

[Dagdoug et al., 2020] Dagdoug, M., Goga, C., and Haziza, D. (2020). Imputation procedures in surveys using nonparametric and machine learning methods : an empirical comparison.

[Deville and Sarndal, 1992] Deville, J.-C. and Sarndal, C.-E. (1992). Calibration estimators in survey sampling. *Journal of the American Statistical Association*, 87(418) :376–382.

[Hastie et al., 2001] Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning.* Springer Series in Statistics. Springer New York Inc., New York, NY, USA.

[Haziza and Beaumont, 2007] Haziza, D. and Beaumont, J.-F. (2007). On the construction of imputation classes in surveys. *International Statistical Review*, 75 :25–43.

[Horvitz and Thompson, 1952] Horvitz, D. and Thompson, D. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260) :663–685.

[Lunceford and Davidian, 2004] Lunceford, J. K. and Davidian, M. (2004). Stratification and weighting via the propensity score in estimation of causal treatment effects : a comparative study. *Statistics in medicine*, 23 19 :2937–60.

[Narain, 1951] Narain, R. D. (1951). On sampling without replacement with varying probabilities. *Journal of the Indian Society of Agricultural Statistics*, 3 :169–175.

[Platt, 1999] Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press.

[Quinlan, 1992] Quinlan, J. R. (1992). Learning with continuous classes.

[Quinlan, 1993] Quinlan, J. R. (1993). Combining instance-based and model-based learning. In *ICML.*

[RUBIN, 1976] RUBIN, D. B. (1976). Inference and missing data. *Biometrika*, 63(3) :581–592.

[Rubin-Bleuer and Kratina, 2005] Rubin-Bleuer, S. and Kratina, I. S. (2005). On the two-phase framework for joint model and design-based inference. *The Annals of Statistics*, 33(6) :2789–2810.

[Silva and Skinner, 1997] Silva, P. N. and Skinner, C. (1997). Variable selection for regression estimation in finite populations. *Survey Methodology*, 23(1) :23–32.

[Zeileis and Hornik, 2007] Zeileis, A. and Hornik, K. (2007). Generalized m-fluctuation tests for parameter instability. *Statistica Neerlandica*, 61 :488–508.

[Zeileis et al., 2008] Zeileis, A., Hothorn, T., and Hornik, K. (2008). Model-based recursive partitioning. *Journal of Computational and Graphical Statistics*, 17 :492 – 514.

6 Annexes

6.1 Algorithm used

Label	Algorithm	Continuous or discrete variable	Hyperparameters
bart	BART	Discrete	
bart_reg	BART	Continue	
CART_reg	CART	Continuous	Min number of observations in each leaf : 20
cart1	CART pruned	Discrete	Min number of observations in each leaf : 10
cart2	CART pruned	Discrete	Min number of observations in each leaf : 20
cart3	CART pruned	Discrete	Min number of observations in each leaf : 30
cart4	CART non pruned	Discrete	Min number of observations in each leaf : 20
cb1	Cubist	Continuous	Not biased 10 agregated models With extrapolation
cb2	Cubist	Continuous	Biased 10 agregated models Without extrapolation
cb3	Cubist	Continuous	Biased 10 agregated models With extrapolation
knn	k-nearest neighbours	Discrete	
knn_reg	k-nearest neighbours	Discrete	
logit	Logistic regression	Discrete	
logit_lasso	Lasso logistic regression	Discreate	Lambda : obtained using 10-fold cross-validation
mob	Model-Based Recursive Partitioning	Discrete	Variable used for the stratification : $X^{(x)}$
rf_reg	Random forests	Continue	Min number of observations in each leaf : 20 Nombre d'arbres aggregÃ©s : 200
rf1	Random forests	Discrete (Probabilities estimation trees)	Min number of observations in each leaf : 10 Number of trees : 100
rf2	Random forests	Discrete (Probabilities estimation trees)	Min number of observations in each leaf : 10 Number of trees : 500
rf3	Random forests	Discrete (Probabilities estimation trees)	Min number of observations in each leaf : 30 Number of trees : 100
rf4	Random forests	Discrete (Probabilities estimation trees)	Min number of observations in each leaf : 30 Number of trees : 500
rf5	Random forests	Discrete (Probabilities estimation trees)	Min number of observations in each leaf : 30 Number of trees : 5E00
svm1	SVM with RBF kernel Platt method to get probabilities	Discrete	Gamma : 0.025 nu : 0.7
svm2	SVM with polynomial kernel	Discreate	Gamma : 0.0001 nu = 0.7 DegrÃ© = 1
xgb1	XGBoost	Continue	Number of trees : 500 Gamma : 10 Proportion for subset : 75% Learning rate : 0.05 Max deepth : 2
xgb2	XGBoost	Continue	Number of trees : 2000 Gamma : 2 Proportion for subset : 100% Learning rate : 0.5 Max deepth : 2
xgb3	XGBoost	Continue	Number of trees : 1000 Gamma : 1 Proportion for subset : 75% Learning rate : 0.01 Max deepth : 1
xgb4	XGBoost	Continue	Number of trees : 500 Gamma : 10 Proportion for subset : 75% Learning rate : 0.05 Max deepth : 3

TABLE 4 – Labels and hyperparameters for each algorithm

6.2 Efficiency of aggregation methods and number of methods aggregated

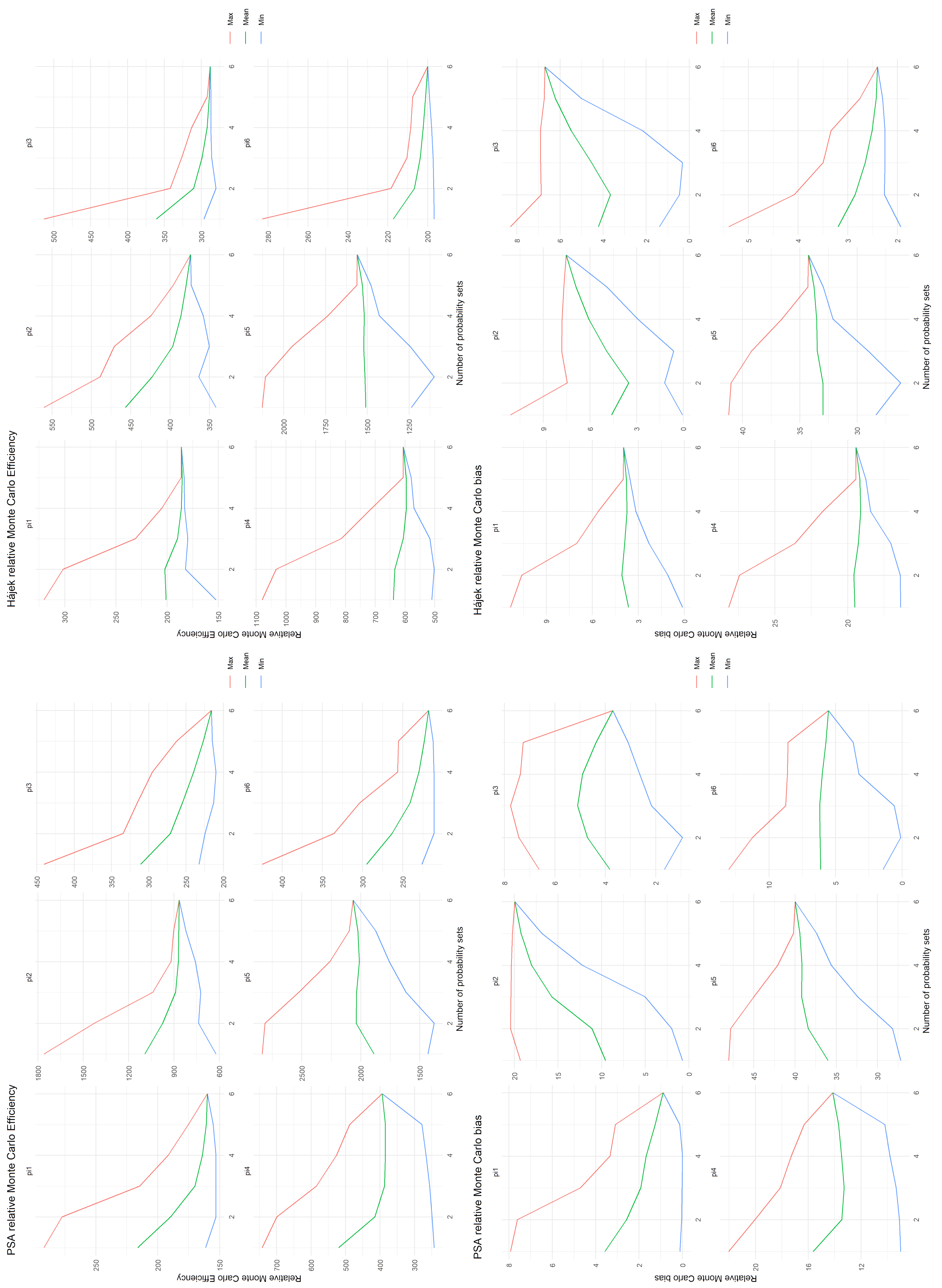

FIGURE 5 – Efficiency of COMPRESS methods

Non informative													
y linear													
β_0	$\beta^{(s)}$	$\beta_1^{(c)}$	$\beta_2^{(c)}$	$\beta_3^{(c)}$	$\beta_{12}^{(d)}$	$\beta_{13}^{(d)}$	$\beta_{14}^{(d)}$	$\beta_{15}^{(d)}$	$\beta_{22}^{(d)}$	$\beta_{32}^{(d)}$	$\beta_{34}^{(d)}$	$\beta_{35}^{(d)}$	$\beta_{33}^{(d)}$
0.5	-0.2	5.0	5.0	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

Informative													
y linear													
β_0	$\beta^{(s)}$	$\beta_1^{(c)}$	$\beta_2^{(c)}$	$\beta_3^{(c)}$	$\beta_{12}^{(d)}$	$\beta_{13}^{(d)}$	$\beta_{14}^{(d)}$	$\beta_{15}^{(d)}$	$\beta_{22}^{(d)}$	$\beta_{32}^{(d)}$	$\beta_{34}^{(d)}$	$\beta_{35}^{(d)}$	$\beta_{33}^{(d)}$
0.5	-10	5.0	5.0	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

Non informative													
y non linear													
δ_0	δ_1	δ_2	δ_3										
4	4	4	4										

TABLE 5 – Coefficients for each y-generator process.

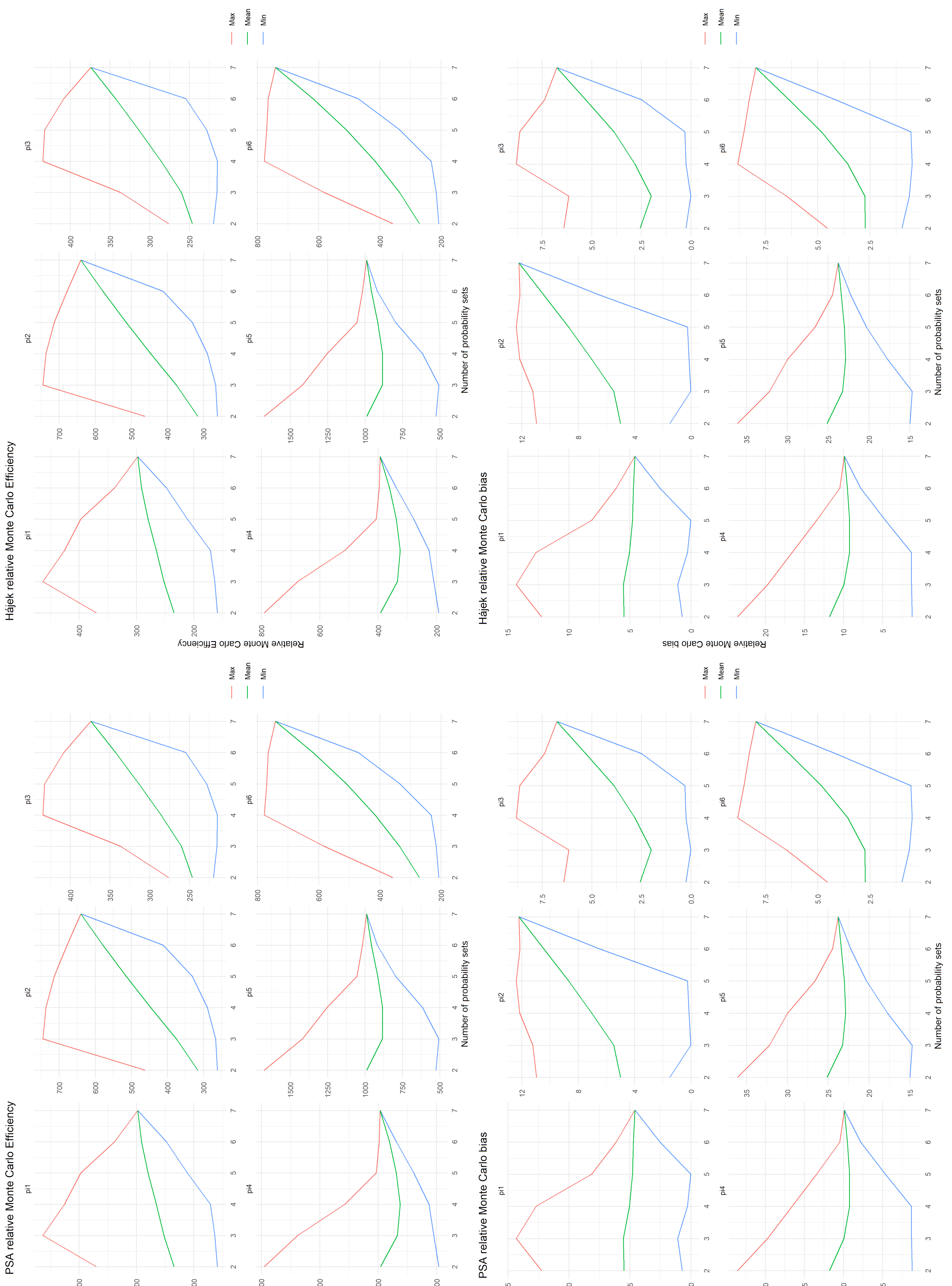

FIGURE 6 – Efficiency of calibration methods

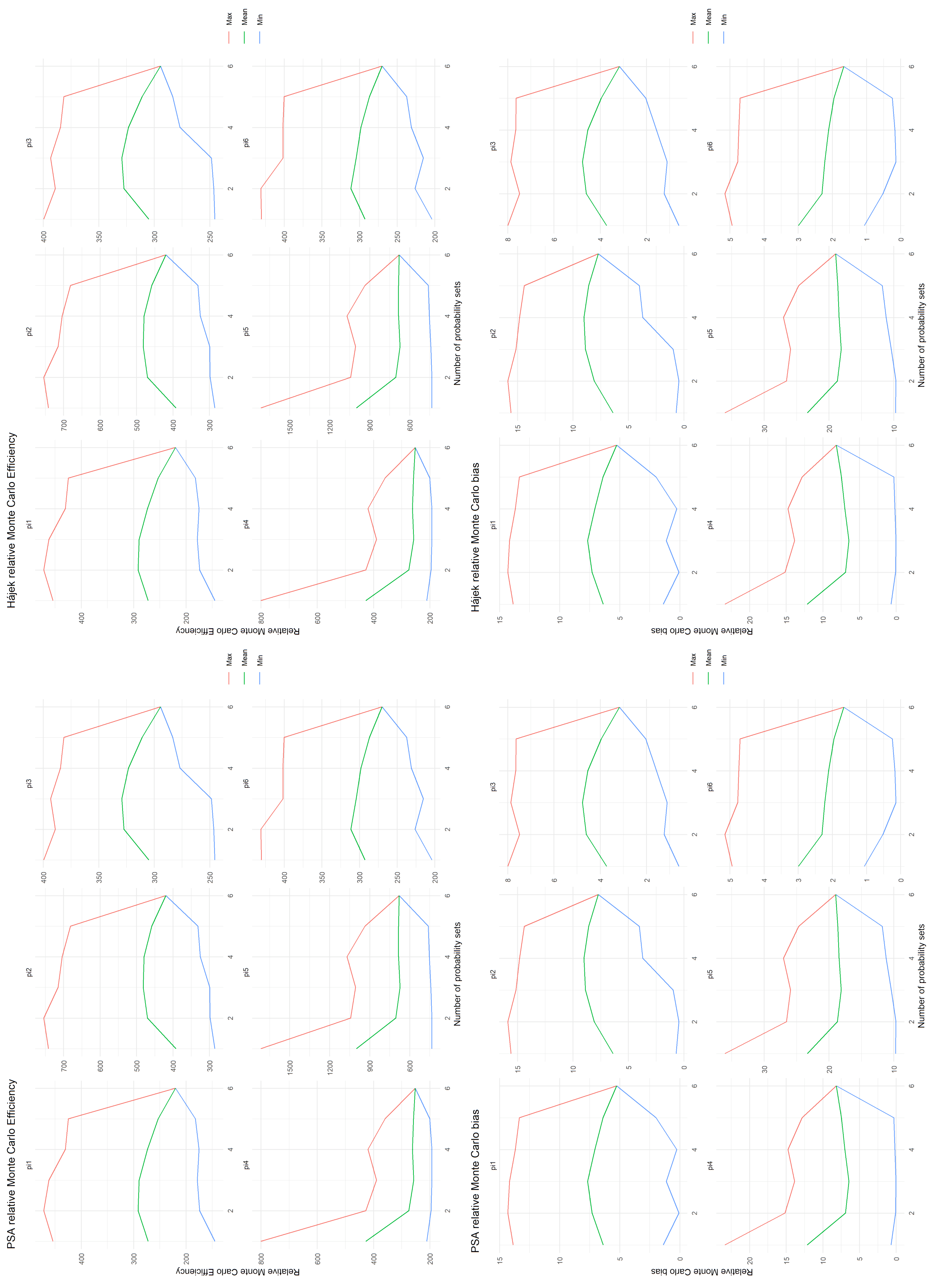

FIGURE 7 – Efficiency of COMPRESS + calibration methods