Sound and Solid Selection of Covariates - a Simulation Study

Kira Dynnes Svendsen (kisv@dtu.dk), Line Clemmensen,
Lars Kai Hansen, Bjarne Kjær Ersbøll *

Abstract

In this the era of big(ger) data, the all-time relevant question of how to detect the features which are truly relevant for an outcome of interest becomes paramount. As the amount of variables in data increases, the degree to which field knowledge is incorporated in the analysis decreases. As more and more automatized machine learning methods for handling and extracting information from data become easily accessible, an overview of the qualities and potential pitfalls of the contemporary and excessively used methods is pertinent.

Here we present the results of a simulation study assessing the performance of different methods for 'blindfolded' or 'field knowledge free' feature selection. We have considered Lasso, Forward selection, Elastic Net, Simplified Relaxed Lasso and two ad hoc methods.

The question of good performance and how to assess it is discussed and the methods are compared on a variety of different assessment measures both new and existing.

Key Words: Feature selection, Forward selection, Lasso, Relaxed Lasso, Simulation study, Variable selection

1. Introduction

The need to reduce the number of covariates in the linear model is a common problem in applied statistics. Classical methods like Forward and Best Subset Selection have been supplemented by penalized methods such as Lasso and Elastic Net, as the amount of data has grown larger and wider.

The main driver for this work is a hope that we can build an intuition about the different variable selection methods, and which method to prefer in which situations.

Quantifying a method calls for a good assessment metric. We will discuss the idea of good performance and use several measures, both new and existing to compare the methods. Data are simulated from a linear regression model using different combinations of width and hight (n, p) and different correlations.

Lastly we reflect upon the more philosophical and statistically theoretical considerations when talking about variable selection in general.

2. Models

In recent work by Bertsimas et al, the authors present an optimization method to solve *the classical best subset selection problem of choosing k out of p features in linear regression given n observations* [2] within a reasonable timeframe. Best Subset Selection has long been considered the best but unfeasible way to do sparse regression modelling; however, in a following article *Extended Comparison of Best Subset Selection, Forward Stepwise Selection, and the Lasso* [5] by Hastie et al. the authors apply Bertsimas new method to

*Technical University of Denmark, DTU Compute, Richard Petersens Plads, Bygning 324, 2800 Kgs. Lyngby, Denmark

simulated data and conclude, that in their setup, Best Subset Selection is not the best performing method of those they consider, and further, that it performs pretty much on par with Forward Selection. The models, assessment measures, and data structure in this work is intended to supplement their simulation study. We follow their choice regarding the Signal to Noise Ratio and values of the raw correlation ρ, but deviate in our choice of covariance structure. Where Hastie et al. look at an autocorrelation-type structure, we consider a block matrix correlation structure with a flat with-in block correlation. Regarding the choice of assesment we include most of the measures used by Hastie et al. which in turn are inherited from Bertsimas [2] but we also propose a new measure hoping to assess each model's capability to select the correct (or in some way almost correct) features. We have compared six methods for variable selection. Lasso, Relaxed Lasso and Forward Selection are all recycled from Hastie et al. but here in company of three alternatives.

Shortly on notation, we consider X to be a stochastic variable, and x to be a realization of this. Data consist of an outcome Y' with $y' \in \mathbb{R}^n$. The predictor matrix X' has $x' \in \mathbb{R}^{n \times p}$. We will consider a centered version of the outcome hereafter referred to as Y and y, and the standardized version of the predictor named X and x. Standardization and centering is just done for technical reasons as some of the methods work best on this type of data. Ultimately it has no impact and the parameters will usually be transformed back to the original scale.

Forward selection (FS)

Forward selection is an intuitively sensible and simple idea, but it has some obvious drawbacks and a history of being misinterpreted or even misused. It is prone to over-fitting, has somewhat poor predictive accuarcy, and the model resulting from a FS-regression has more often than not been solely reported without making it perfectly clear, that a selection has taken place, and that all results should be considered relative to this selection. As a variety of new and fancier methods have emerged throughout the past decades, it is no longer the go-to method for feature selection. It is however interesting to investigate how it actually performs as it is a method everybody knows. We evaluate Forward Selection using the implementation of the method from the `bestsubset` [4] package for `R` [9] developed by Hastie et al. in relation to their work in [5]: At step k, the set of indices for the covariates $\{1, \ldots, p\}$ is split up into $A_{(k-1)}$ (the $(k-1)$ variables in the model at that point in time) and $A^{\complement}_{(k-1)}$ (the rest). Each potential variable $(X_j)_{j \in A^{\complement}_{k-1}}$ is assessed by the *absolute correlation between the residual of the current model and the predictor X_j regressed on* $(X_i)_{i \in A_{k-1}}$. The predictor yielding the maximal absolute correlation is added to the model.

Lasso

Before Lasso was introduced, Forward Selection was often used if interpretablity was the main goal of the analysis. FS does not excel in prediction accuracy and Ridge Regression was found to be a reasonable alternative. By shrinking all parameters towards 0, these models achieved high accuracy but were on the other hand completely un-interpretable on larger (wider) amounts of data. Lasso was introduced as a compromise that would both shrink and select. The Lasso model is the solution to the least squares estimating equation supplemented with an l_1 penalty term:

$$\hat{\beta}^{lasso}(\lambda) = \text{argmin}_{\beta}\{\|Y - X\beta\|_2^2 + \lambda\|\beta\|_1\}$$

where $\lambda > 0$ is a hyperparameter to be tuned. In non-degenerated cases, the solution does not have a closed form. A popular implementation of Lasso is the `glmnet` package [3] in

R [9]. Here the Lasso model is fitted using *coordinate decent* to efficiently obtain numerical solutions. Other algorithmic approaches also exist. Note that we assume the data is scaled and the outcome is centralized. This way we avoid arbitrarily regularizing the intercept and assure equal weight of all covariates when penalizing.

Elastic Net (EN)

Elastic net was introduced in 2005 by Zou et al [10] to bring together the best of Ridge Regression and Lasso. Lasso tends to select only few variables from a group of highly correlated variables which is not always desirable. Further, it is limited to only selecting at most p variables. Elastic Net was designed to overcome these two limitations while keeping interpretability a priority. The Elastic Net solution combines the Ridge and Lasso penalty:

$$\hat{\beta}^{EN}(\lambda, \alpha) = \text{argmin}_{\beta}\{\|Y - X\beta\|_2^2 + \lambda \left[(1-\alpha)||\beta||_2^2/2 + \alpha||\beta||_1\right]\}$$

where $\lambda > 0$ and $\alpha \in [0, 1]$ are hyperparameters to be tuned - often by cross-validation. Note that for $\alpha = 0$ the solution is purely that of the Ridge Regression while $\alpha = 1$ yields the pure Lasso solution. Again we assume the data is scaled and the outcome is centralized.

Relaxed Lasso

In 2006 Meinshausen introduced the *Relaxed Lasso* to improve convergence of the Lasso for sparse high dimensional data [7]. The original two-step procedure assigns two different parameters to deal with first selection and then shrinkage. Hastie et al. consider a simplified version in [5]. This version combines the Lasso solution and the Least Squares solution restricted to the active set chosen by the Lasso:

$$\hat{\beta}^{Relaxo}(\lambda, \gamma) = \text{argmin}_{\beta,\gamma}\{\gamma\hat{\beta}^{Lasso}(\lambda) + (1-\gamma)\hat{\beta}^{LS}(\lambda)\}$$

where $\hat{\beta}^{LS}(\lambda)$ denotes the Least Squares solution restricted to the active set given by λ in the Lasso (note that 0's are filled in where needed to match dimension afterwards) and $\gamma \in [0, 1]$ weighs together the two solutions. As in the original Lasso, λ will do both selection and shrinkage however γ will now re-inflate the parameters towards the LS solution.

Ad hoc models

We also considered two two-step ad hoc estimators. First we rank the importance of all variables and secondly we fit a LS model to the variables more important than some threshold. To decide on the threshold, we add 5 dummy-variables with no information and variance equal to the average empirical variation found in the data. The threshold is set to be the value of the highest ranking dummy-variable plus one standard deviation of all 5. Visual inspection of the variable importance plot looking for an elbow is also a very nice way to decide on a threshold, but as it is difficult to automatize, we chose the simpler approach.

Random Forest OLS

Importance ranking is chosen to be the mean decrease in accuracy based on a forest of 500 trees.

Naive Bootstrap Lasso

The Naive Bootstap Lasso is inspired by the *Bolasso* introduced by Bach et al in [1]. Given a dataset (X, Y) we consider m bootstrap replications: $(\overline{X}, \overline{Y})_{i*}$ for $i^* \in \{1, \ldots, m\}$. Fitting the Lasso model to each replication yields m sets of support A_i^*, where $A_i^* = \{j | \hat{\beta}_j^{i^*, Lasso} \neq 1\}$. Fitting an LS estimator to the support given by the intersection over all replications $\hat{A} = \bigcap_{i^* \in \{1,..,m\}} A_i^*$ yields the Bolasso solution.

The Naive Bootstrap Lasso is less harsh. Instead of intersecting the supports, it ranks the importance of the covariates by the ratio of bootstrap data sets where the Lasso fit picked the covariate, that is $rank(X_j) = \frac{1}{m} \sum_{i^*=1}^{m} 1_{j \in A_i^*}$.

"The truth" as a baseline

To have a *sensible performance* baseline to relate to, we fit an OLS estimate to the true set of variables.

3. Data

Data is simulated from a linear regression model with normally distributed noise and a block covariance structure in the explanatory variables, that is:

$$
\begin{aligned}
&Y = X\beta + \varepsilon, \text{ where } \varepsilon \sim N(0, \sigma^2 I). \\
&X \in \mathbb{R}^{n \times p}, \text{ with } E(X) = 0 \text{ and } Cov(X) = diag(\Sigma_1, ..., \Sigma_k) \\
&\Sigma_i = \begin{bmatrix} 1 & \rho & \rho & \ldots & \rho \\ \rho & 1 & \rho & \ldots & \rho \\ \vdots & & \ddots & & \vdots \\ \rho & \ldots & \rho & 1 & \rho \\ \rho & \ldots & \rho & \rho & 1 \end{bmatrix}
\end{aligned}
$$

Due to limitations in time we only consider $k = 2$. To match the covariance structure, we consider the coefficient regime:

$$
\beta = \big(b_1, ..., b_{\frac{p_0}{2}}, \overbrace{0, ..., 0}^{\frac{(p-p_0)}{2}}, b_{\frac{p}{2}+1}, ..., b_{\frac{(p+p_0)}{2}}, \overbrace{0, ..., 0}^{\frac{(p-p_0)}{2}}\big)
$$

where p is the total number of variables in the data, p_0 is the number of active variables and coefficients b_i are given by $b_i = 1 \ \forall i$. We consider all combinations of the following data configurations:

$$
\begin{aligned}
&p \in \{12, 50, 100, 1000\}, \\
&p_0 \in \{6, 12\}, \\
&n \in \{100, 1000\} \\
\text{and} \\
&\rho \in \{0, 0.3, 0.7\}.
\end{aligned}
$$

In [5], the correlation structure was given by $Cov(X_i, X_j) = \rho^{|i-j|}$ and they considered four different coefficient regimes. We choose to look at the block structure, as it is a quite common assumption to make about real life data. In the special case of uncorrelated coefficients we overlap the setup of [5].

Upon deciding the size of σ^2, we cite a discussion of *fair signal to noise from ratio* leading to a function for σ^2 from Hastie et al. [5]:

Consider the conditional expectation of Y given X, $f(X) = E(Y|X)$, and define the signal-to-noise ratio $SNR := \frac{Var(f(X))}{Var(\varepsilon)}$. In our model we then have σ^2 given as a function of SNR and the covariance matrix: $SNR = \frac{Var(f(X))}{Var(\varepsilon)} = \frac{Var(\beta X)}{\sigma^2} = \frac{\beta^T \Sigma \beta}{\sigma^2}$ hence $\sigma^2 = \frac{\beta^T \Sigma \beta}{SNR}$. The proportion of variance explained (PVE) by a given estimating function g is given by $PVE(g) = 1 - \frac{E[[Y-g(X)]^2]}{Var(Y)}$. By design f maximizes PVE and $PVE(f) = \frac{SNR}{1+SNR}$ is therefore an upper limit for PVE. Hastie et al.'s experience with PVE in real observational data is, that you rarely come across observational data with a PVE above 0.5, and a value of 0.86 is considered extreme. Following [5] we therefore restrict ourselves to looking at SNRs equidistantly spaced on a logarithmic scale between 0.05 and 6 corresponding to an upper limit of PVE ranging between 0.04761 and 0.8571.

4. Assessment measures

When is a method good? What is it actually we want, when we talk about variable selection. Usually we analyse data with two goals, and it might not be the same model or method that we should use to answer to both needs. On the one hand we want a model which predicts well on future observations, on the other hand, we want a model which explains the (key) dynamics that generated the data. We need assessment measures that reflect our interest in both performance, and the more fluffy quality 'interpretability' of the model. How well does the model actually uncover the true signals in the data, and to what degree are superfluous variables removed. We have applied the following measures to address these questions:

Relative test error (RTE)

The relative test error is the amount of variation in the data explained by the fitted model $(x\hat{\beta})$, relative to the irreducible error:

$$RTE(\hat{\beta}) = \frac{E[Y-X\hat{\beta}]^2}{\sigma^2} = \frac{(\beta-\hat{\beta})^T E[X^T X](\beta-\hat{\beta})+E[\varepsilon^T \varepsilon]}{\sigma^2} = \frac{(\beta-\hat{\beta})^T \Sigma(\beta-\hat{\beta})+\sigma^2}{\sigma^2}.$$

A perfect model fit, that is $\hat{\beta} = \beta$, corresponds to $RTE(\beta) = 1$. The null model, that is $\hat{\beta} = 0$ yields $RTE(0) = SNR + 1$.

Proportion of variance explained (PVE)

As described in section 3 the proportion of variance explained is defined by:

$$PVE(\hat{\beta}) = 1 - \frac{E[[Y-X\hat{\beta}]^2]}{Var(Y)} = 1 - \frac{(\beta-\hat{\beta})^T \Sigma(\beta-\hat{\beta})+\sigma^2}{\beta^T \Sigma \beta+\sigma^2}.$$

We recall the perfect score being $\frac{SNR}{1+SNR}$, the null score is $PVE(0) = 0$.

Proportion of selected variables (PSV)

We count the number of variables in the fitted model relative to the correct number from the data-generating-model.

$$PSV(\hat{\beta}) = \frac{\sum_i 1_{\hat{\beta}_i \neq 0}}{\sum_i 1_{\beta_i \neq 0}}.$$

The optimal score is 1, the null score is 0.

Proportion of selected correct variables (PSCV)

We count the number of variables in the fitted model that coincide with the variables in the data-generating-model relative to the total number from the data-generating-model.

$$PSCV(\hat{\beta}) = \frac{\sum_i 1_{\beta_i \neq 0 \wedge \hat{\beta}_i \neq 0}}{\sum_i 1_{\beta_i \neq 0}}.$$

The optimal score is 1, the null score is 0.

Naive measure of proxyness in the model (NP)

The idea behind both PSV and especially PSCV is based on the premise, that we should find the right variables in order to *perform well*. With highly correlated variables, this seems to some extent an unfair premise. We will return to this discussion. For now, we propose a naive measure which assigns some goodwill to a model that finds a variable that is not 'correct', but at least somewhat close. Given a model fit, for each variable found, we assign the value 1 if the variable was actually in the data-generating-model. If not it will get the maximum of covariance between the chosen variable and all variables in the data-generating-model (recall that as $Var(X) = 1$, the correlation and covariance are just the same). For each variable not found which was not suppose to be there we assign the value 1. These 'scores' are summed and seen relative to the total number of variables in the data-generating-model.

$$NP(\hat{\beta}) = \frac{1}{p} \sum_i 1_{\beta_i \neq 0 \wedge \hat{\beta}_i \neq 0} + 1_{\beta_i = 0 \wedge \hat{\beta}_i = 0} + 1_{\beta_i = 0 \wedge \hat{\beta}_i \neq 0} \cdot \max_j \{Cov(X_i, X_j) \cdot 1_{\beta_j \neq 0}\}$$

The optimal score is 1, the null score is $1 - \frac{p_0}{p}$.

Prediction accuracy

We calculate the prediction accuracy of the model fit on an independent test data set generated by the same data-generating-model. For consistency we set $n = 1000$ regardless of how many observations the model fit is based on.

5. Simulations and tuning

All simulations, model fits and assessments calculations were done in `R` [9]. Lasso and Elastic Net were fitted in the `glmnet` [3] package and tuned over the default hyperparameter set for λ. For Elastic Net we used 20 equidistant values of $\alpha \in [0, 1]$ (both endpoints included). Forward Selection was fitted using the function `fs` from the `bestsubset` [4] package. It is implemented as described above, and we use a default of $\min\{n, p, 2000\}$ steps. Relaxed Lasso was fitted using `lasso` from the `bestsubset` package. The function is a wrapper for `glmnet` with a parameter `nrelax` corresponding to γ in the description of Relaxed Lasso above. We used 10 values of γ equidistantly distributed from 0 to 1 both included. In total for each dataset we fitted 100×10 Relaxed Lasso models. For the Random Forest OLS we used the function `randomForest` from the package of the same name [6] using the default of 500 trees. Further we set the option `importance = TRUE` to asses the importance measures of the predictors. In the Naive Bootstrap Lasso we bootstrap $m = 10$ datasets.

Computation time of each simulation applying all methods and all measures, ranged from 1 to 25 minutes depending on the setting. Simulations were run on an HP probook, 32GB RAM, Intel core i7-7500u cpu @ 2.70GHz.

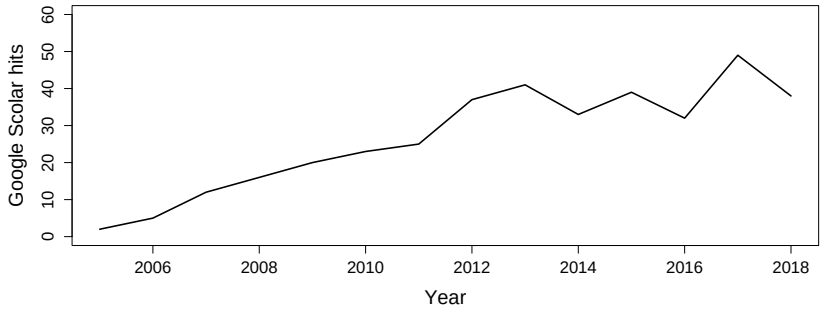


Figure 1: All documents registered on Google Scholar holding the phrase "Relaxed Lasso" in any field (keywords, title, text etc.)

6. Results

We have included a small selection of the results in the appendix: For all three values of $\rho \in \{0, 0.35, 0.7\}$ we consider simulated data from a data-generating-model with $p_0 = 6$ in a low ($p = 12$), medium ($p = 100$) and high ($p = 1000$) setting. For the low and medium setting we consider both $n = 100$ and $n = 1000$, in the high setting only $n = 100$ (due to time constraints). The results are plotted by assessment measure with all 6 models in each plot (the true fit is included as baseline). We report the average with 1 SE bands based on 10 simulations in the plots.

No conclusions from this study are as clear cut, bright and shiny as we had hoped for. The main takehome message is, that the simplified Relaxed Lasso introduced in [5] deserves more attention than it is currently given in the literature: a raw Google Scholar search on 'relaxed lasso' yielded only 369 hits in total throughout the last 13 years (Figure 1). It outperforms all other methods on all assessment measures regardless of the setting. This is in agreement with the main conclusions from Hastie et al. in [5]. The seemingly superiority of the method is thus not affected by the change in correlation structure and evaluation by additional assessments measures. Neither of the additional methods we look at are fair all-round competitors to the Relaxed Lasso. Lasso and Elastic Net behave somewhat similar. They manage quite well on the traditional assessment measures (PVE, RTE and MSE), but seem to do that by being very greedy in the sense that they include far too many features. They seem unaffected by the magnitude of the correlation. The Forward Selection also does a fair job on these measures. Compared with the Relaxed Lasso it is a little more reluctant to include features - which in turn makes it the lowest scoring when it comes to finding exactly the right features. This is on the other hand the reason for its good performance on the Naive Beta measure, as it gets a lot of credit for not including irrelevant features.

The two ad hoc measures Random Forest OLS and Bootstrap Lasso do not impress in their current state. The Random Forest OLS is surprisingly poor performing in all setups. This is probably due to the fact that the data actually has a linear structure which the Random Forest is quite ill suited to fit. The Random Forest is much celebrated when the structure is non-linear or unknown, but it seems to backfire using it when a linear structure is truly present in the data. The Bootstrap Lasso might on the other hand have some potential. In short the general performance decreases very much when the data widens (p grows). Larger n will soften the fall but not enough. On top of that, the predicting power is definitely the worst of all methods (regardless of p and n). However the explainability measures and

PVE and RTE are all competitive even to the Relaxed Lasso in the low setting. The poor performance on the fat data seems to be due to a unfortunate choice of cut-off after the ranking process. Only using 5 dummy variables no matter the size of data simply will not do the job. As p grows the probability that the dummy variables will ever be selected decreases which in turn results in all variables ever picked out in any bootstrap data set being included in the final model. Simple adjustments should be imposed on the cut-off method to correct for this. A humble suggestion is to include $r = \frac{p}{3}$ dummy variables. Increasing the number of bootstrap datasets might also help. This is a subject for further simulations.

Regarding the Naive measure of proxyness (NP) we note, that on this data it never really gets a fair chance to differentiate between different choices of 'wrong' variables. As all variables within a block are equally correlated, choosing a 'wrong' variable will always contribute with ρ instead of 1 to the nominator - hence all 'wrong' variables are equally 'wrong'. Whether the measure is useful in addressing the question of not only selecting the right variables but also those "looking like" the right variables remains to be investigated. We note however, based on this experience, that the measure seems to be in need of some tuning. We suggest two possible refinements which will draw the measure in two different directions:

1 Based on a 'better safe that sorry' approach the reward for picking a true variable should be somewhat higher than that for not choosing a wrong one.

2 Incorporation of the estimated effect size of the variable in question into the measure. The rationale being that, it might be 'wrong' but as the coefficient is neglectable - we will not really look at it anyway - so it doesn't matter so much and we will punish it less.

Further, for the sake of completeness, one should probably consider involving the absolute value of the correlation instead (as we only considered weakly positive correlations this was not necessary). These thoughts tap back into a more philosophical discussion of what is it we are actually looking for and what is good.

7. A reflection at the end

How does the idea of *finding the right variables* make sense? Does it make sense? When *true* and *non true* variables are highly correlated, is it then wrong to pick the *non true*, which undoubtedly has the same signal as the *true*? Does this set-up even exist in real data? Is it not more likely, that if the correlation is certain, the observed variables are, to some extent, driven by (some of) the same latent forces. Maybe it would be a fairer comparison, if the models were applied to data generated from a latent model, and the assessment reflected the degree to which the latent structure is excavated. Variable selection is a very difficult craft, it's easy to do wrong, but is it even possible to do well? The need for giving data a diet becomes more and more inevitable as the amounts of data grow in both directions - especially when data becomes fat or even obese. The initial selection is already performed before we start working with the data by a combination of nature and choice. Some things are unmeasureable, then we might try with proxies sometimes good some times bad. Other things might be expensive or time consuming and hence we leave them out - but that is also selection. Good field knowledge might give you reason to eliminate specific variables, but if your data holds information on thousands of things incorporating this knowledge becomes very difficult. This work has been a reminder of the humility which is fundamental when analytic results are being presented. It is of enormous importance that the whole story is told - not just the result.

References

[1] Francis R Bach. "Bolasso: model consistent lasso estimation through the bootstrap". In: *Proceedings of the 25th international conference on Machine learning*. ACM. 2008, pp. 33–40.

[2] Dimitris Bertsimas, Angela King, Rahul Mazumder, et al. "Best subset selection via a modern optimization lens". In: *The annals of statistics* 44.2 (2016), pp. 813–852.

[3] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. "Regularization Paths for Generalized Linear Models via Coordinate Descent". In: *Journal of Statistical Software* 33.1 (2010), pp. 1–22. URL: http://www.jstatsoft.org/v33/i01/.

[4] Trevor Hastie, Rob Tibshirani, and Ryan Tibshirani. *bestsubset: Tools for best subset selection in regression*. R package version 1.0.6. 2017.

[5] Trevor Hastie, Robert Tibshirani, and Ryan J Tibshirani. "Extended Comparisons of Best Subset Selection, Forward Stepwise Selection, and the Lasso". In: *arXiv preprint arXiv:1707.08692* (2017).

[6] Andy Liaw and Matthew Wiener. "Classification and Regression by randomForest". In: *R News* 2.3 (2002), pp. 18–22. URL: https://CRAN.R-project.org/doc/Rnews/.

[7] Nicolai Meinshausen. "Relaxed lasso". In: *Computational Statistics & Data Analysis* 52.1 (2007), pp. 374–393.

[8] Python Core Team (2018). *Python: A dynamic, open source programming language*. Python Software Foundation. 2018. URL: URL%20https://www.python.org/.

[9] R Core Team. *R: A Language and Environment for Statistical Computing*. version 3.4.4. R Foundation for Statistical Computing. Vienna, Austria, 2018. URL: https://www.R-project.org/.

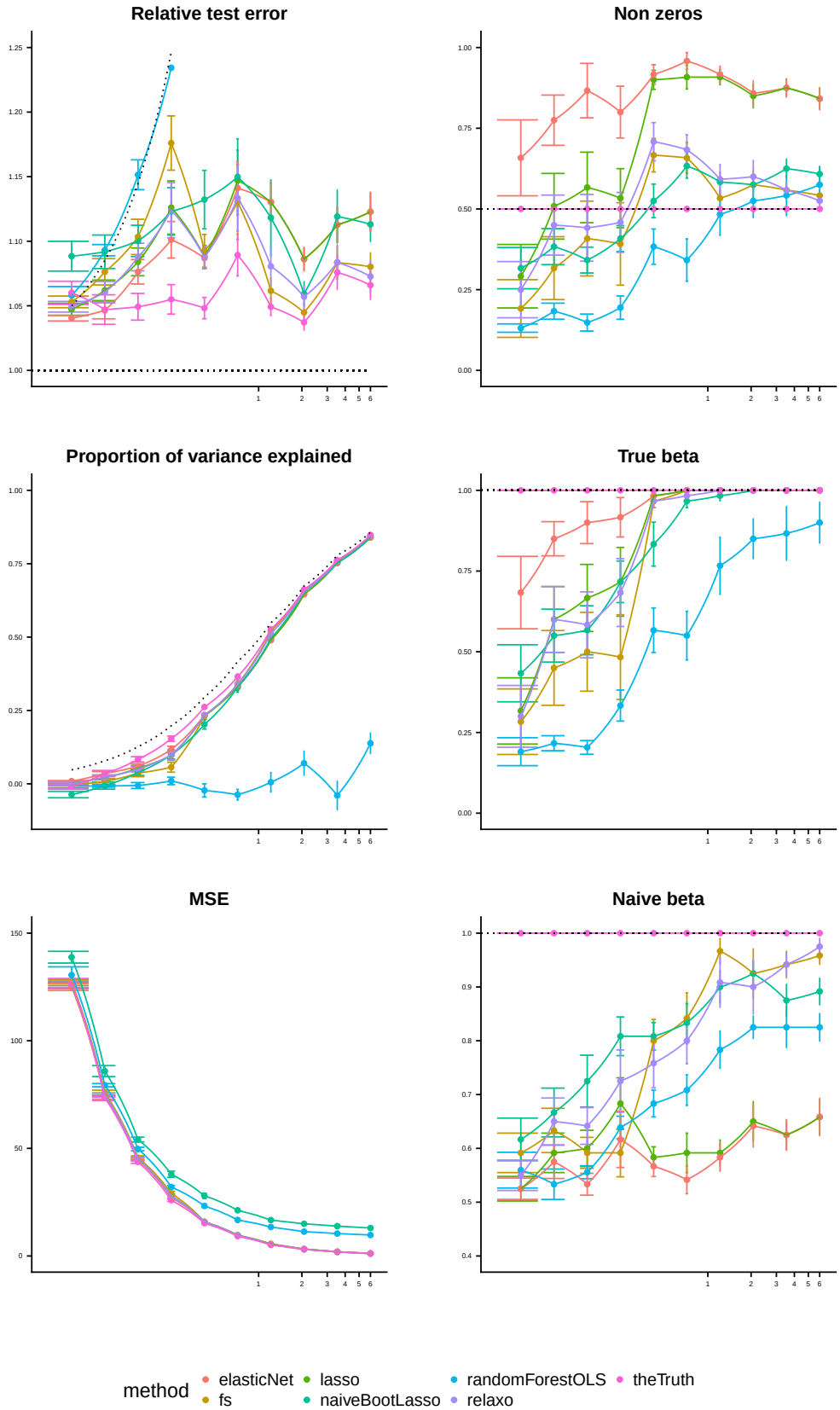
[10] Hui Zou and Trevor Hastie. "Regularization and variable selection via the elastic net". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2 (2005), pp. 301–320.

Appendix

Result plots

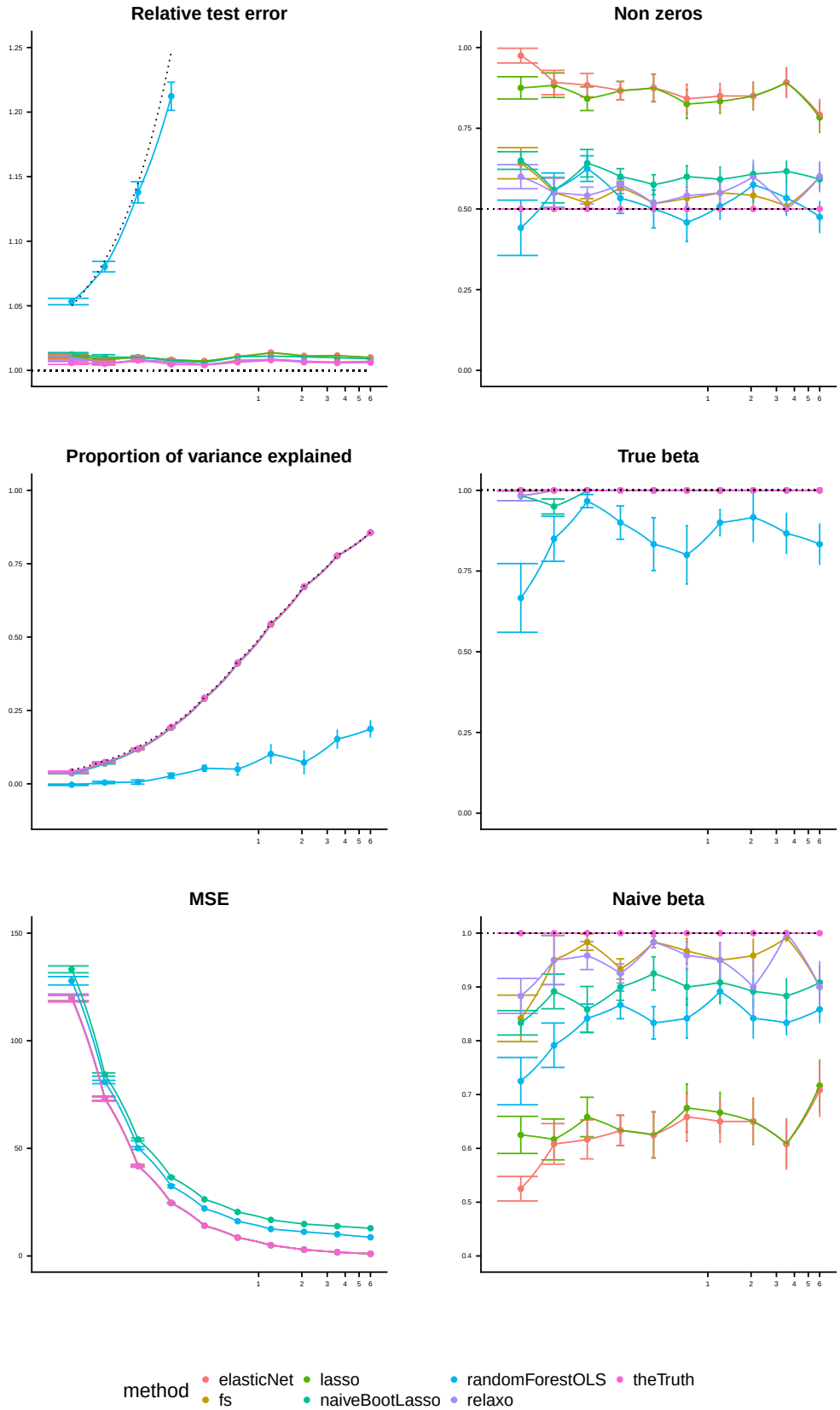
Low setting

$\rho = 0$, $n = 100$, $p = 12$ and $p_0 = 6$



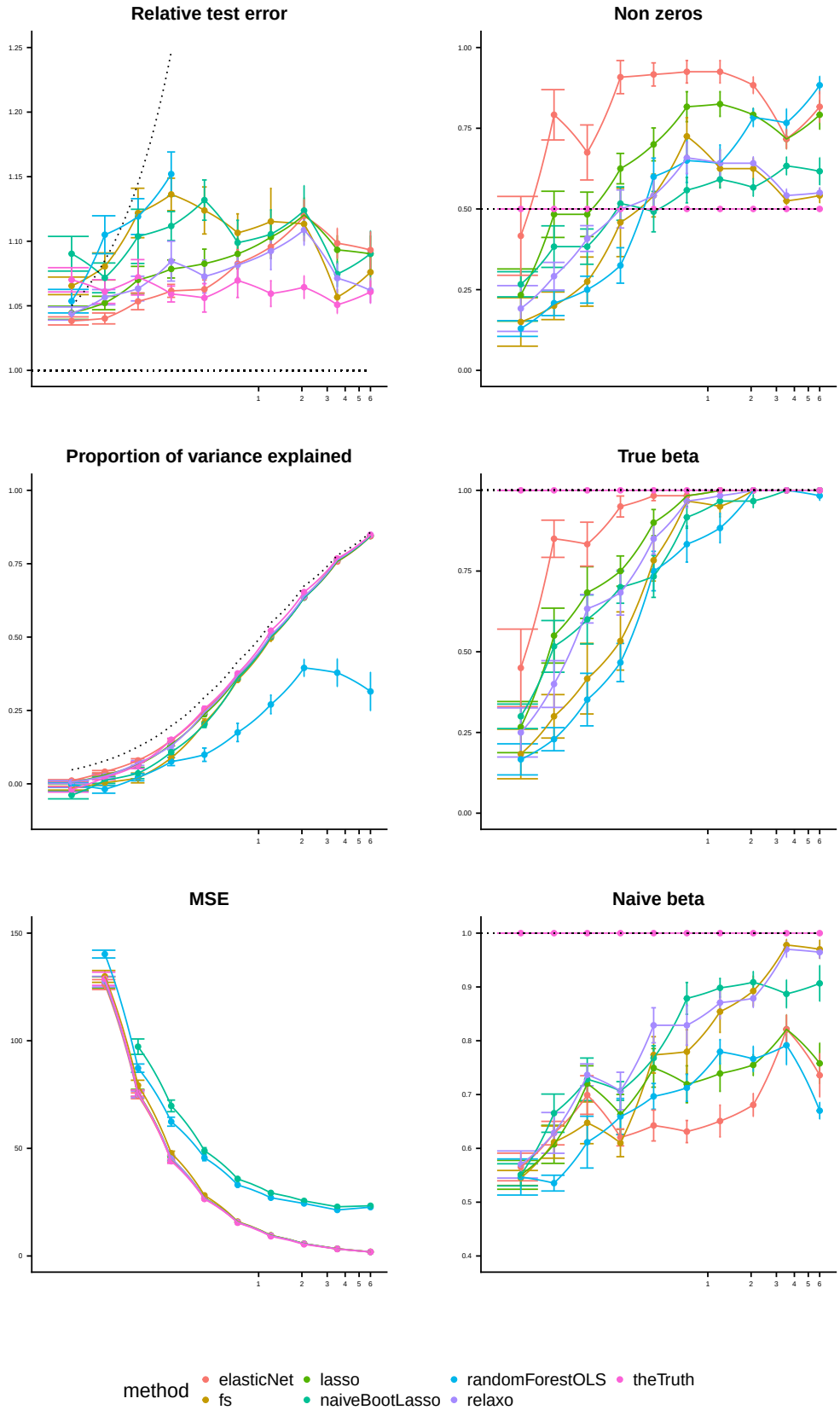
Low setting

$\rho = 0$, $n = 1000$, $p = 12$ and $p_0 = 6$



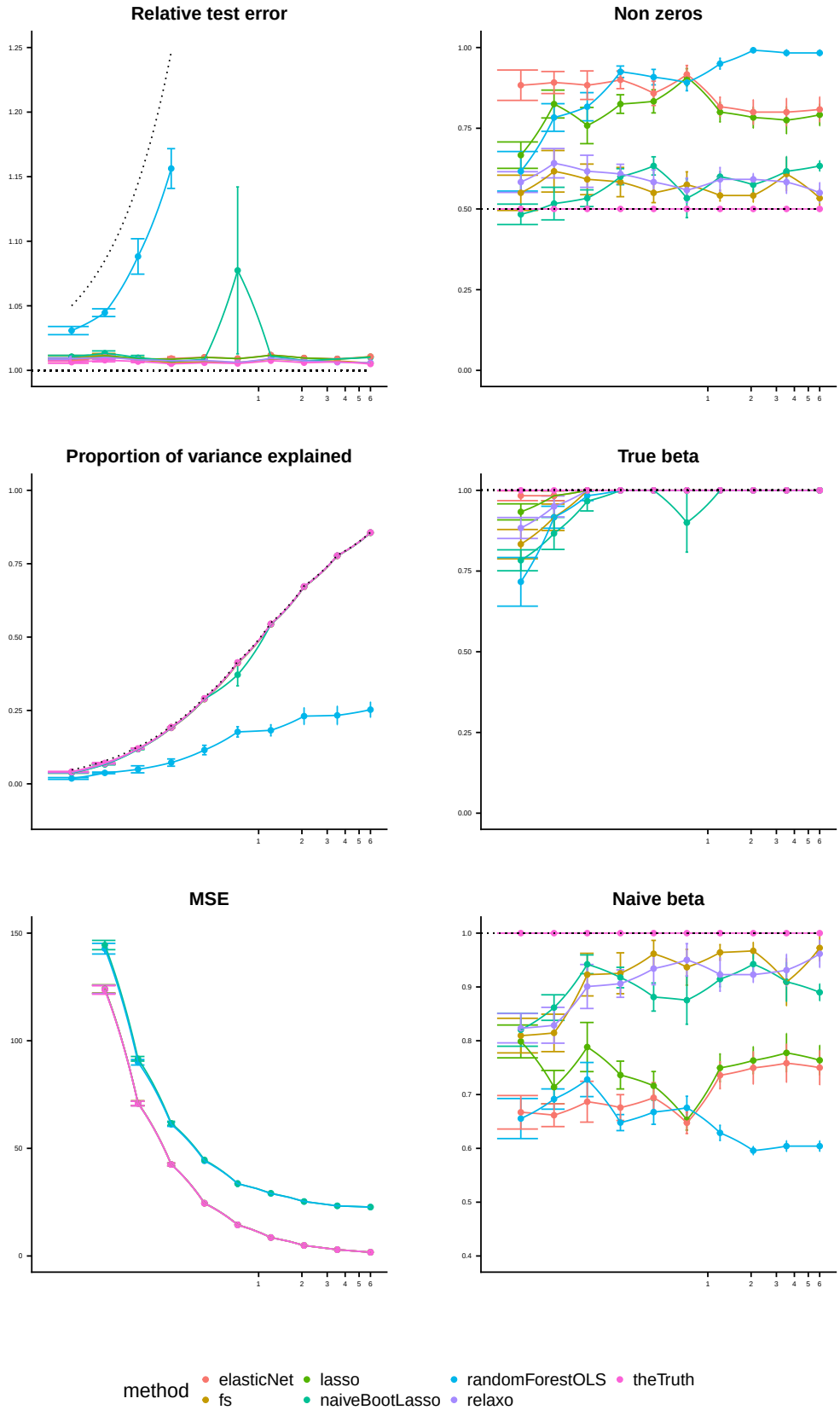
Low setting

$\rho = 0.35$, $n = 100$, $p = 12$ and $p_0 = 6$



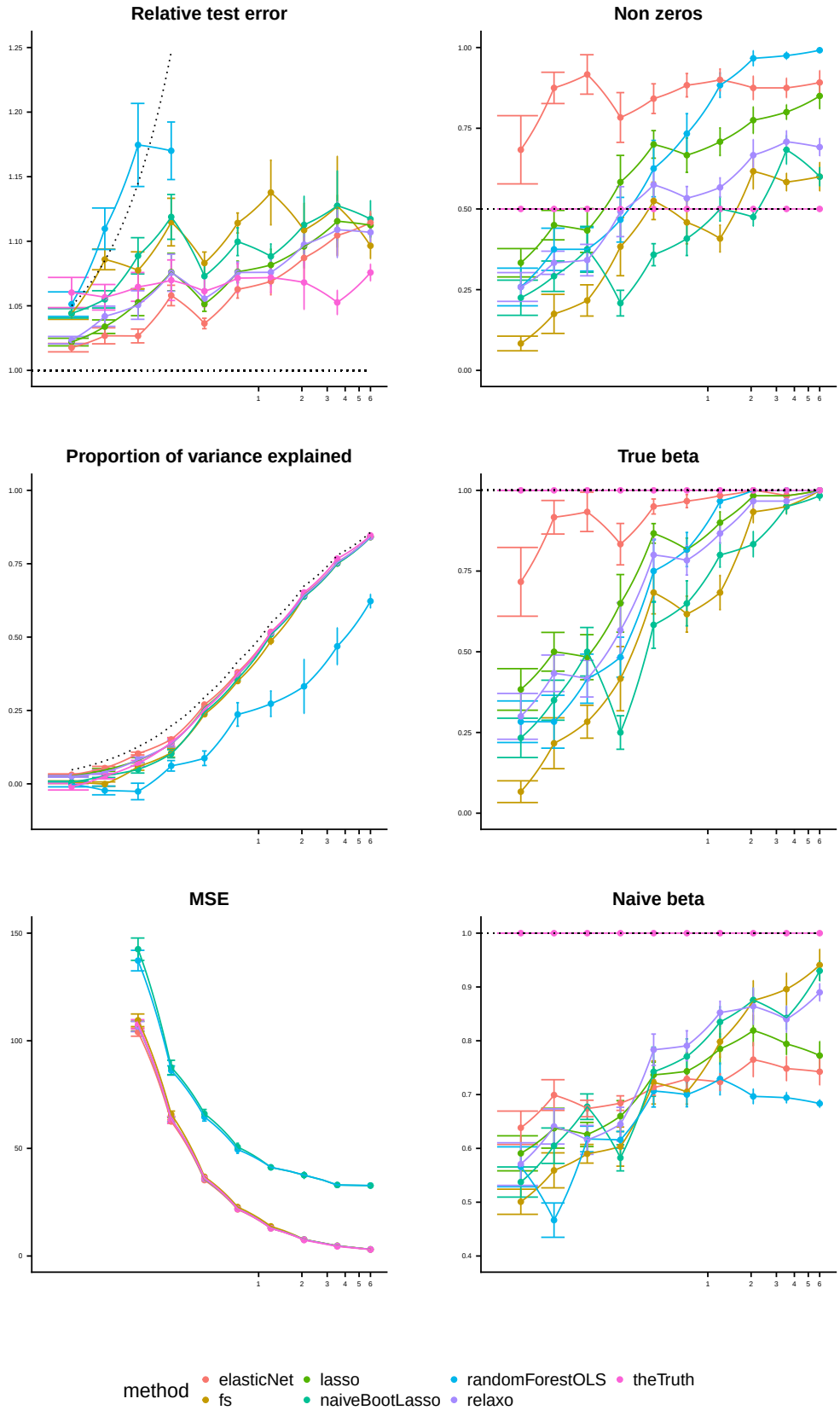
Low setting

$\rho = 0.35$, $n = 1000$, $p = 12$ and $p_0 = 6$



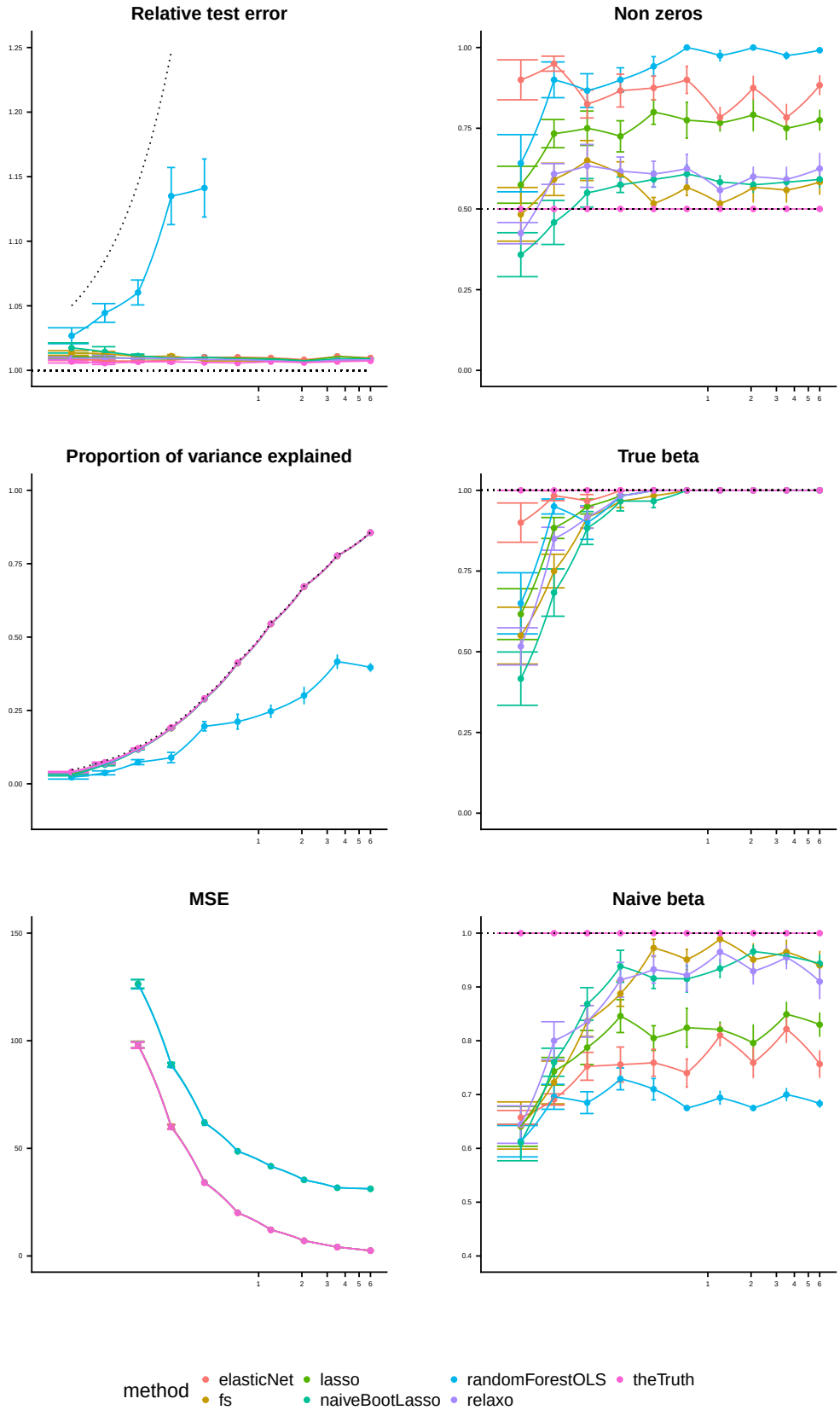
Low setting

$\rho = 0.7$, $n = 100$, $p = 12$ and $p_0 = 6$



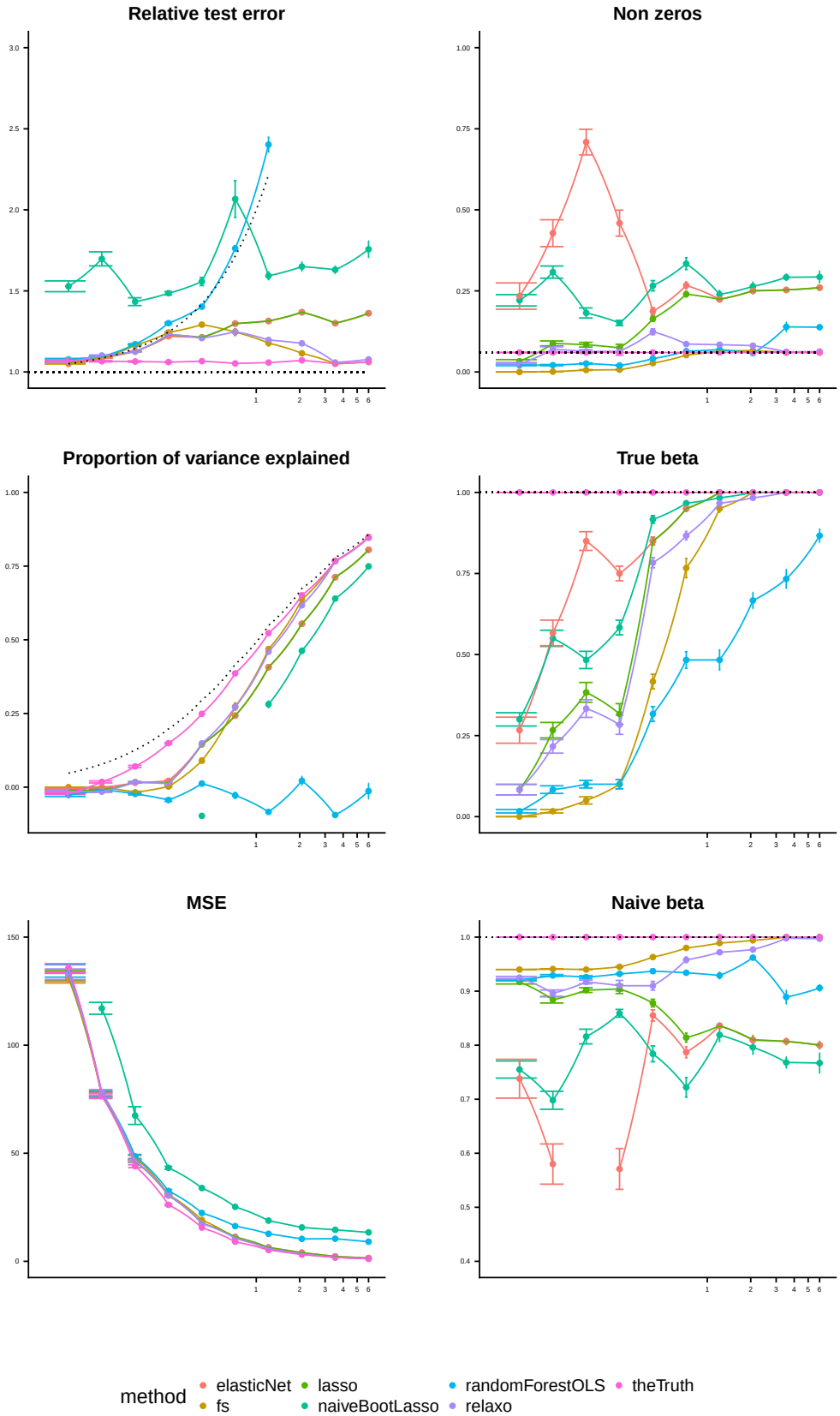
Low setting

$\rho = 0.7$, $n = 1000$, $p = 12$ and $p_0 = 6$



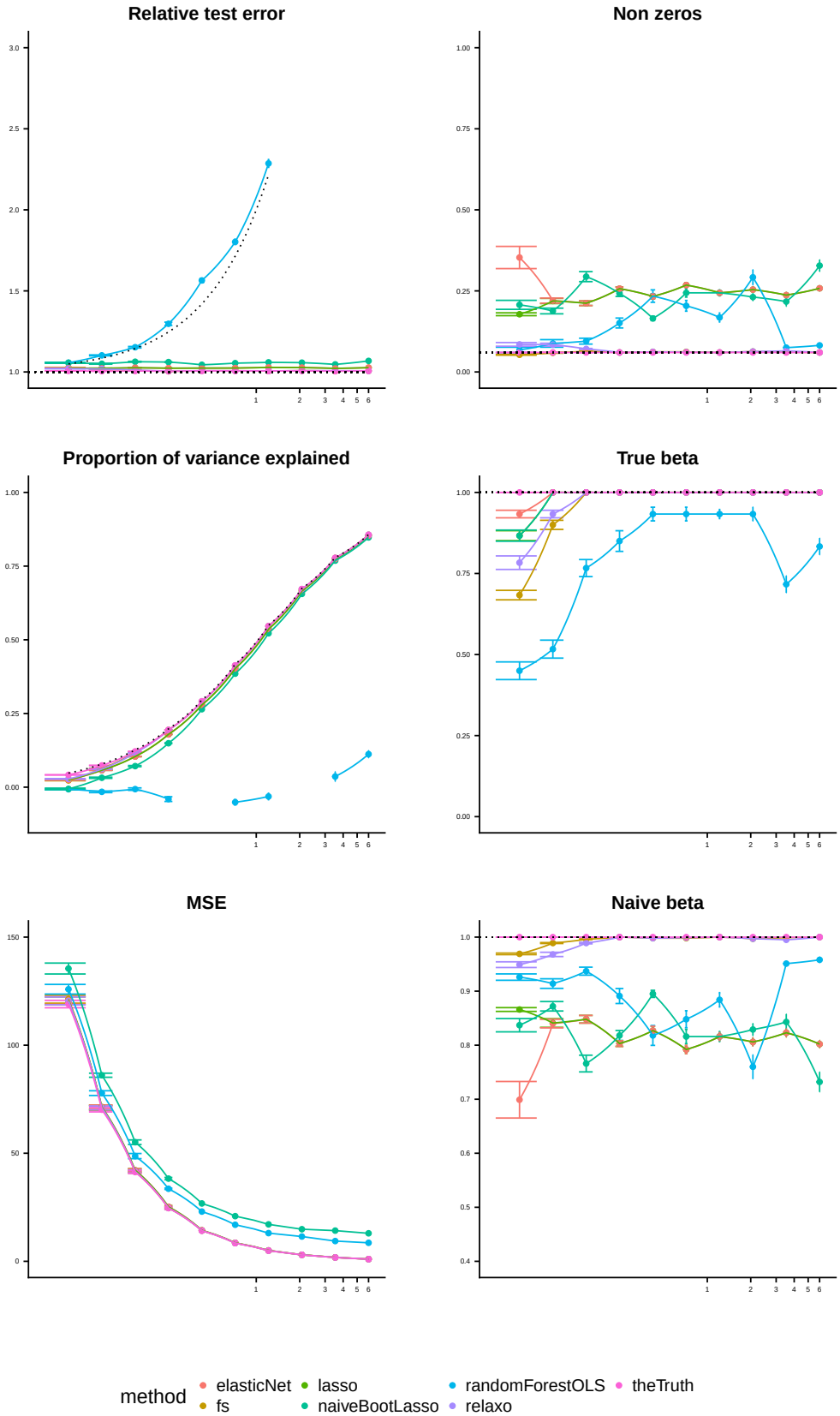
Medimum setting

$\rho = 0$, $n = 100$, $p = 100$ and $p_0 = 6$



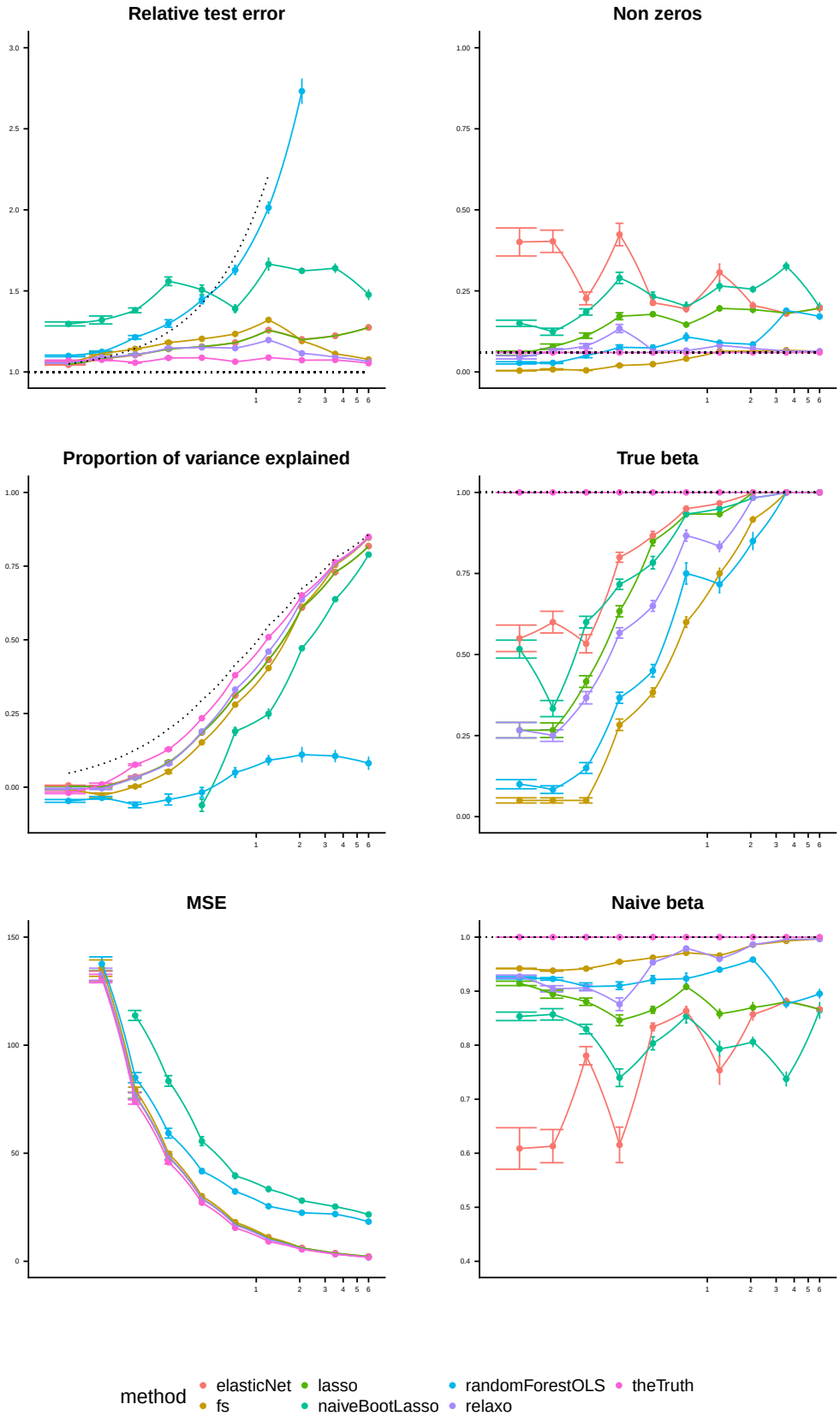
Medium setting

$\rho = 0$, $n = 1000$, $p = 100$ and $p_0 = 6$



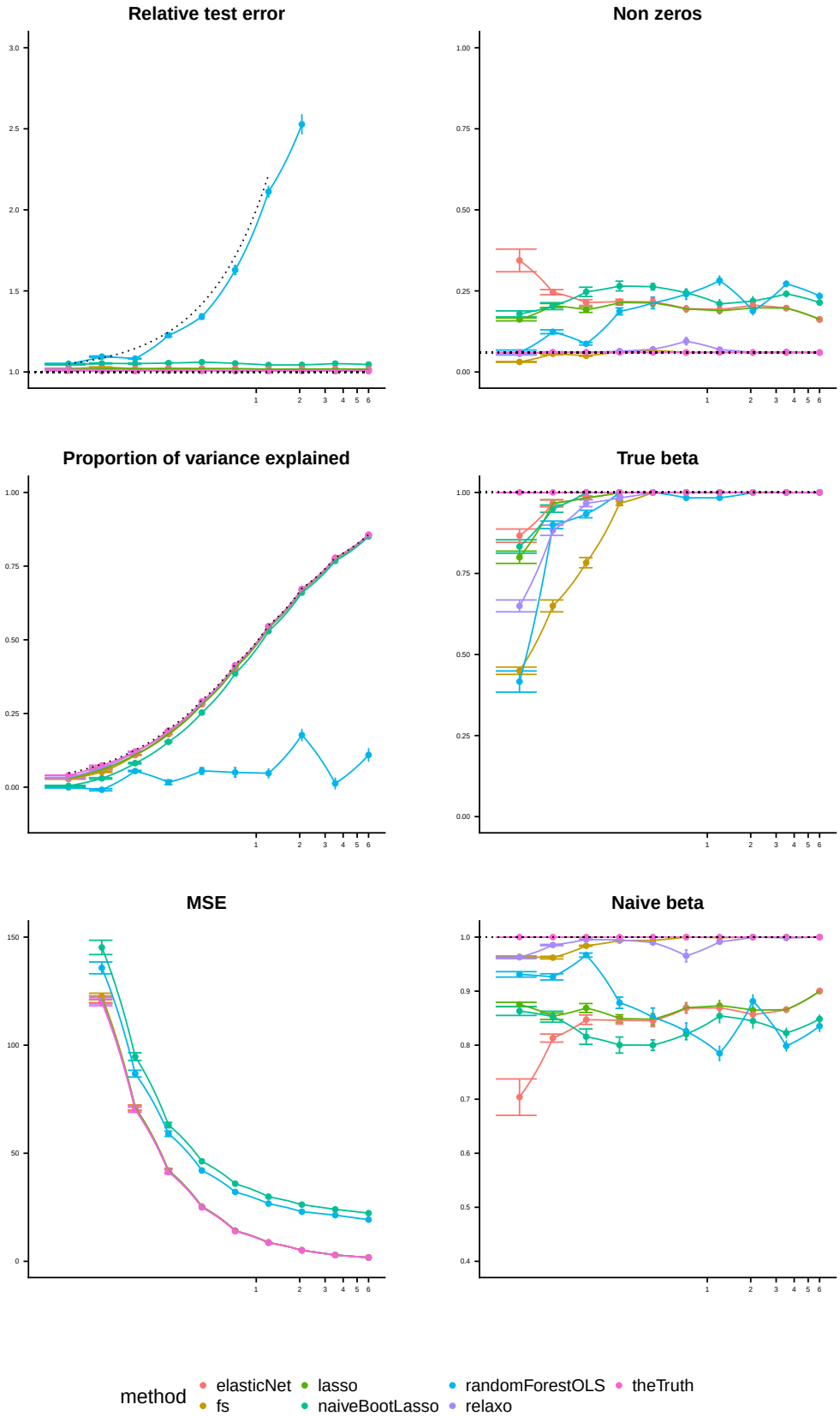
Medimum setting

$\rho = 0.35$, $n = 100$, $p = 10$ and $p_0 = 6$



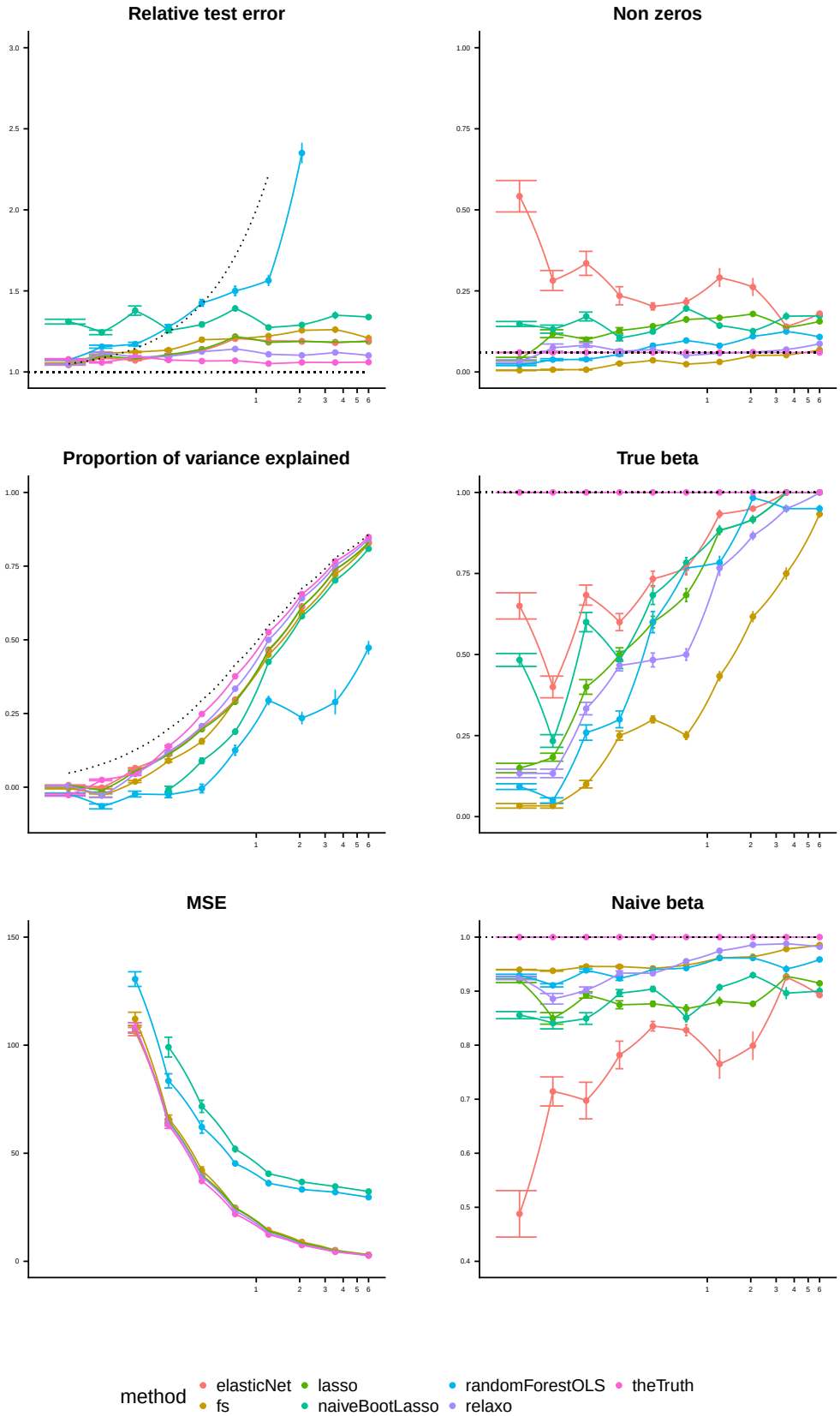
Medium setting

$\rho = 0.35$, $n = 1000$, $p = 100$ and $p_0 = 6$



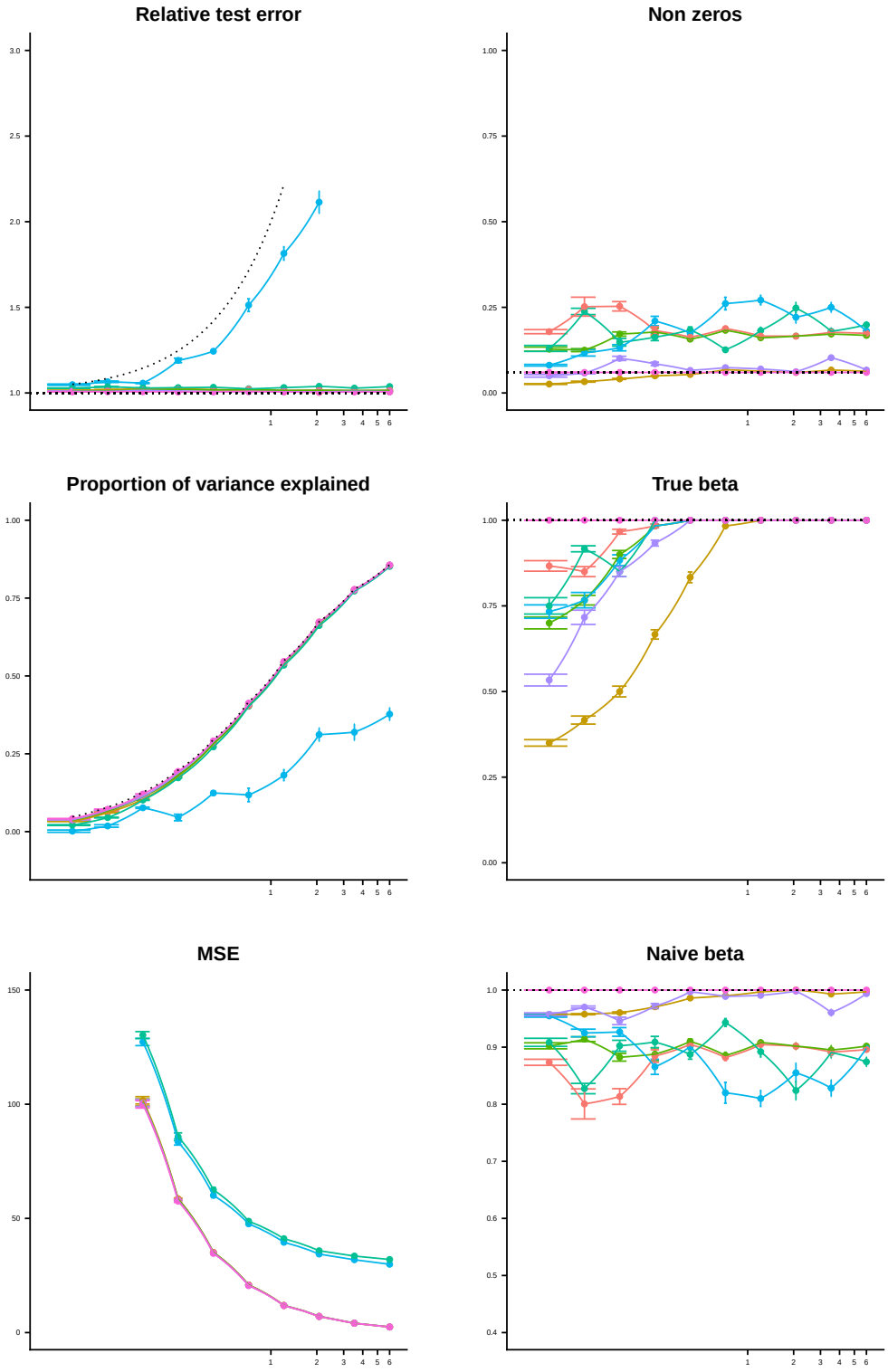
Medimum setting

$\rho = 0.7$, $n = 100$, $p = 100$ and $p_0 = 6$



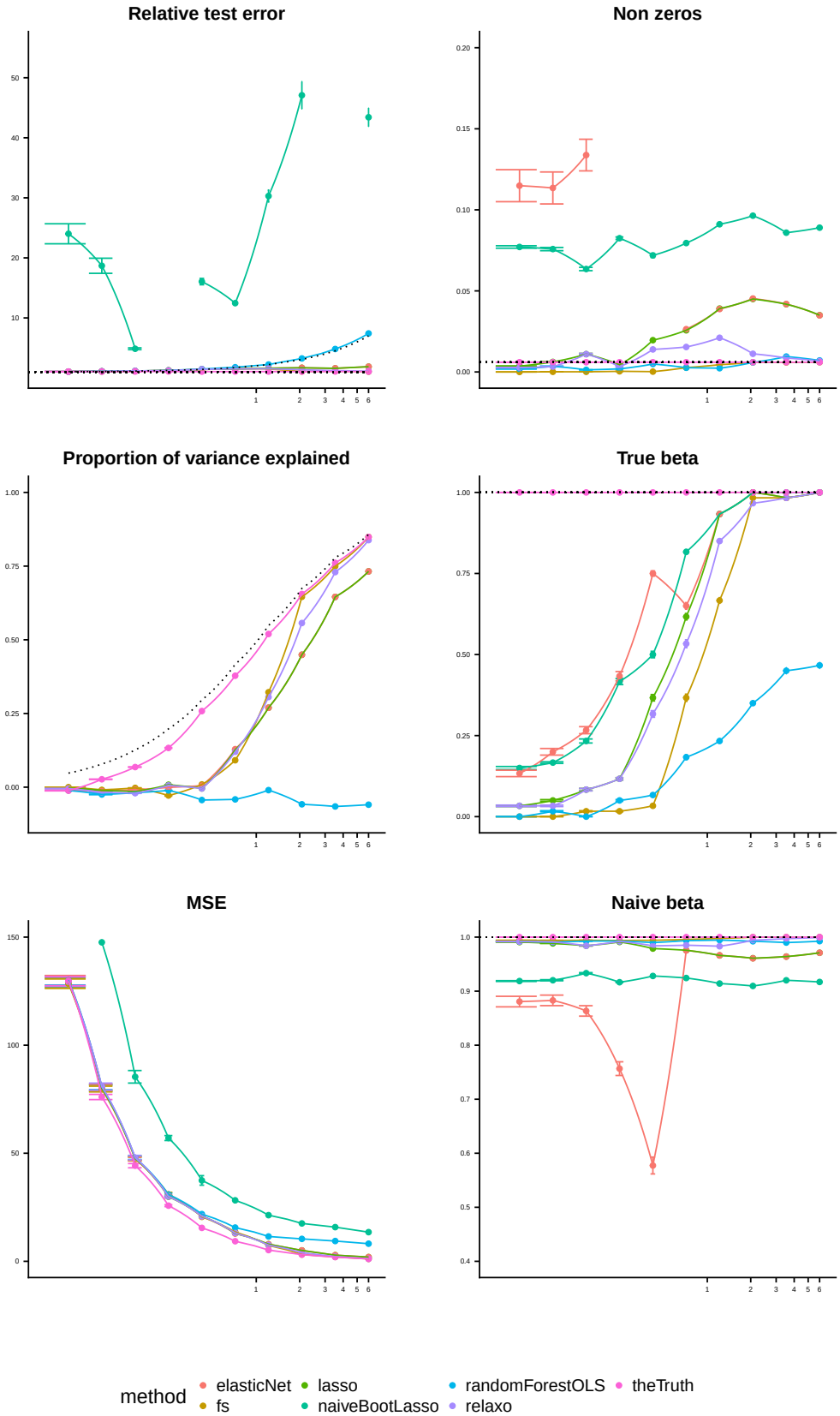
Medium setting

$\rho = 0.7$, $n = 1000$, $p = 100$ and $p_0 = 6$



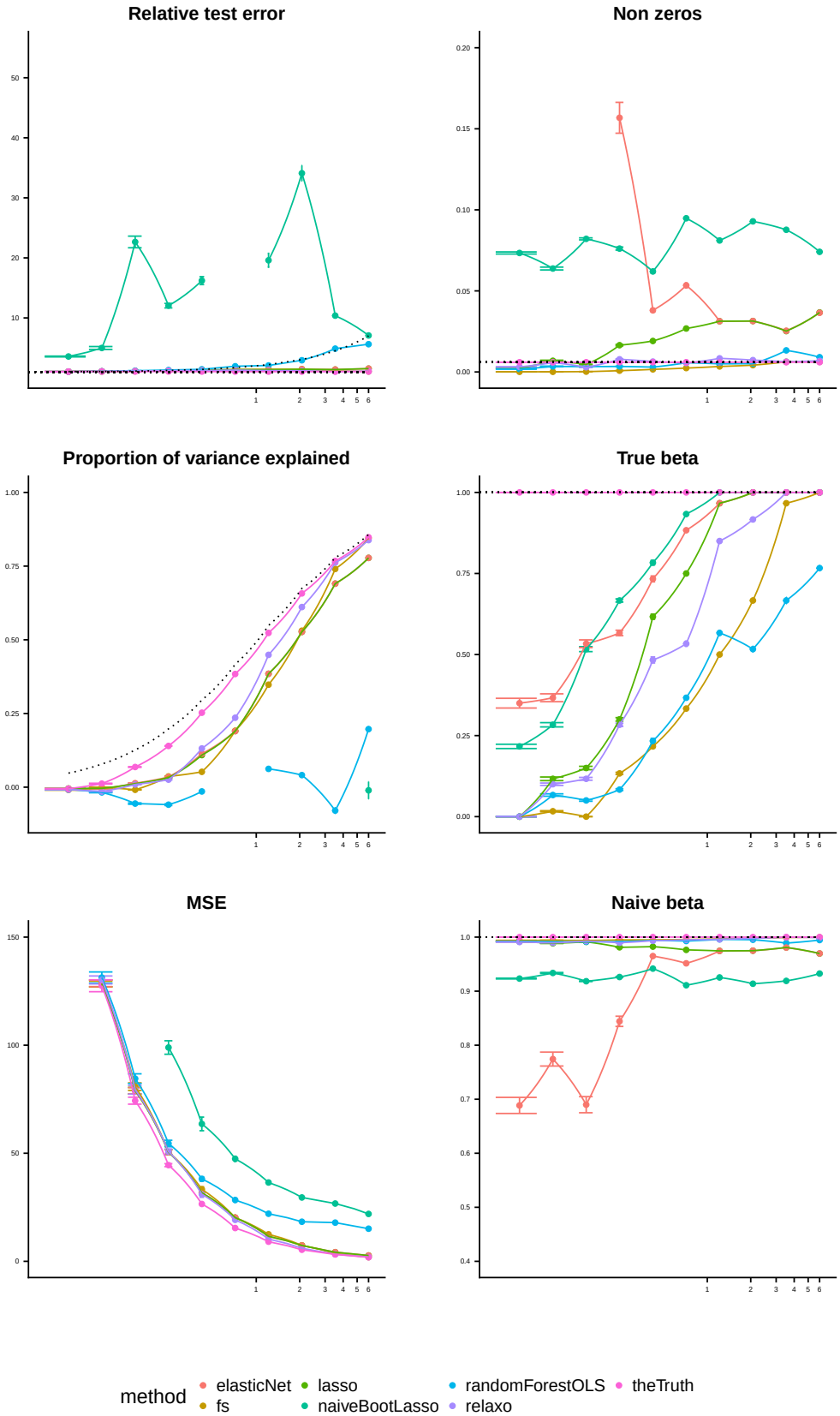
High setting

$\rho = 0$, $n = 100$, $p = 1000$ and $p_0 = 6$



High setting

$\rho = 0.35$, $n = 100$, $p = 1000$ and $p_0 = 6$



High setting

$\rho = 0.7$, $n = 100$, $p = 1000$ and $p_0 = 6$