Automatic variable selection algorithms

May 2021

Abdoul Aziz Berrada, Cécile Brissard, Morgane Caillosse, Hugo Hamon

Directed by: Philippe De Peretti

Programming in: SAS/IML

Sorbonne School of Economics, University, Paris, France

Abstract

When facing high dimensional data set, choosing the best subset of predictors is a real challenge. It is in this context that automatic variable selection algorithms were developed. Widely used, we could wonder if these algorithms perform well. In this paper, after presenting well-known Statistical Learning and Machine Learning algorithms, we test whether these algorithms perform well. To do so, we generate 3 different Data Generating Processes and test each algorithms with different information criteria. Finally, we define contexts under which algorithms perform the best.

Contents

1 Introduction

In econometrics and data science fields, the purpose is to have parsimonious models that can explain a lot without having too much information, i.e. variables. But how to select a set of relevant variables when facing a $LARS$ge number of variables? Indeed, high dimensional data set makes data more scattered. This tends to distort traditional ways of analyzing data. Moreover, as the need to classify data often corresponds a need to group individuals with similar properties, a higher dimensionality of data tends to increase dissimilarities between individuals. Hence, when facing a $LARS$ge number of variables, it gets pretty much impossible to select the best subset of predictors that are not redundant, not very explanatory, or non-significant information for the model.

To overcome this problem, automatic selection of variable is often used. Automatic variable selection is a process that aims to select a subset of variables considered relevant by the algorithm process. The input data of the variable selection process is the initial set of variables that forms the representation space and the set of training data of the studied issue. There are a $LARS$ge number of variable selection methods. These methods can be put in two main categories: Statistical Learning and Machine Learning. While Statistical Learning focuses more on the interpretability and description, Machine Learning aims, mainly, at predicting.

The aim of this paper is to describe and to test the reliability of various Statistical Learning and Machine Learning algorithms. To do so we will apply each selection process to our 3000 data sets of 100 observations each, previously built by 3 different Data Generating Processes (1000 databases for each DGP).

The purpose of choosing DGP is to know the exact structure of correlation between variables in each data set. This enables us to set up an evaluation procedure. Indeed, when performing variable selection algorithms on each data set, we will know which variables must have been selected and which one should not have been. In other words, this evaluation procedure aims at evaluating the performance and the relevance of the subset. Eventually, this level of knowledge on the data will allow us to define - in our opinion - the best model for variable selection.

The overall structure of this article has been divided in five sections. Among them, the models, the result's extraction method and the results themselves are divided into two sub-sections, opposing Statistical Learning and Machine Learning. Section 2 is a thorough description of the models which shall be used in the paper. Section 3 is focused on the methods applied on SAS software to simulate data and extract results. These results are described, for each model, in Section 4. Eventually, a discussion on the overall results of the paper will be made in Section 5.

2 Models

2.1 Statistical Learning

Since statistics is the mathematical study of data, studies can't be conducted without data. Statistical modelling is usually carried out according three objectives : description is the first one. It consists in defining the links between, multiple variables X_j and the target, which shall be described as precisely as possible. The second purpose of a statistical analysis is to explain the data through the correlation relationships ; those may allow us to define X_j as causalities for Y[5]. Here we can relate the determination of risk factors for prostate cancer based on clinical and demographic variables as an example. At last, the third objective, known as prediction, permits us to model the possibility that an individual from this same population will develop prostate cancer. However, Statistical learning, opposes itself to Machine learning, especially because its main purpose is the explanation and knowledge of a dataset, in giving a fairly weaker prediction power[2].

Both Statistical and Machine learning depend on data : Statistical Learning is formalized as a relationship between variables, whereas Machine Learning learns from data without explicitly programmed instructions.

According to PERFICIENT [6], a digital transformation consulting firm : "Statistical learning is mainly about inference, most of the idea is generated from the sample, population and hypothesis, compared to machine learning which focuses on predictions, supervised learning, unsupervised learning and semi-supervised learning."

In Économétrie, Modèles et Applications [11], B. Crépon et N. Jacquemet make a parallel between what inference is for statistics and what conclusions represents regarding population characteristics that can be derived from the observed sample. Inference consists in, as an example, asking ourselves what would be a variable mean in the population according to the mean of this same variable in a smaller sample.

2.1.1 Criteria for selecting variables

When attempting to explain or understand a phenomenon, it is preferable to define the most efficient model. Thus, it matters to choose the most explanatory model through information criteria comparison. We are now going to present the diversity of criteria which is at our disposal.

- **F test**
  An F-test is define as a statistic test which, under the null hypothesis/ normal error hypothesis, has an F-distribution : therefore it will follow in law $F(q, (N-p-1))$. It is mainly used for the comparison of statistical models adjusted on a data set, in order to identify the most suitable model for the initial population ; the one from whom the data were sampled. Mathematically, the F statistic measures the Residual Sum of Square (RSS) variation for each added parameter in the model. Let us consider q explanatory variables, X a matrix with p columns where $p = q + 1$, and $Y = X\beta + \epsilon$ such as :

$$RSS(\beta) = \sum_{i=0}^{N} (y_i - \hat{y_i})^2 = \sum_{i=0}^{N} (y_i - X\beta_i)^2 \tag{1}$$

  According to the previous equation F test formula is define as :

$$F = \frac{(RSS_0 - RSS_1)/q}{RSS_1/(N-p-1)} \tag{2}$$

- **Likelihood Ratio Test (LRT)**
  If we consider the maximum likelihood estimator of θ denoted $\hat{\theta}$, than under H_0 hypothesis the same estimator will be denoted $\hat{\theta_0}$ with $\theta \in \Theta_0$. Consequently, the LRT is given as :

$$\lambda = -2log(\frac{L(\hat{\theta_0})}{L(\hat{\theta})}) \tag{3}$$

  These q constraints under null hypothesis imply $\lambda(x_1, ..., x_p) \hookrightarrow X^2(q)$. In comparison the best model keeps being the one with the highest likelihood.

- **Akaike Information Criterion (AIC)**
  Whether, for the use of the Likelihood ratio test discussed above, nested models are needed, Akaike (1971-74) proposes the following generalized measure :

$$AIC = -2log(L(\hat{\beta})) + 2p \tag{4}$$

  with p explanatory variables.
  When it comes to comparisons, the better model is given by the one which minimizes the AIC.

- **Akaike Information Criterion (AICc)**
  The information score of the model (the lower-case 'c' indicates that the value has been calculated

from the AIC test corrected for small sample sizes). The smaller the AIC value, the better the model fit.

$$AICc = AIC + \frac{2k^2 + 2k}{n - k - 1} \tag{5}$$

- **Bayes Information Criterion (BIC)**
  This criterion penalizes more strictly complex models. Let N be available observations number, thus we shall have :

  $$BIC = -2log(L(\hat{\beta})) + p * log(N) \tag{6}$$

  As it can be seen in its definition, this criterion depends on N and the relative size of N and p. However, it penalizes more strongly the free parameters than AIC does. And, like for the BIC, the better model is the one which has the minimal AIC.

2.1.2 Variable selection algorithms in statistical learning

Nevertheless, when the number of explanatory variables is $LARSge$, it is not easy to explore their possible combinations (i.e. models) to compare the above information criteria. To circumvent this problem, automatic variable selection algorithms have been proposed, but their functioning and results are different. Let us describe these solutions :

– **Forward method** : Let us take a model with k explanatory variables. The $Forward$ method consists in testing k regressions with a one variable model, and choosing one-by-one the most significant variable according to the tests and criterion. On the first step, Y, the explained variable, is regressed on the most significant variable and then one explanatory variable is added at a time (among the rest of the variables) to the model (the one that best optimizes the information criterion such as minimizes the AIC and the BIC or maximizes the F-test and the LRT). It is when the variable is selected (and not deleted) that the algorithm goes further with the $k - 1$ regressions left, until there are no more significant variables to add that optimizes the criterion.

  As an example, if we have a model with five explanatory variables $X_n \forall n \in 1, ..., 5$, we will start the regression with the constant only, and add one by one each variable, regressing each time the obtained part-model. We will therefore, obtain five models, choosing the variable which contains the greatest explanatory power and thus, optimizes the chosen information criterion. Once we select a variable, we shall use the same method again with the four others, testing each one at a time. The process stops either when we opted for all variables, or when information criterion are not optimized anymore.

– **Backward method** : As it name indicates, this method opposes itself from the previous one. This time, the algorithm starts from the complete model. Once the most unsignificant (or the variable that minimize the less the criterion) variable is spotted, the algorithm follows the same pattern regressing the same model but with $k - 1$ variables. It is only when all variables are defined significant that the $Backward$ process stops (or when the criterion is minimized).

– **Stepwise method** : This method follows the same pattern as the $Forward$ algorithm, except two differences : first, it begin with only the constant, and mostly, it tests again all variables already existing in the model at each regression. The aim of this continuous updating is to avoid non-significant variables to appear : a previously significant one can be the otherwise when we add a new variable. It is a phenomenon that both $Forward$ and $Backward$ methods are not able to detect. It may be the case if there is a correlation between two variables. When this occurs, $Stepwise$ process selects the most significant variable and delete the other one, just like the $Backward$ method would do. The process stops when the algorithm can't add new significant variables. We will notice that $Stepwise$ method is not operating well with a big amount of explanatory variables, which is why, other algorithms such as $Stagewise$ are going to be introduced in the next part.

2.2 Machine Learning

The increase in dimension results in sparse data or, in other words, a lack of density. For instance, higher sets of data make harder to identify individuals with similarities because higher dimension tends to accentuate dissimilarities. As mentioned in the introduction, this led to distortion in the traditional way of analyzing data. This is especially true with statistical methods that required statistical significativity. In 1961, Richard Bellman called this phenomenon "the curse of dimensionality" in his book "Adaptive Control Processes: A Guided Tour" [12]. Especially, the curse of dimensionality tends to make harder to choose a small set of data from a database with multicolinear variables. There are many ways to handle the curse of dimensionality. One of them is to select a set of variables based penalized regressions. Penalized regression aims to find the parameter β that minimizes an objective function :

$$\hat{\beta} = argmin_{\beta}\{\frac{1}{n}\left\|Y - X\beta\right| + \lambda\left|\beta\right|\} \tag{7}$$

λ aim to find a compromise between the representativeness of the data and the complexity of the model. A general model can lead to a loss of information and a complex model can lead to bad prediction because of overfitting. The smaller is the λ, the more general is the model. Consequently, the higher is the λ, the more complex is the model. In this section, we present different approach for penalized regression.

2.2.1 Stagewise

Afterwards, let us detail the *Stagewise* method. As for the other algorithm it consists in a series of regressions: the first one is made using the X_k the more correlated to Y the explained variable to obtain the residuals, which we will consider as the new explained variable, now replacing the Y ; the second one will regress another again the most correlated variable on the new Y, and retrieve the residuals thus obtained. We cease these chain regressions when no more variables are correlated significantly to the residuals.

More precisely, at the beginning all coefficients are equal to 0 and at each iteration the coefficient associated with the variable most correlated to the residuals (i.e. new explained variable, Y) is increased slightly, at the size of the learning rate, ϵ. This learning rate is, at first, very low, but little by little, the value of the coefficient tends towards the value of the Ordinary Linear Square, until it is another variable which is the most correlated to the residuals. In fact, if we regress the residuals r_1 on a variable X_1, we obtain new residuals r_2, but the variable most correlated to these residuals r_2 is still the previous one X_1. Which is why we must continue to make regressions by increasing at each iteration the coefficient of X_1, until X_1 is no longer the variable most correlated to the residuals. The operation is rehearsed until there is no more correlation between the residuals and any variable. We thus have to :

1. Take residuals $r = y - \hat{y}$ avec β_1, β_2, ..., $\beta_j = 0$;
2. Find the variable x_i the most correlated to these residuals r ;
3. Update the coefficient β_i as follow : $\beta_i = \beta_i + \delta_i$ où $\delta_i = \epsilon * sign[corr(r, x_i)]$;
4. Update the residuals as $r = r - \delta_i * x_i$;
5. Restart from Step 2, until there is no more correlation between the residuals and any variable.

Usually this approach offers best results than the three others we have detailed above, whether it tooks more time. [essayer de trouver une ref pour prouver ça]

For each of these methods, we shall define a fit criterion for the model ; i.e. not only to have significant variables but to select variables that minimize some of the criterion described in the Statistical Learning part[1].

[1] see 2.1.1 Criteria for selecting variables.

2.2.2 Least-angle regressions (LARS)

Thus, when there are many variables, we shall consider a hyperplane. Model's coefficients are found through optimization, minimizing the sum squared residuals between $\hat{Y}$, the prediction, and Y the observed value, such as : $loss = \sum_{i=1}^{n}(y_i - \hat{y_i})^2$

Too many explanatory variables can however cause stability problems. To cope with this problem, it is possible to change the objective function by including costs so as to penalize models with many explanatory variables. Linear models using this modified objective function are called "penalized linear regression"[8]. There are several ways to include this penalty, here is the first one:

- The $l1$ penalty penalises the model using the sum of coefficients' absolute values. It allows to reduce as much as possible the coefficients values, until some of them become null, which evicts associated explanatory variables :

$$\begin{cases} min \sum_{i=1}^{n}(y_i - \hat{y_i})^2 \\ st \sum_{j=1}^{J}|\beta_j| \leq s \end{cases} \tag{8}$$

Definition
Introduced in 2004 by Efron and al. [4], least angle regression is an algorithm used for regressions on high dimensional data to select the variables that are correlated to the explained one. *LARS* approach is similar to a *Forward Stepwise* regression. In fact, it is linked to *Stagewise* and *LASSO* algorithms, such that it is possible to switch from one to the other. It includes a penalty on $l1$'s standard, however unlike *LASSO*, *LARS* does not have hyperparameters : "the weighting is discovered automatically by *LARS*".

Algorithm
This algorithm requires as many steps as there are variables to select. At its beginning, the associated coefficients with all the explanatory variables are null.

1. Normalize the data so that they have a mean of 0 and a variance of 1 ;
2. We start with all the coefficients $b_j = 0 \quad \forall j$;
3. We find the explanatory variable x_i the most correlated with y ; we increase (in the form of a percentage) the number of coefficients ;
4. We increase (by iterating) the coefficient b_i associated with x_i until a new variable x_h has a strong correlation with the current residual: $r = y_y$ (residual of the regression of Y on x_i) ;
5. The coefficients associated with x_i and x_h are increased in an equi-angular direction until another variable x_k is correlated with the residuals ;
6. We increase the coefficients associated with i, h and k in the equi-angular direction, this is the "least angle direction" ;
7. The loop continues until all the coefficients are in the model.

In attendance of noise in the dependent variable and of multicolinearity in the explanatory variables, there is no guarantee that the variables selected by the model are the true causal variables.

From *LARS* to *LASSO*
As we have implied above, *Forward Stagewise* regression and *LASSO* can be considered as restricted versions of *LARS*:

- The difference between *LAR* and *LASSO* appears when a coefficient crosses the value 0. Indeed, when it happens in *LASSO*, the explanatory variables for which the coefficient has taken 0 for value is dropped and the direction recalculated.

⩾ For the *Forward Stagewise*, we start with *LARS*. However, when the sign of the coefficient b_j opposes itself to the direction of $corr(, x_j)$, we project the direction in the positive cone (the direction of $corr(, x_j)$) and we use this new projected direction to carry on the algorithm.

2.2.3 Ridge, LASSO & ElasticNet

These methods should be used in precise cases, such when there is overfitting ($n < p$; n = sample & p = number of parameters) : the variance is too important, the prediction of a variable compared to another is too sensitive to the variations of this same variable. The goal is to add a bias ($epsilon \neq 0$) in order to decrease the variance. They should be used too when the goal is to select the most interesting variables for the prediction, by minimizing or suppressing the others.

Ridge ($l2$ norm)
For linear models, overfitting generally leads to solutions that are too complex, taking into account noise or spurious correlations within the predictors. Regularization aims to overcome this phenomenon by constraining, either by biasing or reducing, the capacity of the learning algorithm to promote simple solutions. Regularization penalizes "large" solutions by forcing the coefficients to be small, i.e., by reducing them to zero.
The function $J(w)$ is to be minimized with respect to w and is composed of a loss function $L(w)$ which serves the goodness of fit, and a penalty term denoted $\omega(w)$ which serves the regularization to avoid overfitting. $J(w)$ is thus a compromise where the respective contribution of the loss and penalty terms is controlled by the regularization parameter λ. Below is the representation of the combination of the loss function $L(w)$ and the penalty function $\omega(w)$:

$$J(w) = L(w) + \lambda\omega(w) \tag{9}$$

The respective contribution of the loss and the penalty is controlled by the regularization parameter, denoted λ. This penalty on the coefficients is coupled to the $l2$ norm: it penalizes with the **Euclidean norm** of these coefficients while minimizing the losses.

$$Ridge(w) = \sum_{i}^{N} \left(y_i - x_w^T\right)^2 + \lambda\|w\|_2^2 \ = \|y - x_w\|_2^2 + \lambda\|w\|_2^2 \tag{10}$$

→ The more *lambda* increases the more it reduces the w coefficients towards 0.

→ This approach penalizes the objective function by the Euclidean norm of the coefficients. The solutions with *LARS*ge coefficients thus become not ideal

→ *Ridge* allows to reduce the impact of some variables by reducing their coefficients but does not allow to remove the "bad" variables!

Least Absolute Shrinkage and Selection Operator ($l1$ norm)
As for *Ridge*, the ability of *LASSO*[14] to select a subset of variables is due to the nature of the λ constraint on the coefficients. The only difference is the application of the **norm** $l1$, unlike *Ridge* where $l2$ was applied. This constraint will bias the capacity of the learning algorithm. *LASSO* allows to remove the parameters linked to the strongly correlated variables if necessary, and to keep only one variable among those which are correlated between them.
The *LASSO* function is therefore[7] :

$\text{LASSO}(w) = \|y - X_w\|_2^2 + \lambda\|w\|_1$
(11)where λ is :

$$\lambda = 2\sqrt{2\sigma^2 \frac{\ln\left(\frac{2d}{\nu}\right)}{n}} \tag{12}$$

However, we choose λ, by trial and error, or using the **Cross-Validation**, we ought to minimize the variance ; the ideal method is the $10 - fold\ cross - validation$.

ElasticNet
ElasticNet Regression[3] allows to combine $LASSO$ and $Ridge$, and is therefore useful when the number of variables is so $LARS$ge that we don't know their usefulness on the explained variable, nor if they are correlated with each other, etc. In this case, where we cannot choose between $Ridge$, reducing the impact of certain variables, and $LASSO$, suppressing auto-correlation, we do both : we have to use a combination of $LASSO$ and $Ridge$. We then have two parameters λ :

- λ_1 for the $LASSO$ regression,
- λ_2 for the $Ridge$ regression,

These two minimizations are added to the sum squares of the residuals (basic OLS) according to the following formula:

$$E_{Net}(w) = Ridge(w) +$$

$\text{LASSO}(w) = \|y - X_w^T\|_2^2 + \lambda_1\|w\|_1 + \lambda_2\|w\|_2^2 = \|y - X_w^T\|_2^2 + \alpha\left(\rho\|w\|_1 + (1-\rho)\|w\|_2^2\right)$
(13)where α is a global penalty and ρ assigns the importance of $LASSO$ over $Ridge$. For example, for a pure $Ridge$ we have $\rho = 0$, while for a pure $LASSO$ we have $\rho = 1$.

3 Method

This paper aims to test the performance of the variable selection algorithm presented in the previous Section. To do so, data sets with a known correlation structure were needed. Indeed, in order to know whether an algorithm performs well or not, we need to make sure that the data we are working on follows a precise correlation structure. Thus, we simulate 3000 databases from 3 Data generating processes. Each database contains a set of 31 variables (1 explained variable and 30 explanatory variables) and 100 observations. Then, we test each algorithm on all 3000 data sets and conclude on their performance.

3.1 Simulating correlated variables with multinormal distribution

For each Data Generating Process, we generate 1000 samples of size $N = 100$ from a multivariate Normal distribution. That is to say that each vector of a sample follows a univariate normal distribution and the relationship between two vectors of a sample is determined by a specified variance covariance matrix.

3.2 Toeplitz matrix

Simulating a specific correlation structure obviously implies creating a correlation matrix. Correlation matrices are symmetric positive semi-definite. To create matrices that fulfills these properties we work with Toeplitz matrices. Toeplitz matrices have an interesting patterns. Indeed, it is a a diagonal-constant matrix, meaning that the entries in the matrix depend only on the differences of the indices, if symmetric, is has all properties of a correlation matrix. Let us consider $X = (X_1, X_2, ..., X_K)$, then the Toeplitz matrix, in the symmetric case, is such that :

$$\begin{pmatrix} X_1 & X_2 & X_3 & \cdots & \cdots & \cdots & \cdots & X_K \\ X_2 & X_1 & X_2 & X_3 & & & & \vdots \\ X_3 & X_2 & X_1 & X_2 & \ddots & & & \vdots \\ \vdots & X_3 & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & X_3 & \vdots \\ \vdots & & & \ddots & X_2 & X_1 & X_2 & X_3 \\ \vdots & & & & X_3 & X_2 & X_1 & X_2 \\ X_K & \cdots & \cdots & \cdots & \cdots & X_3 & X_2 & X_1 \end{pmatrix} \tag{14}$$

3.3 Level of correlations

In absolute value, we consider that:

- Two highly correlated variables have a correlation coefficient between 1 and 0.6;
- Two moderately correlated variables have a correlation coefficient between 0.6 and 0.3;
- Two weakly correlated variables have a correlation coefficient between 0.3 and 0.

We consider two differents correlation matrices for the first and the second data generating processes.

3.4 Data Generating Processes (DGP)

3.4.1 First Data Generating Process (DGP1)

Using Toeplitz matrix, we obtain the following correlation matrix :

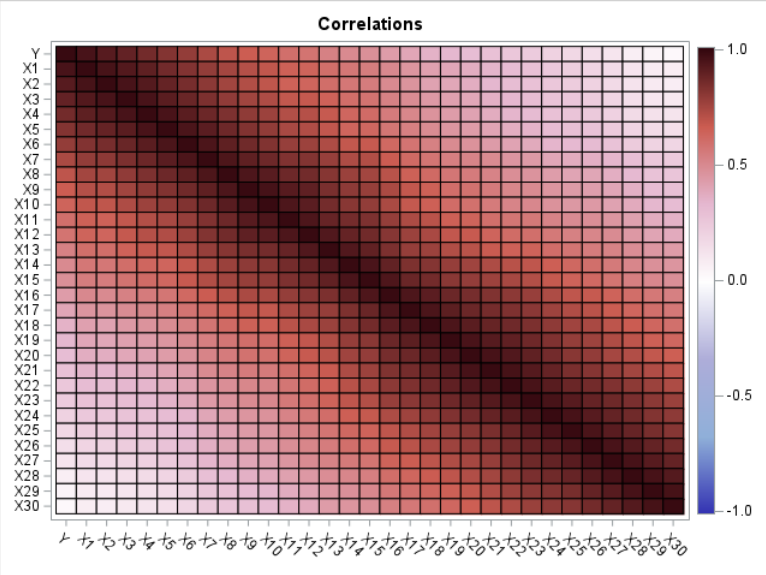


Figure 1: Correlation matrix

Each variable is highly correlated with the closest variables in the matrix. The further away a variable is from another, the less correlated it is. For instance, let us consider the variable Y. Y is highly correlated with X_1 to X_{10}, it is moderately correlated with X_{10} to X_{20} and weakly correlated with X_{20} to X_{30}. See the matrix in the Annex for more details.

3.4.2 Second Data Generating Process (DGP2)

After having generated the first data set using the Toeplitz matrix, we analyze the variable selection results given by each statistical learning algorithm. The extraction of the results gives us a visual account of the number of times each explanatory variable is selected, according to the processes and chosen criteria. Globally, and as explained in the descriptive part of the bar charts [2] below, the different models have been taken in default by our data set. This is explained by the very high correlation of the first variable with Y and therefore by the fact that this single variable is the main source of information for the model. Variables that are farther from X_1 (i.e. the least correlated with it), are then complementary sources of information, selected in spite of the fact that they are not the most relevant to the explanation of Y.

In order to continue our presentation towards Statistical and Machine Learning selection, we decided to generate new data with the same process (ie. Toeplitz Matrix and Multivariate Normal Distribution). However, the correlation coefficients will be chosen this time so as not to give as much importance to the first ten variables. Thus, we obtain a heat map following the same reading as the first one. More precisely, the correlation of X_1 to Y is 0.7. It follows that all the ensuing correlations will be lower than this, which gives us :

- ⩾ Ten first variables have a correlation coefficient to Y between 0.7 and 0.5;
- ⩾ The ten followings have a correlation coefficient to Y between 0.4 and 0.2;
- ⩾ The last ten have a correlation coefficient to Y between 0.1 and 0.001;

[2] see 4.1.1 Results, Statistical Learning, First Data Generating Process

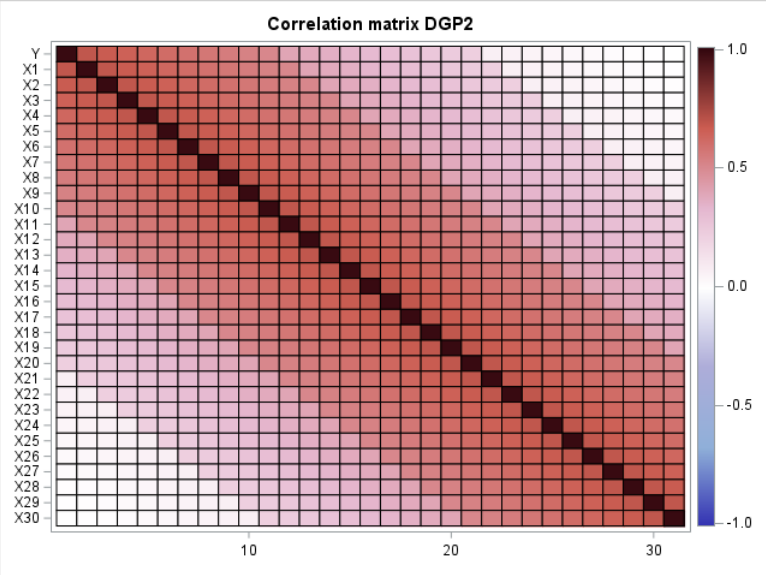


Figure 2: Correlation matrix

3.4.3 Third Data Generating Process (DGP3)

Finally, we decided to generate a last data set to ensure how results are consistent. To generate our third data generating process, we :

- Simulated 1000 data sets of 30 variables using Multivariate Normal Distribution and used the identity matrix as a variance covariance matrix;
- Generated Y according to the following equation:

$$Y = 0.6777 \times X_1 + 0.414 \times X_2 - 0.5814 \times X_3 \tag{15}$$

Here, the point was to have significant correlations between very few variables and Y, and random correlation with X_4 to X_{30} and Y. There were also non significant random correlations between the covariables.

3.5 Results' extraction

3.5.1 Statistical Learning

Once we obtained our data sets from each DGP, we shall see which variable was selected for each model. Selected variables have a correlation coefficient assigned. By extension, the variables with no coefficient assigned, appearing as a missing value in the coefficient results, are those that have been removed, the "unselected" ones.

In order to compare the diverse selection's processes, we need to obtain, not the correlation coefficients as a single result, but the frequency of selection of each variable according to the different data sets. Consequently, we choose to replace each coefficient by 1 and each missing value with a 0. The matrix we thus obtain will be employed to establish a bar chart giving the frequency of each variable that has been chosen. We shall have for each process we stated : *LARS*, LASSO, ElasticNet, *Stepwise*, *Backward*, *Forward*, a bar chart representing the number of times a variable is picked among the 1000 regressions. In the following section, these results will allow us to make a comparison between the different variable selection's methods exposed in the models[3].

[3]See the section 2. Models

However, with the obtained bar charts, some information remains missing such as the probability of selection of this or that variable, and even more the joint probabilities that several given variables are to be selected at the same time.

In order to obtain these probabilities, we build a new database composed of 0 and 1, which assigns 1 when two, three or more variables are selected in the same data set; 0 if at least one variable is not selected with the others. For instance, we define backward12, as the joint selection of X_1 and X_2 with the *Backward* method : if this condition is met, the value 1 is written in the column, otherwise 0. Once this has been done, we only have to count the number of 1 among the 1000 bases to have a probability between 0 and 1, i.e. from 0/1000 if X_1 and X_2 are never jointly selected, to 1000/1000, if they are attached each time.

3.5.2 Machine Learning

Cross validation : k-folds
Penalized regression method can lead to both simple and complex models hence, working with selection methods such as *LASSO*, *LARS* or *ElasticNet* implies choosing an optimal value of the regularization parameter λ of the *LASSO* or *LARS* estimator. To do so, we used the K-fold method.

K-fold cross validation aims to split a data set into K disjointed subsets where 1 subset is used for validation and the model is performed on the $K-1$ other subsets. Then the validation subset is used to compute the prediction error. This has to be done for all K subsets in order to have K estimations of the prediction error. This method enable us to choose the best λ parameters for each models, that is, the λ that minimizes the prediction error.

Stopping criterion : PRESS
Here we used cross validation as stopping criterion. As developed in SAS documentation [10], this means that at step k of the selection process, the algorithm determines the best variable to enter or leave the model. Then, the PRESS score is computed. If the PRESS score for this model is higher than the score for the model at step k, the selection process terminates at step k.

The PRESS score is defined as follow :

$$\text{PRESS} = \sum_{i=1}^{n} (y_i - \hat{y}_{i,-i})^2 \tag{16}$$

4 Results

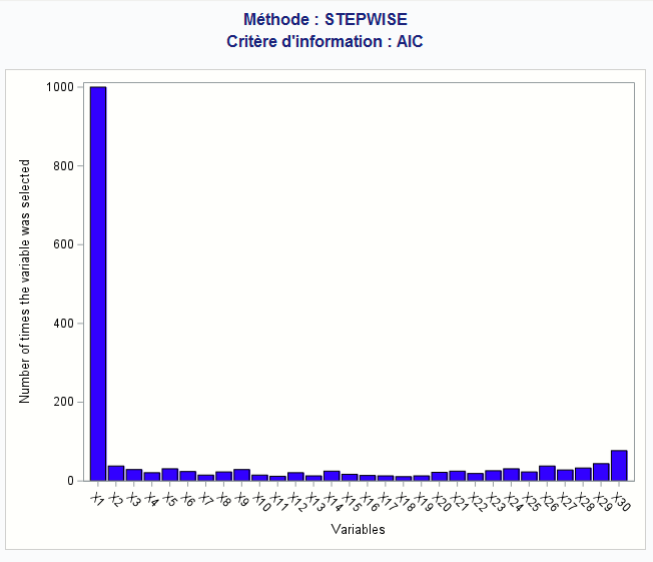
4.1 Statistical Learning

4.1.1 First Data Generating Process

Now that we have the graphs associated with the probability of a variable to be selected by each of the models; as well as those of the joint probabilities that several variables are selected at the same time, let us describe those in order to compare and evaluate the accuracy of each model.

Each table of graphs represents a model among *Forward*, *Backward* and *Stepwise*. In each of them there are four graphs representative of the four chosen selection criteria. At the top left, the bar chart without criteria is used as a control to account for the discrepancies between the criteria: at the top right, the BIC, at the bottom left the AIC and at the bottom right the AICC. This configuration will be applied in this part only for the graphics of the *Backward* process. In other words, since the graphs provide the same information regardless of the chosen criteria for the *Forward* and *Stepwise* models, we will present only one instead of eight.

Individual probabilities for variables of being selected by the model
This section's purpose is to show the bar chart, compliantly to their construction, described above in the Results' extraction's Method. The main result of the graphs is, that the three models have been taken in defect by the chosen correlation matrix. Indeed, all models taken together, the variable X_1 is constantly chosen: its individual probability of selection is therefore 1, or 100%.



Thus, we see that the results are almost the same for all models, and under all criteria. It should also be noted that the variable X_{30}, although constructed as the least correlated to the explained variable via the use of the Toeplitz matrix, is significant at almost 10%. Though, we ought to think that by choosing X_{30} the models are seeking information not already contained in X_1 : this effect is particularly valid with X_{30}. Indeed, there is no inter-variable correlation between the first and the last X.

This conclusion allows us to affirm that, when faced with a variable (X_1) that is too correlated to the explained variable, the models are defaulted. Indeed X_1 is 95% correlated to Y. In a "normal" setting, i.e. with a lower correlation of X_1 (e.g. 0.7), the results will be different as we will see with the description of the results of the second generating process below.

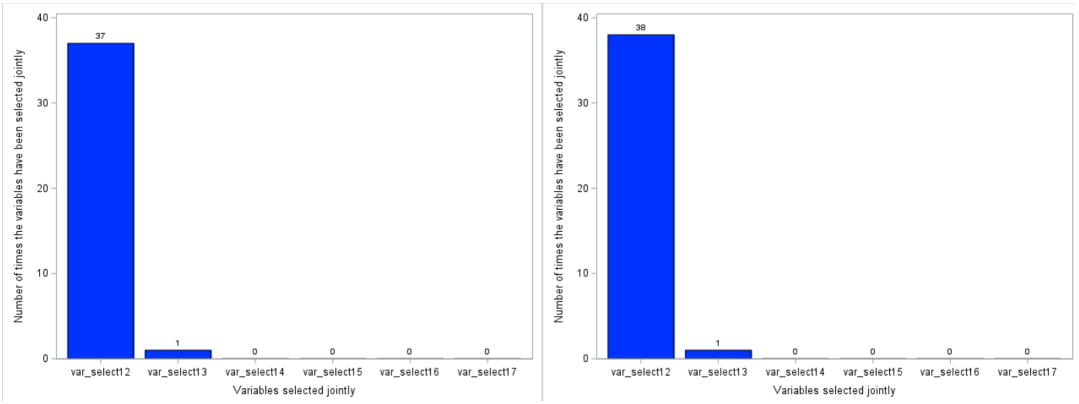
If the *Forward* and *Stepwise* processes present similar graphs, we notice that the *Backward* model stands out by its selection. Indeed, the variable X_1 is always selected at 100%, so every time on the 1000 simulated databases, but the other variables are selected more often than it was the case with the two other models. Though, they are selected almost all at the same frequency, the variables X_2 to X_{30} are selected between 1 and 2 times out of 10 with no criterion applied, about 1 time out of 5 with the BIC, and the AICC, and at least in 20% of the cases with the AIC. More precisely, we notice that the slope of this distribution is slightly increasing. This is a characteristic of the graphs that should alert us to the irrelevance of the models - in the case of an overly correlated X_1.

Indeed, the other variables are then totally forgotten by the model, and it is the most explanatory ones that are less often selected in favor of those being less correlated to the first one. For instance, it is the 29^{th} variables that is the most selected afterX_1 to nearly 30% with AIC, and more or less 20% without criteria or with AICC and BIC choices. It is remarkable that the AIC is therefore less selective than the other criteria.

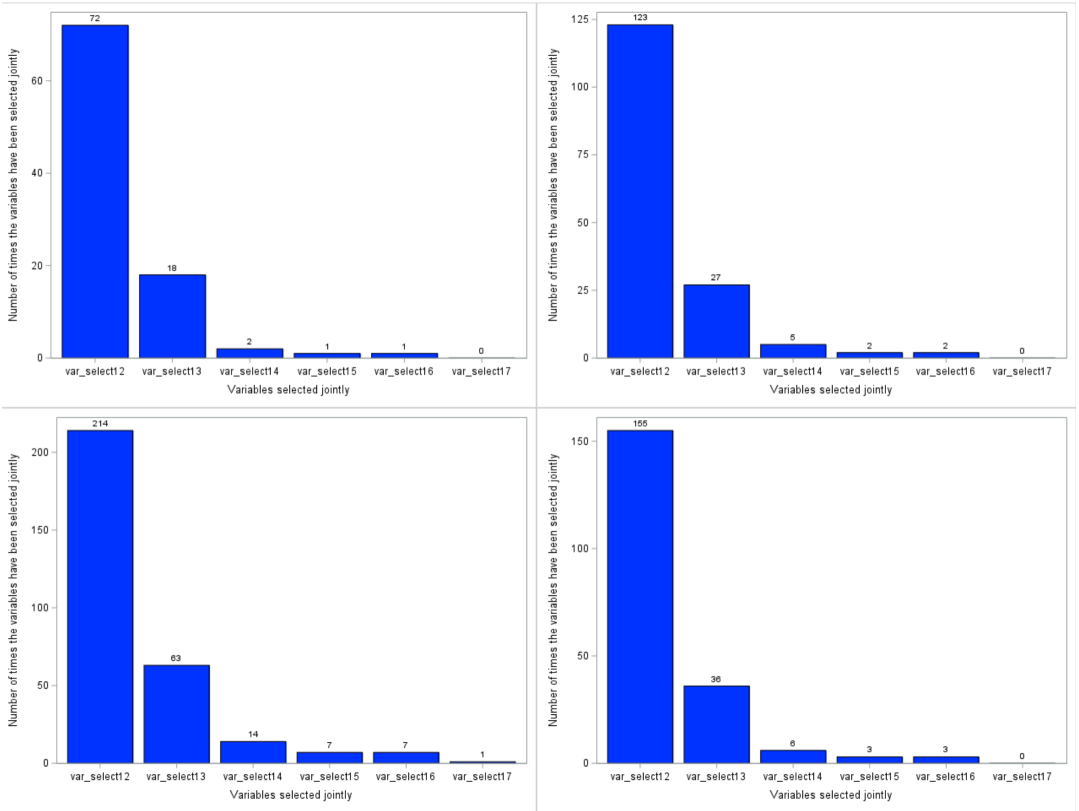
In any case, with variables that are less and less correlated to Y, it is expected that the distribution will be decreasing: the more important the correlation is, the more significant the variable must be. But here, X_1 concentrating all the explanatory power of the model, the frequency of the other variables' selection is biased. What about the joint selection in this framework? Let us explain the joint probabilities bellow.

Joint probabilities for variables of being selected by the models

Again, the main result is that the joint probability of selection of the first two variables is $\frac{38}{1000}$ with the *Forward* and *Stepwise* processes (graph on the right), no matter of the criterion (except $\frac{37}{1000}$ for the *Stepwise* without criterion on the left graph). We shall see that, this leaves no room for the joint selection of the other variables, where X_1, X_2 and X_3 are picked together in only 0.001% of the cases.



Alike the simple probabilities, the *Backward* model differs from the two others in terms of the joint probabilities of the explanatory variables' selection, as we observe below :



Thus, even in the particular setting discussed here, it appears to be the most effective among those models. Not only do the first two variables not always have the same probability of being chosen jointly, but

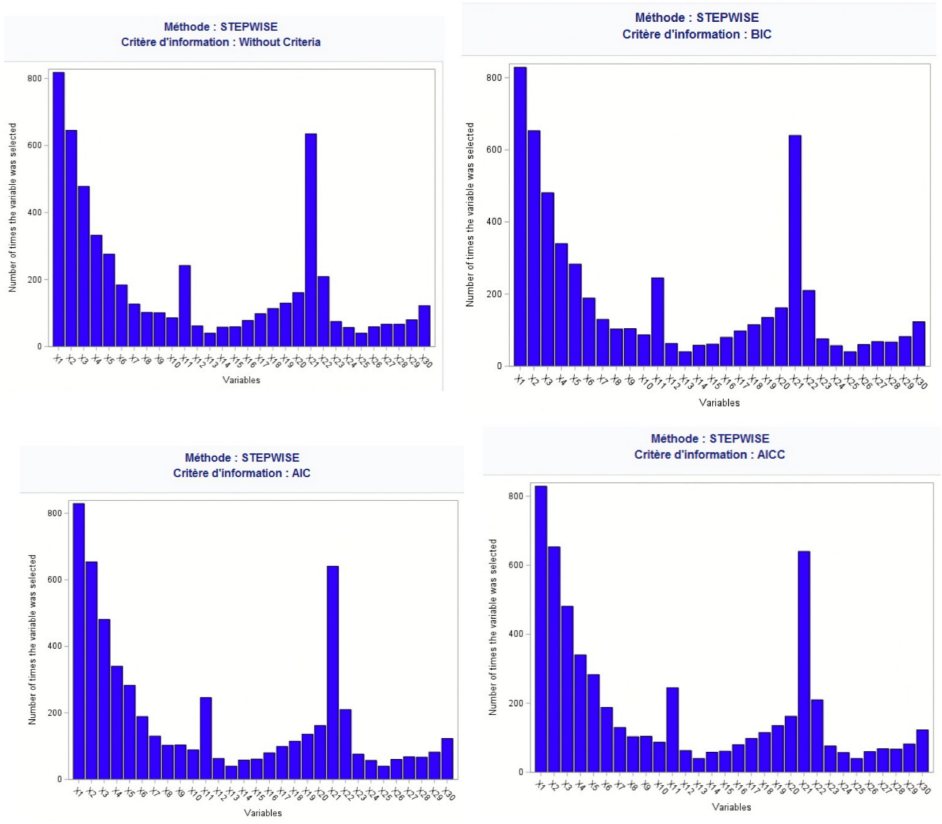
the *Backward*'s bar charts show us here the normally expected decrease in frequency. These graphs also allow us to see the relevance of the AICC (bottom right) and BIC (top right) criteria, compared to the AIC (bottom left). The latter reflects a less restrictive selection where the first two variables are jointly picked in 21.4% of cases, the first three at 6.3%, the fours at 1.4% and the fives and six at 0.7%.

In view of these results, we are obliged to generate a second matrix - again with decreasing correlation coefficients - but starting from a less important coefficient for X_1 in order to allow the selection models not to be biased. We analyze these results in the following section.

4.1.2 Second Data Generating Process

In light of the results provided by the first data generating process, we chose to work with a second correlation matrix. Unlike the first one, this second correlation matrix allows to generate variables less correlated with the dependant variable. Here, the variable which is the most correlated with the dependant variable has a 0.7 correlation coefficient. Applying the three algorithms to the 1000 data sets created by this second DGP, we obtained very different results than those we obtained with the first one.

Individual probabilities for variables of being selected by the model
Here, as opposed to the first data generating process, the highest correlated variable X_1 is not always chosen by either of the three algorithms. Moreover, this time, we can see differences in the choice of variable for the three algorithms.

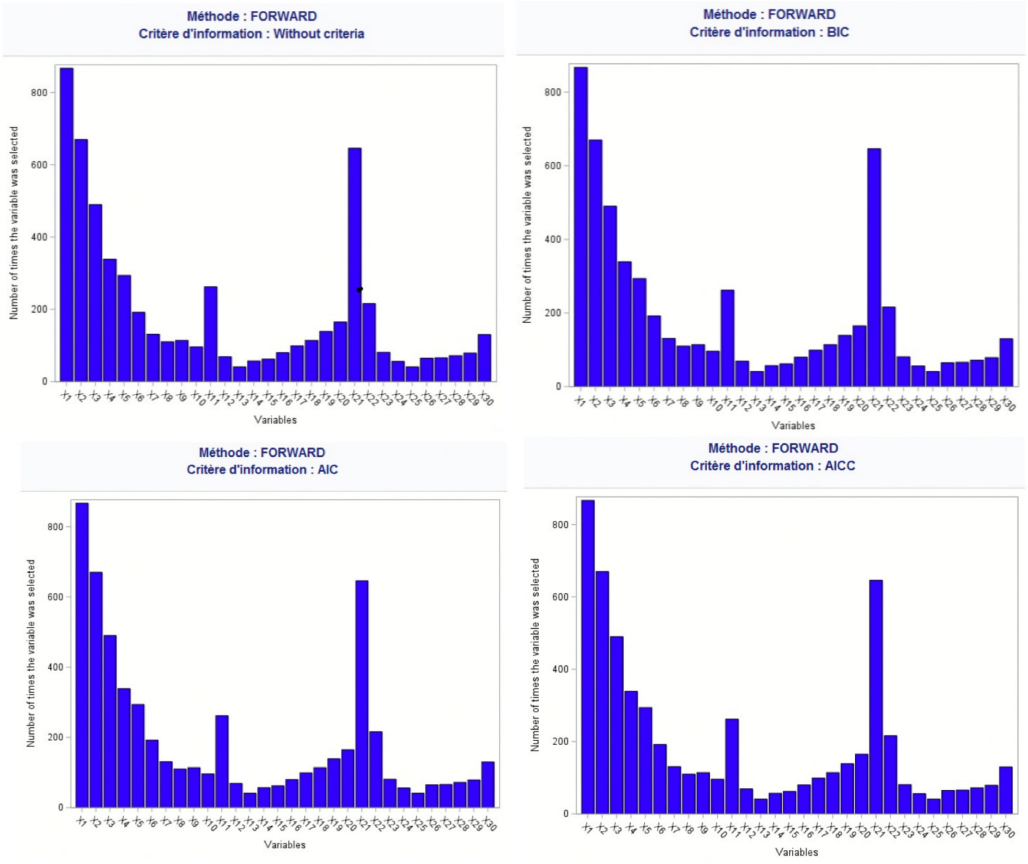


In the case of *Stepwise* regression, we found that, whatever the information criterion is, X_1 has a 80% probability of being chosen by the model, considering that the correlation coefficient between Y and X_1 is 0.7, this is a rather good result. X_2, which is slightly less correlated to Y then X_1 has, for each information criteria, a 60% probability of being chosen by *Stepwise* algorithm.

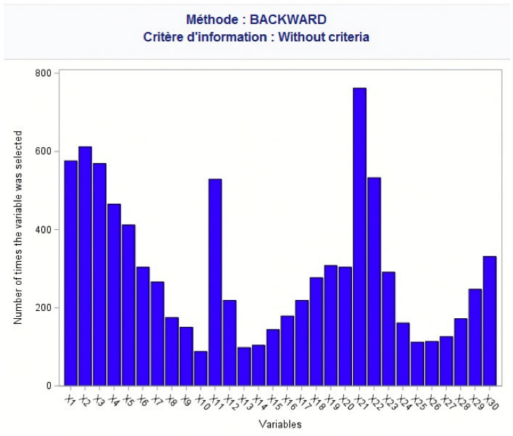
For variables X_3 to X_{10}, the probability of being chosen by *Stepwise* algorithm decreases so do the correlation coefficients between these variables and Y. However, X_{11} which has 41% of common variation with Y, has higher chances to be selected by the algorithm (30% probability) than X_5 to X_{10}. The same pattern happens for X_{13} to X_{21}. Indeed, has the correlation coefficient goes lower and lower between these variables and Y, their probabilities of being selected increases. This could be explained by inner correlations between independent variables.

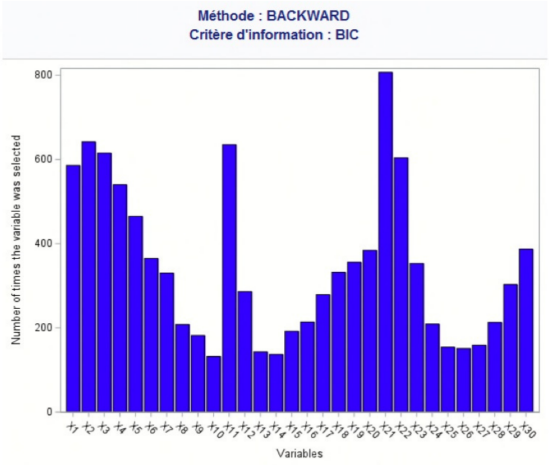
For instance, the first tenth variables have high level of correlation between them. But those variables share low level of correlation with X_{11} to X_{21}. As a consequence, while X_{21} shares only 8% of variation with Y, is has a 60% probability of being selected by the model which is more than the probability of X_2 of being chosen. But X_{21}, on the contrary to X_2, has a low level of correlation with X_1 meaning that X_{21} could bring information that X_2 does not.

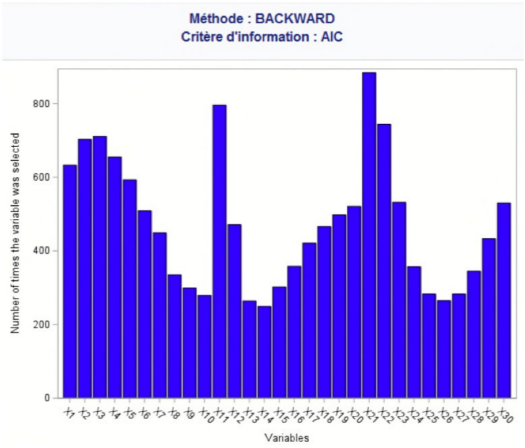
As you can tell by the figure below, the *Forward* algorithm shows exactly the same results, in term of individual probability, as the *Stepwise* algorithm.

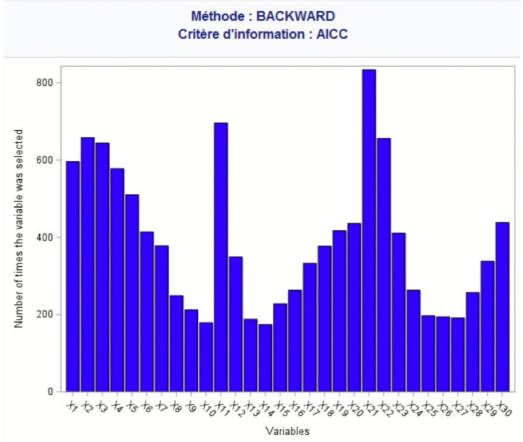


However, *Backward* algorithm does show really different results then *Forward* and *Stepwise*. Indeed, without any information criteria, X_1 which shares 70% of variation with Y has less then 60% probability of being chosen by the model. On the opposite, X_{21}, which shares 9% of Y's variations, has a 75% probability of being selected. The results are slightly better when choosing AIC as information criterion but do not get better neither with BIC nor AICC.


Méthode : BACKWARD
Critère d'information : Without criteria
Number of times the variable was selected
Variables


Méthode : BACKWARD
Critère d'information : BIC
Number of times the variable was selected
Variables

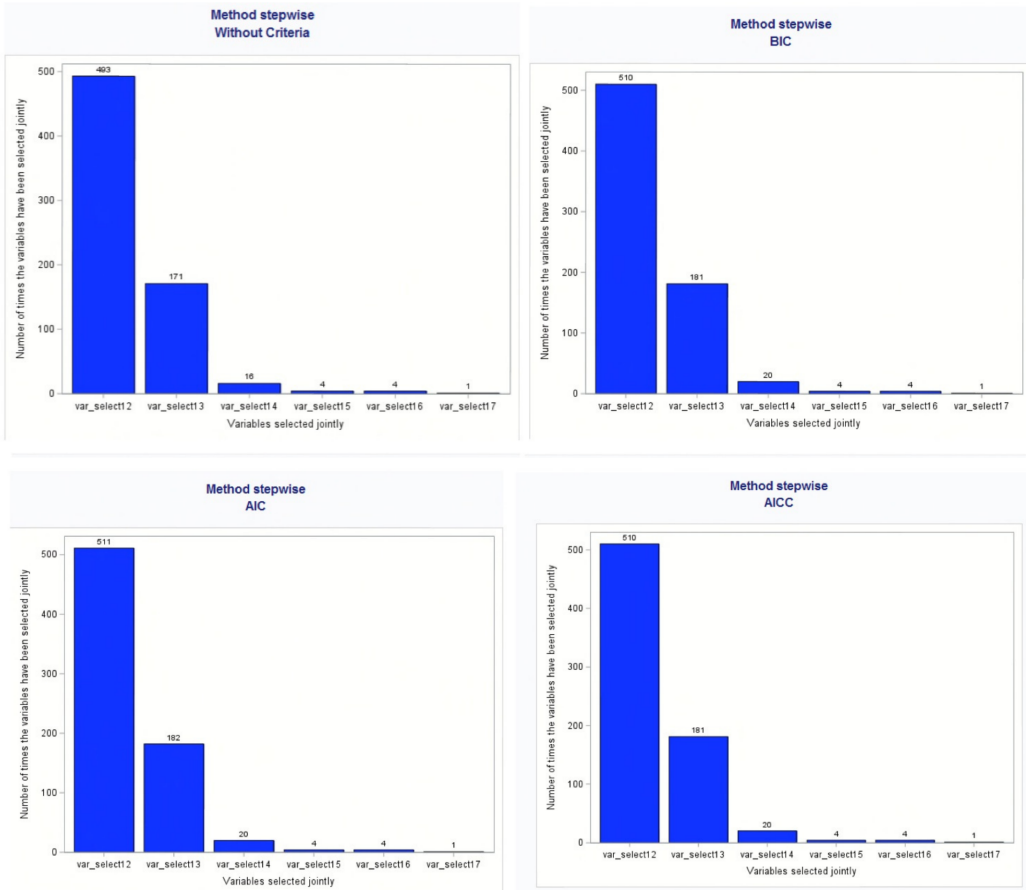

Méthode : BACKWARD
Critère d'information : AIC
Number of times the variable was selected
Variables


Méthode : BACKWARD
Critère d'information : AICC
Number of times the variable was selected
Variables

Joint probabilities for variables of being selected by the model

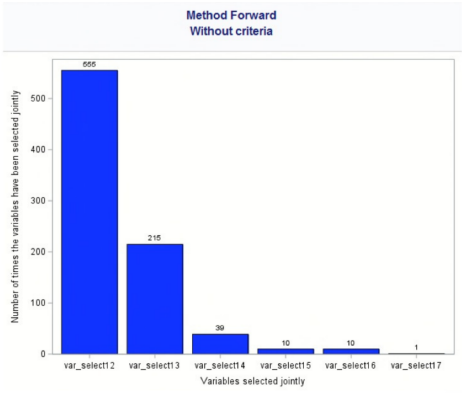
While individual probabilities of being selected do not vary much among algorithms nor information criteria, joint probabilities for a group of variables of being selected are really different, not only from one algorithm to another but also from one information criterion to another.

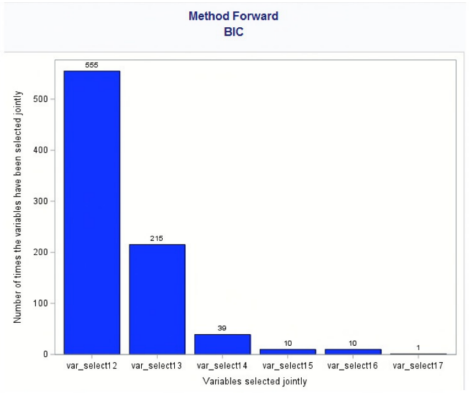
Starting with the *Stepwise* algorithm without information criteria, the probability for X_1 and X_2 of being selected jointly is 49.3%. The variables X_1 to X_3 has 17.1% chances of being selected jointly. The joint probability converges toward 0 when adding X_4 to X_7. In the case of *Stepwise* algorithm, the criteria which performs the best is AIC, however, BIC anc AICC exhibits pretty much the same results as AIC. Indeed, with AIC as an information criteria, X_1 and X_2 have 51.1% probability of being chosen jointly. With BIC and AICC, the same probability is 51%. Once again, when adding other variable such as X_3, the probability decreases until converging toward 0.

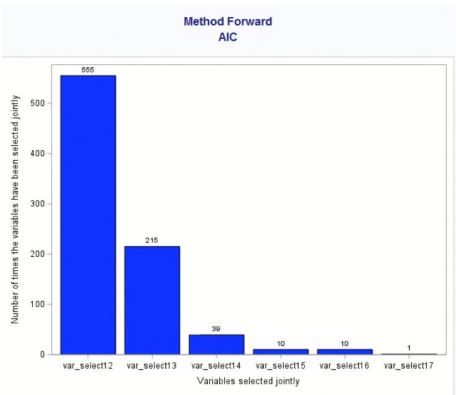


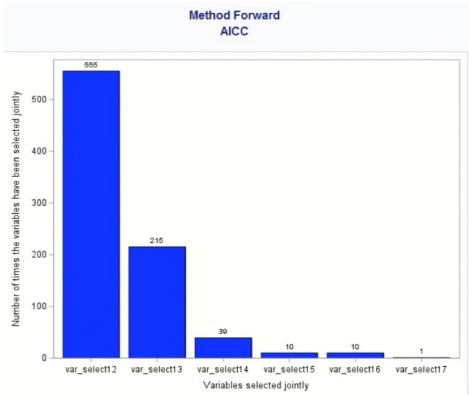
The graphic below highlights that *Forward* algorithm performs better than *Stepwise*. Here, the results are the same for every information criteria. X_1 and X_2 have 55.5% chances of being selected jointly. X_1, X_2 and X_3 have 21.5% chances of being selected jointly, that is 4 percentage points more than *Stepwise*.

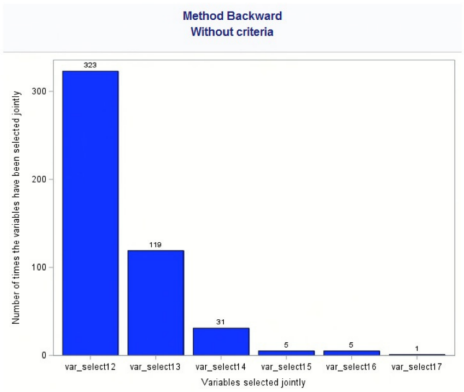
The joint probability for X_1, X_2, X_3 and X_4 of being chosen is 0.39%, while this is not a good result, this still is more than two times the joint probability for X_1 to X_4 of being chosen by the *Stepwise* algorithm.

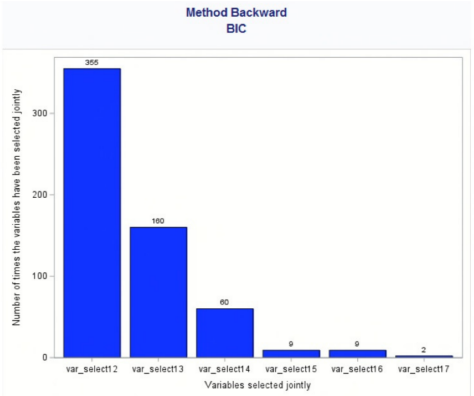

Method Forward
Without criteria
Number of times the variables have been selected jointly
555
215
39
10
10
1
var_select12
var_select13
var_select14
var_select15
var_select16
var_select17
Variables selected jointly

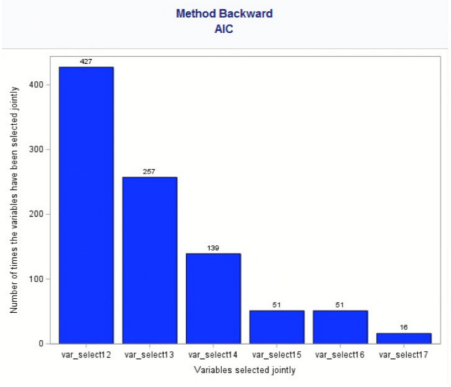

Method Forward
BIC
Number of times the variables have been selected jointly
555
215
39
10
10
1
var_select12
var_select13
var_select14
var_select15
var_select16
var_select17
Variables selected jointly

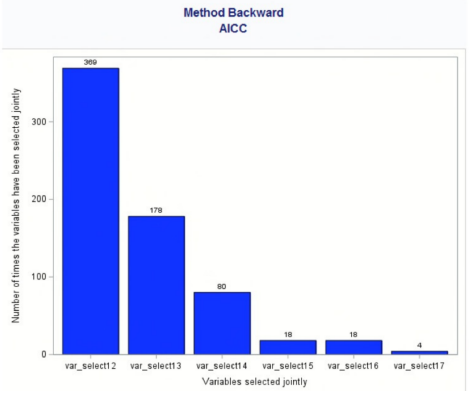

Method Forward
AIC
Number of times the variables have been selected jointly
555
216
39
10
10
1
var_select12
var_select13
var_select14
var_select15
var_select16
var_select17
Variables selected jointly


Method Forward
AICC
Number of times the variables have been selected jointly
555
215
39
10
10
1
var_select12
var_select13
var_select14
var_select15
var_select16
var_select17
Variables selected jointly


Method Backward
Without criteria
Number of times the variables have been selected jointly
323
119
31
5
5
1
var_select12
var_select13
var_select14
var_select15
var_select16
var_select17
Variables selected jointly


Method Backward
BIC
Number of times the variables have been selected jointly
355
160
60
9
9
2
var_select12
var_select13
var_select14
var_select15
var_select16
var_select17
Variables selected jointly

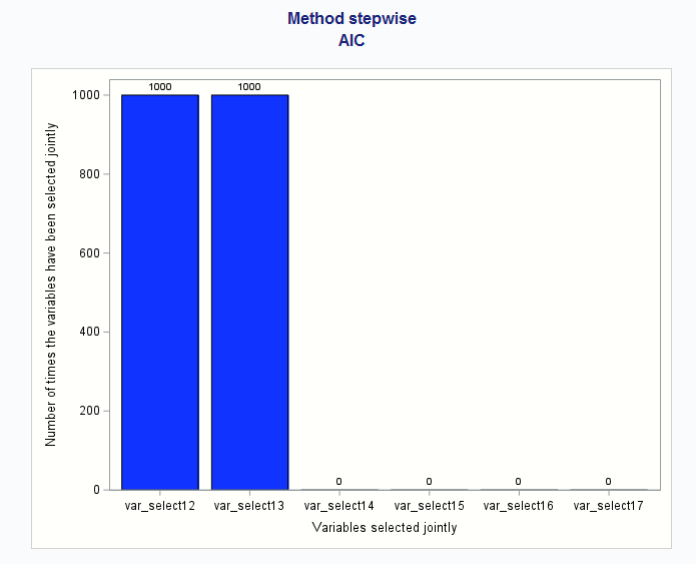

Method Backward
AIC
Number of times the variables have been selected jointly
427
257
139
51
51
16
var_select12
var_select13
var_select14
var_select15
var_select16
var_select17
Variables selected jointly


Method Backward
AICC
Number of times the variables have been selected jointly
369
178
80
18
18
4
var_select12
var_select13
var_select14
var_select15
var_select16
var_select17
Variables selected jointly

Regarding the *Forward* and *Stepwise* algorithms results, we may say that the *Backward* algorithm is the least efficient. Indeed, the figure above shows that, without information criteria, X_1 and X_2 have only 32.2% probability of being chosen jointly. It gets a little better with BIC and AICC as information criteria. Indeed, the joint probability are respectively 35.5% and 36.9%. Once again, AIC is the better information critera as the probability for X_1 and X_2 of being chosen jointly is 42.7%.

4.1.3 Third Data Generating Process

Unlike the first two data generated processes, this third DGP has a simple correlation structure with little noise coming from correlations between co-variables. The results are clear, for all algorithms and all information criteria, the variables selected are always X_1, X_2 and X_3 for every database. We can easily conclude that, when there is almost no noise, each algorithm is able to recognize which variable is truly providing information on Y.



4.2 Machine Learning

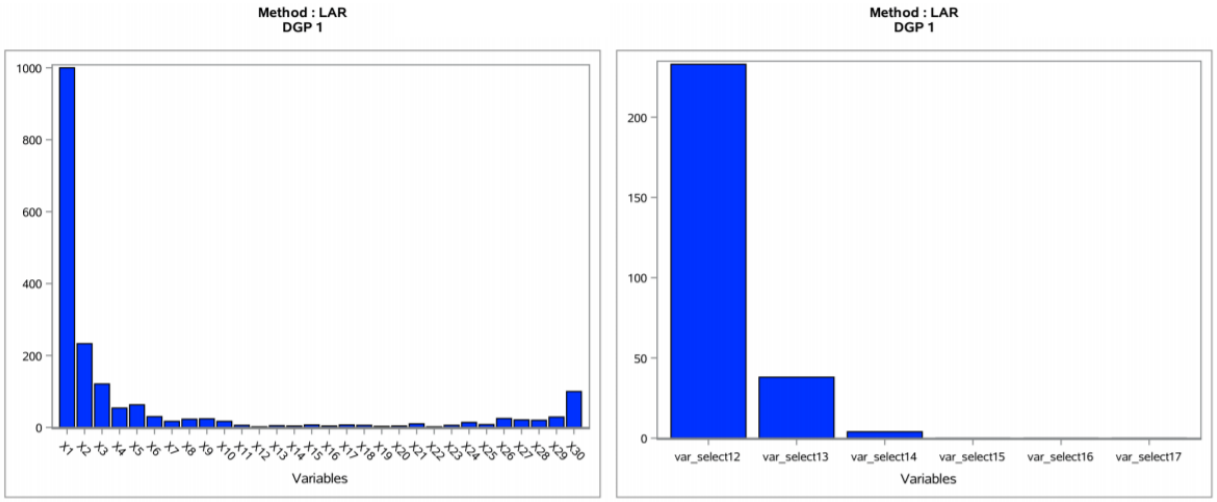
After having analyzed the results of the statistical learning's selection processes, we compare in this section the variables' results of machine learning algorithms.
As for the description of the previous results we will proceed by data generating process from the first to the third, commenting on the results of the three deployed algorithms:

- *LARS*
- *LASSO*
- *ElasticNet*

For the LARS algorithm, we ran it only with the AICC. For the last two algorithms, we ran them through k-fold thresholds and the stop criterion was the PRESS statistics. We obtain two rows of three graphs each. The first row represents the selection probabilities of each of the variables, and the second the joint selection probabilities according to the previous explanation [4]. Thus, from left to right, we visualize the probability and joint probability bar charts, for a 3-fold, then 5-fold, and finally 10-fold's selection.

4.2.1 First Data Generating Process

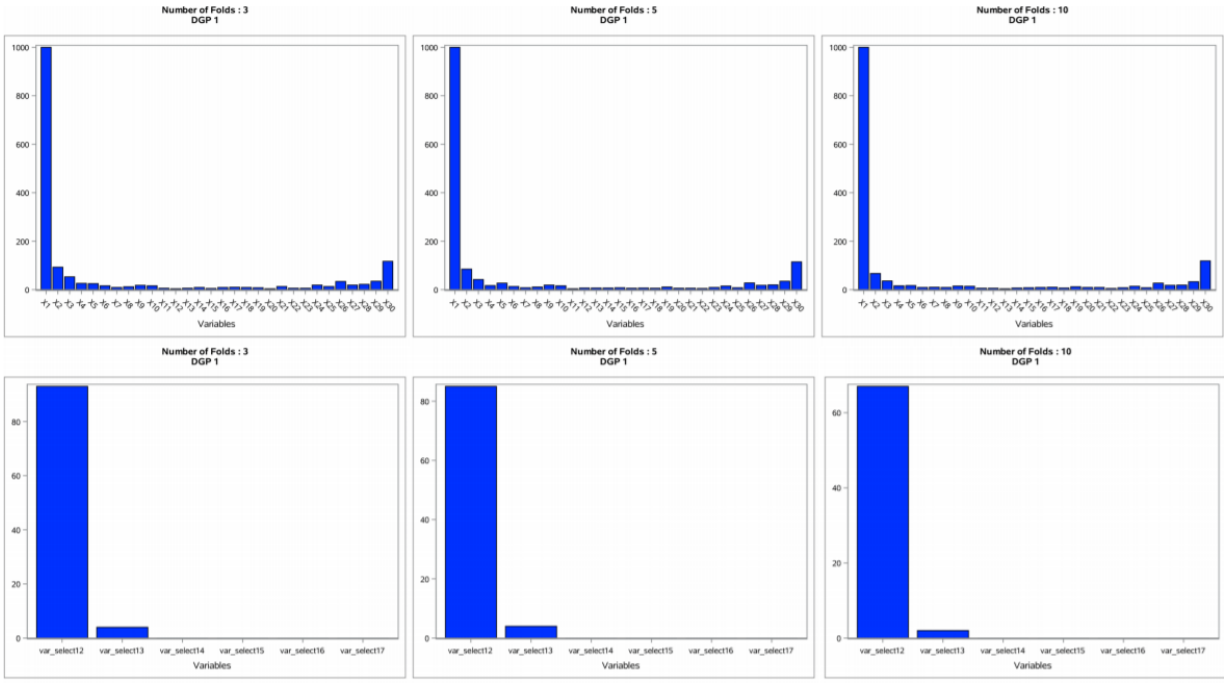
At first, for these three models, the data generated with this first process, here again we face the same issue: all models taken together, the variable X_1 is constantly chosen with an individual probability of selection equal to 1. The chosen correlation matrix appears to take in defect not only statistical learning selection's processes but also the machine learning ones. Moreover, we encounter the same problem regarding the explanatory power of the variable X_1 which erases the other variables' presence when they are themselves too close, to its explanatory power, i.e. reflecting perhaps the same determinants. Anyway, this implies again the unexpected but explained selection of the last variable. X_{30}, because very little correlated to X_1 is selected alone in 15% of the cases. A probability valid for the three algorithms, at each chosen folds.



However, the graph just above representing selection by *LARS*, provides a different answer from the next two others. Indeed, the variables X_2 to X_5, although very few times picked with *LASSO* or *ElasticNet*, have here selection probabilities of 25, 15, 8 and 9/100, respectively.
And, for the joint selection, *LARS* also makes a difference, jointly selecting the first three variables at a higher frequency than *LASSO* and *ElasticNet* allow and going up to 4 variables jointly picked: on one

[4] See Section 4.1 Statistical Learning in Results.

hand, X_1 and X_2 with a joint selection frequency of 25%, and on the other hand, joined to X_3 this frequency stays at nearly 5%.



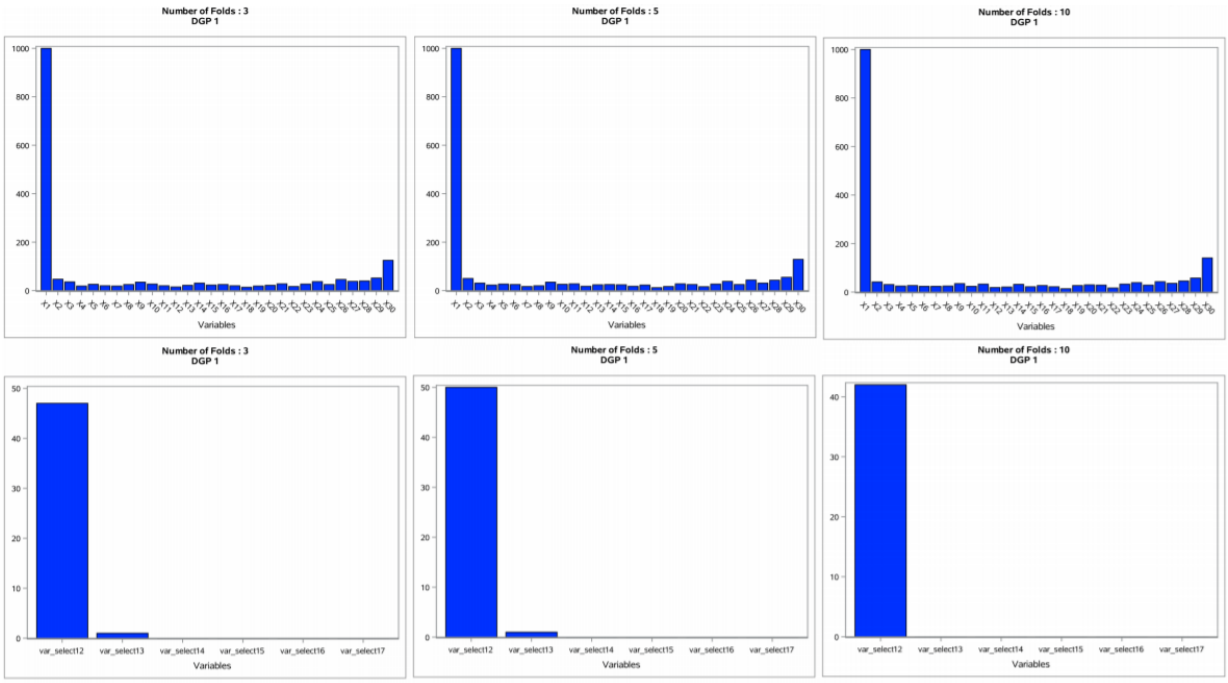
Following the analysis of the variables elected by *LARS*, we denote a lower accuracy for the algorithms using the $l1$ norm. They both seem to be more affected by the strong correlation between X_1 and the explained variable, and the decorrelation of X_1 with X_{30} : the bar charts are then substantially similar to those obtained with statistical learning. Finally, the joint probabilities remain a little more differentiated than the simple ones. But depending on the algorithm and the number of k-folds set, the following frequencies are appearing:

LASSO: X_1 and X_2 jointly:

- 3-fold gives a 9.5/100 frequency
- 5-fold : 8.5/100 frequency
- 10-fold : 7/100 frequency

LASSO: X_1, X_2 and X_3 together :

- 3-fold and 5-fold : approximately 0.5/100.
- 10-fold jointly selected around 2 time or 0.2%.



Finally, the joint use of the $l1$ and $l2$ norms via *ElasticNet* algorithm further refines the selection, as it reduces the probabilities, giving the here-under values:

ElasticNet: X_1 and X_2 jointly:

- 3-fold gives a 4.7/100 frequency
- 5-fold : 5/100 frequency
- 10-fold : 4.2/100 frequency

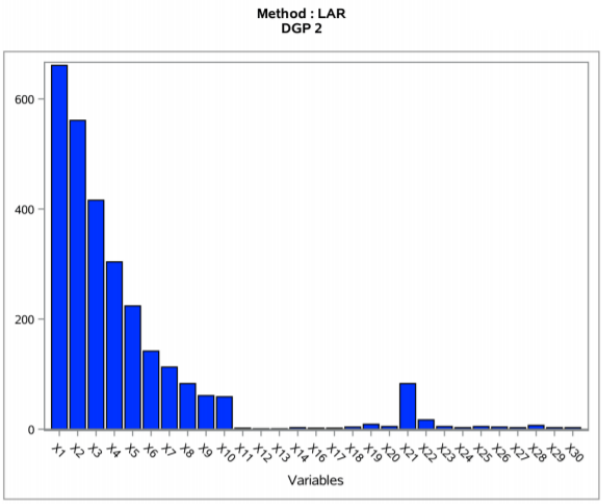
ElasticNet: X_1, X_2 and X_3 together :

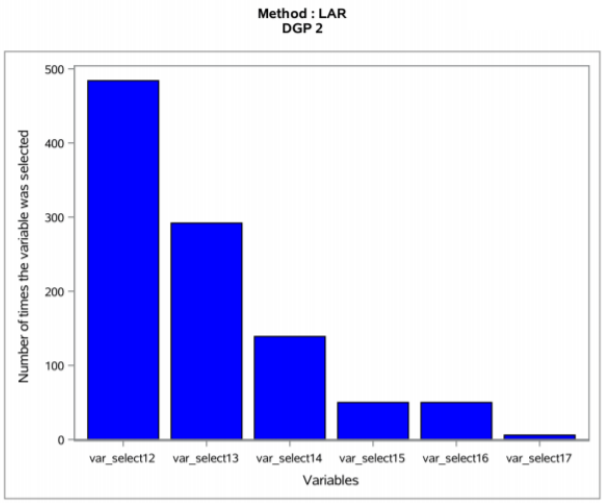
- 3-fold and 5-fold : one time or 0.1%.
- 10-fold no jointed selection

4.2.2 Second Data Generating Process

In this section, for comparison with the first controverted DGP's results obtained, we apply machine learning algorithms to the 2^{nd} 1000 generated data sets. Let us give more details about the very different results we obtained.

The same way we have seen in the statistical algorithms part, the data issued from the second DGP do not default the models like the first does. Here we can further compare *LARS*, *LASSO* and *ElasticNet* in between and to *Backward*, *Forward* and *Stepwise*. However, there are similarities between *LASSO* and *LARS*. For both, we shall note the frequency decreases from X_1 to X_{10} and the next more often selected variable is X_{21}. Yet, this same 21^{st} variable was also picked with the statistical models, always more frequently than its 11^{th} equivalent, and even than the first one with *Backward* algorithm. Here the frequency remains small but we can thus affirm the explanatory power that has X_{21} on Y.





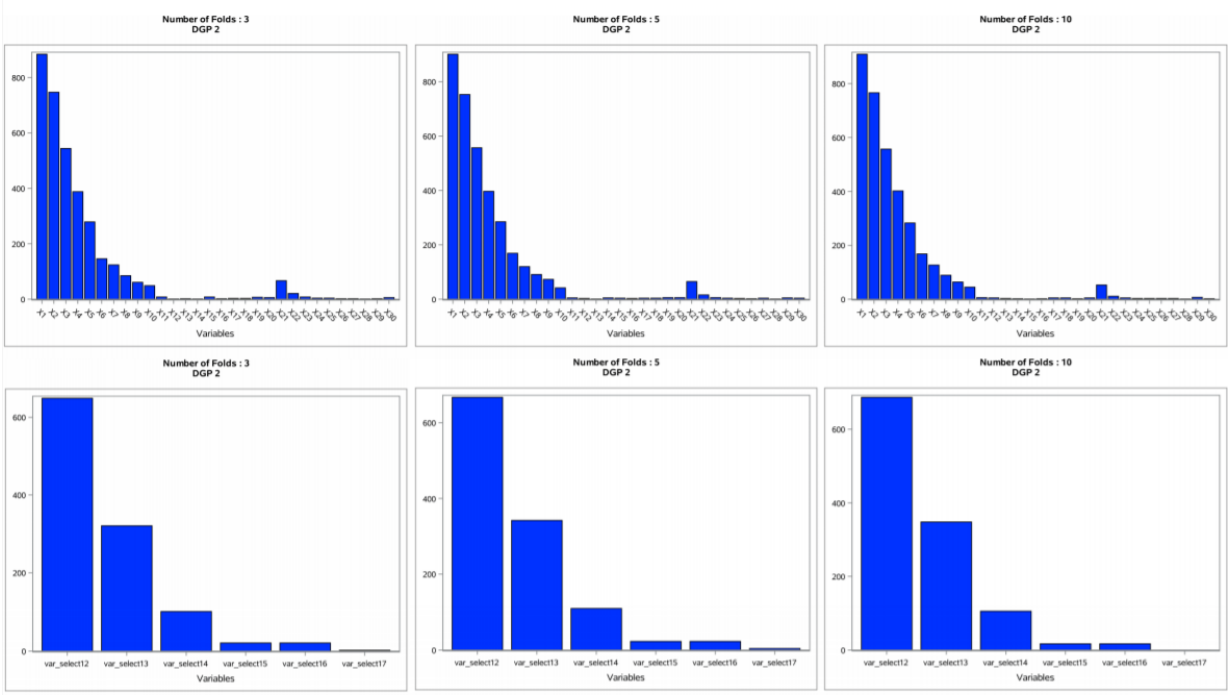
Let us add that *LARS* process globally picks the Xs fewer times than they are with *LASSO*. For instance, X_1 is chosen 6.3 times out of 10; X_3 is chosen more than 4 out of 10 times; X_5, more than 2/10 times and X_{10}, over 6 times out of 100, as shown on the above graph. For the joint selection, the two first X are almost selected half the time (i.e. a 0.5 probability), the frequency is then quite reduced as the third variable is add with a 30% probability, by half again with a value of 15% after the addition of the following variable, till it reduces to 5% when six or seven X are jointly picked and to 1 out of every 100 times from $X_{1\ to\ 8}$.
As said a few sentences earlier, the *LASSO* process gives bigger probabilities of selection, though there is no divergence according to the k-fold's number : we will therefore comment on one graph for models with 3, 5 or 10 folds combined. To compare with *LARS*, let us take the values for X_1 ; X_3 ; X_5 and X_{10} again :

- X_1: less than 90%
- X_3: approximately 55%
- X_5: less than 30%
- X_{10}: and finally, as for *LARS*, over 6/100 times.

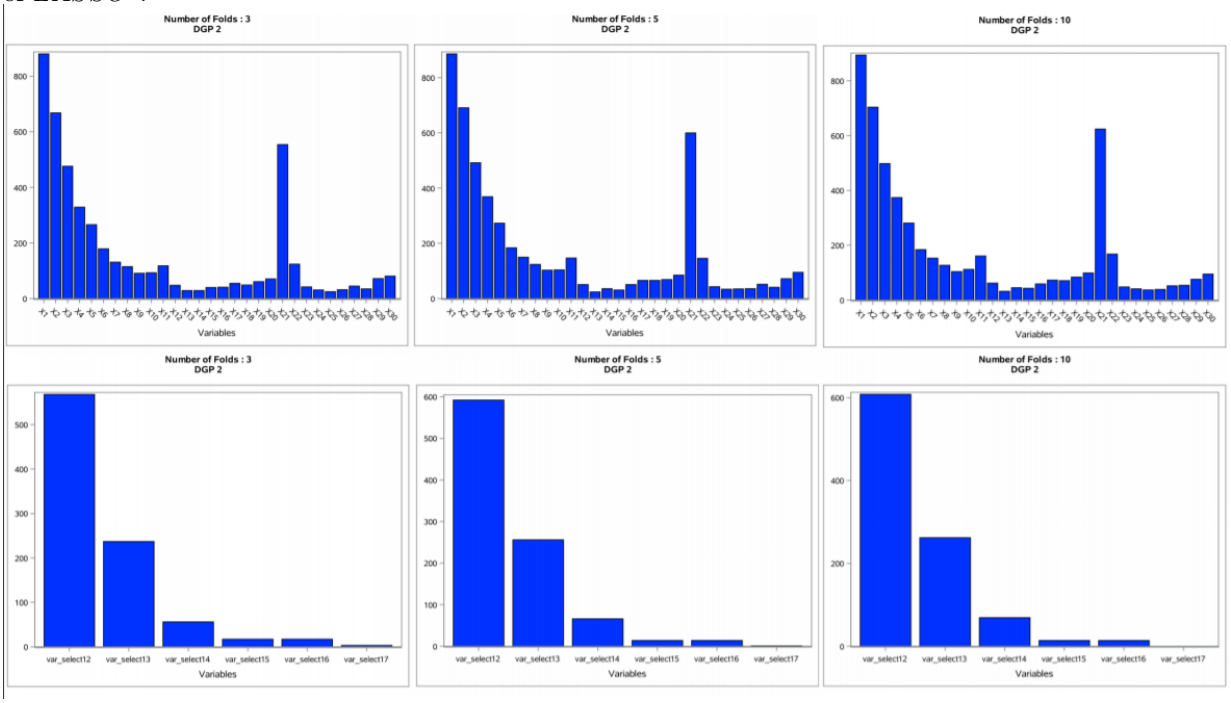
The decrease in variable selection frequency between the first and the 10^{th} is thus more abrupt than it is with *LARS*. For the joined probabilities, the frequency is, here again, more importantly distributed in favor of the first two Xs, approaching 70% and 35%, respectively. This implies a lower frequency of joint selection as the number of variables increases. Let us resume the results in the table above, we obtain frequencies as :

Probabilities of selection with *LASSO*			
Jointly selected Variables	3-fold	5-fold	10-fold
X1 - X2	65%	68%	70%
X1 - X3	33%	35%	35%
X1 - X4		15%	
X1 - X5 & X1 - X6		2%	
X1 - X7	1%	1%	none

To conclude, *LASSO* selects with higher frequency a few number of variables at the time than *LARS*, and with a lower one when there is more than four Xs.



Now we have a better understanding of the potential differencies within the diverse models. Yet, an acknowledgement of the ones *ElasticNet* has with *LASSO*, shall be required since *ElasticNet* is an improvement of *LASSO*[5].



As it is the case for the previous process, *ElasticNet* does not offer any divergence between the number of folds selected for the simple probabilities, but we can see them for the joined probabilities. The main difference with *LASSO*, is the frequency at which X_{21}, is selected : in 60% of the cases. It appears that an

[5]As explained in 2.2.3, $ElasticNet = Ridge + LASSO$

algorithm managing $l1$ and $l2$ norms, founds an important explanatory power in this variable. A comparable fact to what we observed on the bar charts of each processes classified in Statistical learning.
Another fact is $ElasticNet$ more often opts for every Xs. This way, every variable are selected at least 2 times in a 100 data sets and, about the decreasing probabilities of selection of the 10^{th} first, it pursues the same scheme as the $LARS$.

Finally, $ElasticNet$ provides sensitively the same joint probabilities than its equivalent precursors. The two first variables are almost selected more than half the time, and increases with the number of fold : for 3-fold it's at 56%, 59% for 5-fold and at 61% for 10-fold. Adding the following X, the frequency is once again quite reduced since the probability stands between 25% for the smallest number of k-folds and 30% for 10-fold. Afterwards, we have the same kind of gap we had with the $LASSO$: from $X_{1\ to\ 4}$ are only jointly selected 8 times out of 100. It is by half again with a value of 15% after the addition of the following variable, till it reduces to 5% when six or seven X are jointly picked and to 1 out of every 100 times from $X_{1\ to\ 8}$. The two following attached selections are 2/100 and the variables 1 to 7 are jointly chosen only with a probability of 1 or less, since this is not at all the case for the 10-fold $ElasticNet$ process.

4.2.3 Third Data Generating Process

This third data generating process being the simple correlation structure described above[6], we obtained the same results as with Statistical learning models : X_1„ X_2 and X_3 are picked in every algorithm for every number of fold. $LARS$, $LASSO$ and $ElasticNet$ algorithms are in a position to determine the Xs explaining the variable of interest.
Therefore, the bar chart represents exactly the same frequencies as the one shown in the Section 4.1.3 : Third data Generating Process.

[6]See Section 3.4.3 Third Data Generating Process.

5 Discussion

After having explained the differences that govern Statistical and Machine Learning categories ; the intuition and the functioning of the six deployed models ; statistical tools and criteria used in this article ; as well as the different methods of data simulation and results extraction, we acquired the necessary results to conclude our study.

Hence, this paper puts in evidence 3 mains results :

Definition of conditions under which algorithms perform poorly (DGP1):
When we consider high correlation structure between Y and X_1, each algorithm, whether Statistical Learning or Machine Learning one's, are taken in default. This means that, after having found an explanatory variable that exceed a certain degree of correlation with the target, algorithms are not able to find the next most highly correlated variable. This is especially true in the presence of inter-variables correlation.

However, results highlight differences in the way that Statistical Learning and Machine Learning algorithms handle the steps after having found the most correlated variable. Indeed, while Statistical Learning algorithms seem to select randomly the next most correlated variable, Machine Learning algorithms show a tendency to select the right variables more often. By considering the structure of the correlations of DGP1, we would expect a decrease in the individual probabilities of being selected by the model, meaning that X_1 should be selected more often than X_2 and that X_2 should be selected more often that X_3 and so on...

Though, the number of time variable that X_{11} to X_{29} are selected by the two kind of algorithm put $Forward$ that this is not what happened. For instance, $LARS$ algorithms basically never select these variables, which is what we expected, while $Backward$ selects them a good number of time. Thus, when performed under this conditions, Machine Learning algorithms does better that Statistical Learning algorithm since it provides a less random selection.

Machine Learning results are more robust that Statistical Learning one's (DGP2):
By creating a second Data Generating Process, we wanted to see what differences remains between Machine Learning and Statistical learning algorithms when not taking in default. Thus, we created a less correlated correlation structure where X_1 explained 70% of Y's variations. Once again, X_1's probability of being selected should be higher than X_2, then X_2 probability of being selected should be higher than X_3 and so on... Eventually, we would expect probabilities of being selected to "converge" toward 0. We would also expect the same decrease in the joint probabilities of being selected.

Statistical Learning algorithm, and especially $Backward$, do not fit our expectations. Indeed, in the $Backward$ set of selected variable, X_1 individual probability of being selected was lower than X_21. This highlights $Backward$'s lack of robustness. As for $Stepwise$ and $Forward$, results were closer to what we were expecting but still showing unexpected individual probabilities of being selected. Indeed, X_21 individual probability of being selected were higher than X_3 but still lower than X_1.

Machine Learning algorithms' results fitted what we supposed. $LARS$, $LASSO$ and $ElasticNet$ were less affected by the noise coming from correlation between explanatory variables. Thus, these algorithms are better suited than Statistical Learning algorithm for noisy correlation structure.

Machine and Statistical Learning provide the same results when significant correlation are attribute to few variables (DGP3):
Ultimately, the last Data Generating Process allows us to show the efficiency of all the models in detecting the significant correlation of the Xs with the variable of interest. During this third data simulation method we have voluntarily established the values of the correlations between X_1, X_2, X_3 and Y, in order to test the capacity of the selection processes to eliminate the variables whose correlation is not significant. Attempting to confound the selection, we also set up random non-significant correlations between the Xs. Whether it is for the 3 Machine Learning models and no matter the number of folds chosen, or for the 3 Statistical Learning models and regardless of the criteria, no model was found to fail.

For each of them, the first three explanatory variables are selected with a probability of 1. Their joint selection frequency is therefore also 100%, in the case of X_1 and X_2 attached but also when X_3 is added.

To conclude, even if *Stepwise* and $Forward$ were close to our expectations for the DGP2 ; Machine Learning algorithms are more accurate and performs better than Statistical Learning ones, when it regards variables selection.

We could have used other criteria to define which model is the best. For instance, the Mallows CP compares the accuracy and bias of the full model to those of models containing a subset of the predictors, providing the right balance regarding the number of predictors in the model. The lower the Mallows CP the more accurate the model, if it is close to the number of variables then the model is unbiased in estimating the true regression coefficients and predicting future responses. Nevertheless, if a predictor is highly correlated with another predictor, the Mallows CP is not displayed in the results. We can therefore say that it would have been of no use in the case of the first data set simulation.

References

[1] Sadhan Samar Maiti Bishwa Nath Mukhexjee. On some properties of positive definite toeplitz matrices and their possible applications. 1987.

[2] DeepAI. Statistical learning theory. https://deepai.org/machine-learning-glossary-and-terms/statistical-learning-theory.

[3] Feki Younes Edouard Duchesnay, Tommy Löfstedt. *Statistics and Machine Learning in Python (Release 0.3 beta)*. 2020.

[4] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–451, 2004. https://www.bibsonomy.org/bibtex/2e29f17ef93e84e4b68a633687d55ea0d/jabreftest.

[5] Universalis encyclopédie. Inférence statistique classique. https://www.universalis.fr/encyclopedie/statistique/4-inference-statistique-classique/.

[6] Eric Enge. Machine learning vs. statistical learning. 2020. https://blogs.perficient.com/2018/01/29/machine-learning-vs-statistical-learning/.

[7] Anisse Ismaili et Pierre Gaillard. Le lasso, ou comment choisir parmi un grand nombre de variables à l'aide de peu d'observations. 2009.

[8] M. Fernández-Delgado, M.S. Sirsat, E. Cernadas, S. Alawadi, S. Barro, and M. Febrero-Bande. An extensive experimental survey of regression methods. *Neural Networks*, 111:11–34, 2019. https://www.sciencedirect.com/science/article/pii/S0893608018303411.

[9] Li Lexin Guo Zifang, Lu Wenbin. Forward stagewise shrinkage and addition for high dimensional censored regression. 32, 2015. https://doi.org/10.1007/s12561-014-9114-4.

[10] SAS Institute. The glmselect procedure. https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.4/statug/statug_glmselect_details27.htm.

[11] Nicolas Jacquemet and Bruno Crepon. *Économétrie : méthodes et applications*. 10 2018.

[12] Bellman RE. *Adaptive Control Processes. A Guided Tour*. Princeton University Press, Princeton, NJ, 1961.

[13] Adrià Serra. Explaining subset selection and regularization methods for linear-squares. 2020.

[14] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B, vol. 58, no 1*, pages p. 267–288, 1996.

[15] Ryan J. Tibshirani. A general framework for fast stagewise algorithms. *Neural Networks*, 2015. https://www.jmlr.org/papers/volume16/tibshirani15a/tibshirani15a.pdf.

6 Appendix

6.1 First data generating process : correlation matrix

covMatrix

	Y	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16	X17	X18	X19	X20	X21	X22	X23	X24	X25	X26	X27	X28	X29	X30
Y	1	0.9641764	0.9317492	0.9044704	0.870387	0.8349235	0.7992319	0.750315	0.7052505	0.6754155	0.6507411	0.6085209	0.5916232	0.5415902	0.510315	0.4868167	0.4413081	0.4039964	0.3607801	0.339826	0.3134233	0.2943654	0.2691389	0.2471407	0.2170417	0.1876148	0.1565197	0.1235164	0.0832643	0.0552055	0.0357276
X1	0.9641764	1	0.9663126	0.9405101	0.9155836	0.8736034	0.8381518	0.8012121	0.7651005	0.7275128	0.699526	0.6612901	0.6493009	0.610249	0.5773537	0.5582238	0.5147795	0.469763	0.4265947	0.4032643	0.3790459	0.350183	0.3205761	0.2959736	0.2633423	0.2365917	0.2048365	0.1711971	0.1287249	0.0993099	0.0862646
X2	0.9317492	0.9663126	1	0.9642004	0.9378338	0.9016382	0.8623702	0.8161176	0.770057	0.7340894	0.7040744	0.6584098	0.6460606	0.6123492	0.5778338	0.5593639	0.509907	0.4733033	0.4209061	0.4028803	0.3783618	0.3413501	0.3158836	0.2954095	0.2682148	0.2457006	0.2153735	0.1776778	0.1329973	0.099622	0.0855326
X3	0.9044704	0.9405101	0.9642004	1	0.9638284	0.9248845	0.8918932	0.850165	0.8075488	0.770369	0.740474	0.6960936	0.6908371	0.6549175	0.6138734	0.5937954	0.5421662	0.5012421	0.4524812	0.4264866	0.4013321	0.3677288	0.3417102	0.3160756	0.2880408	0.2714311	0.2453645	0.2009961	0.1542394	0.1203	0.1081188
X4	0.870387	0.9155836	0.9378338	0.9638284	1	0.9633843	0.9255206	0.8855926	0.8474047	0.809565	0.78003	0.7316532	0.7249805	0.6870207	0.6409841	0.6215422	0.5636724	0.5215362	0.4761236	0.4408881	0.4191779	0.3831503	0.3478308	0.3238164	0.2973177	0.2832043	0.2607381	0.2194179	0.1724092	0.1440984	0.1312571
X5	0.8349235	0.8736034	0.9016382	0.9248845	0.9633843	1	0.9588239	0.9242004	0.8829643	0.840096	0.8046805	0.7535074	0.7367057	0.6978218	0.6461086	0.6380198	0.5828863	0.5463546	0.5029583	0.4651065	0.4436604	0.4122052	0.3767657	0.3564716	0.3178038	0.3031263	0.2893129	0.2527333	0.2065287	0.1727093	0.1546595
X6	0.7992319	0.8381518	0.8623702	0.8918932	0.9255206	0.9588239	1	0.9554515	0.9146955	0.8759076	0.8394839	0.7928353	0.78003	0.7408221	0.6941134	0.6863486	0.6321032	0.5886829	0.5433063	0.5055266	0.480096	0.4445485	0.4143894	0.3918392	0.3482628	0.3332733	0.3255446	0.2926493	0.2507651	0.2133213	0.1971197
X7	0.750315	0.8012121	0.8161176	0.850165	0.8855926	0.9242004	0.9554515	1	0.9548155	0.9153435	0.8842244	0.8420282	0.8362076	0.7932673	0.7436424	0.7348335	0.6776838	0.6278668	0.5871227	0.5487549	0.5309211	0.5057066	0.4675668	0.4494209	0.4043564	0.3834023	0.3707531	0.3459826	0.3065227	0.2662466	0.2529253
X8	0.7052505	0.7651005	0.770057	0.8075488	0.8474047	0.8829643	0.9146955	0.9548155	1	0.9541074	0.9287009	0.8840204	0.8728113	0.8291509	0.7842064	0.7667687	0.7157276	0.6615182	0.6271707	0.5815782	0.5635524	0.5342214	0.4923252	0.4742634	0.4387879	0.4141254	0.4064086	0.3762376	0.3352295	0.2929493	0.2767117
X9	0.6754155	0.7275128	0.7340894	0.770369	0.809565	0.840096	0.8759076	0.9153435	0.9541074	1	0.9685449	0.9324572	0.9140474	0.8562856	0.8140054	0.7941074	0.7487789	0.6935974	0.6609421	0.6114611	0.5971677	0.5634203	0.5216322	0.5060786	0.4682748	0.4463966	0.4351515	0.4050885	0.3663366	0.319952	0.3013621
X10	0.6507411	0.699526	0.7040744	0.740474	0.78003	0.8046805	0.8394839	0.8842244	0.9287009	0.9685449	1	0.9543234	0.9242844	0.8684428	0.8248665	0.8017762	0.7490669	0.6920972	0.6648905	0.6178578	0.6014761	0.5638884	0.5307771	0.5183078	0.4797	0.4583018	0.4517492	0.420402	0.3885269	0.3437264	0.3182118
X11	0.6085209	0.6612901	0.6584098	0.6960936	0.7316532	0.7535074	0.7928353	0.8420282	0.8840204	0.9324572	0.9543234	1	0.9534713	0.8985179	0.8679148	0.8485929	0.7980438	0.7451185	0.7151275	0.6642064	0.6505611	0.6085329	0.5830783	0.5726373	0.5342334	0.5048425	0.4936334	0.4650225	0.4249265	0.3841224	0.3622202
X12	0.5916232	0.6493009	0.6460606	0.6908371	0.7249805	0.7367057	0.78003	0.8362076	0.8728113	0.9140474	0.9242844	0.9534713	1	0.9445905	0.9014821	0.8754395	0.8288389	0.7815062	0.7518512	0.6969577	0.6860846	0.6479568	0.620006	0.6086649	0.5718692	0.5425863	0.5245125	0.5036784	0.4557096	0.4120252	0.389943
X13	0.5415902	0.610249	0.6123492	0.6549175	0.6870207	0.6978218	0.7408221	0.7932673	0.8291509	0.8562856	0.8684428	0.8985179	0.9445905	1	0.9444104	0.909739	0.8515572	0.7962316	0.7816742	0.7345455	0.7153075	0.6812841	0.659778	0.6527453	0.6142214	0.5856466	0.560384	0.5364176	0.4958896	0.459802	0.4436604
X14	0.510315	0.5773537	0.5778338	0.6138734	0.6409841	0.6461086	0.6941134	0.7436424	0.7842064	0.8140054	0.8248665	0.8679148	0.9014821	0.9444104	1	0.959604	0.9070507	0.8492769	0.8332673	0.7958116	0.7724452	0.7435584	0.7189439	0.7030063	0.6672667	0.6346955	0.6074047	0.5686289	0.5233603	0.4853645	0.4687429
X15	0.4868167	0.5582238	0.5593639	0.5937954	0.6215422	0.6380198	0.6863486	0.7348335	0.7667687	0.7941074	0.8017762	0.8485929	0.8754395	0.909739	0.959604	1	0.9522712	0.8925533	0.870051	0.8220582	0.7989439	0.7807501	0.7559556	0.7319052	0.6888209	0.6531053	0.630363	0.5881428	0.5488269	0.5181278	0.4910291
X16	0.4413081	0.5147795	0.509907	0.5421662	0.5636724	0.5828863	0.6321032	0.6776838	0.7157276	0.7487789	0.7490669	0.7980438	0.8288389	0.8515572	0.9070507	0.9522712	1	0.9544434	0.9239604	0.8875488	0.8628743	0.8408521	0.8149055	0.7914191	0.7415782	0.7027903	0.6875488	0.6536814	0.6136334	0.5748455	0.5462466
X17	0.4039964	0.469763	0.4733033	0.5012421	0.5215362	0.5463546	0.5886829	0.6278668	0.6615182	0.6935974	0.6920972	0.7451185	0.7815062	0.7962316	0.8492769	0.8925533	0.9544434	1	0.9586319	0.919796	0.8870087	0.8653225	0.8457126	0.8188659	0.7752055	0.7366457	0.7135914	0.6766157	0.6427723	0.6069367	0.5756736
X18	0.3607801	0.4265947	0.4209061	0.4524812	0.4761236	0.5029583	0.5433063	0.5871227	0.6271707	0.6609421	0.6648905	0.7151275	0.7518512	0.7816742	0.8332673	0.870051	0.9239604	0.9586319	1	0.9563636	0.9250045	0.8994779	0.8785719	0.8507651	0.8117492	0.7715932	0.7516832	0.7079628	0.6774557	0.6405161	0.6063486
X19	0.339826	0.4032643	0.4028803	0.4264866	0.4408881	0.4651065	0.5055266	0.5487549	0.5815782	0.6114611	0.6178578	0.6642064	0.6969577	0.7345455	0.7958116	0.8220582	0.8875488	0.919796	0.9563636	1	0.9637324	0.9338254	0.9130753	0.8822082	0.8425563	0.7972277	0.7770537	0.7386619	0.7063186	0.6646025	0.6379358
X20	0.3134233	0.3790459	0.3783618	0.4013321	0.4191779	0.4436604	0.480096	0.5309211	0.5635524	0.5971677	0.6014761	0.6505611	0.6860846	0.7153075	0.7724452	0.7989439	0.8628743	0.8870087	0.9250045	0.9637324	1	0.9681248	0.9444104	0.9121152	0.8692469	0.8242424	0.8028443	0.759796	0.7334533	0.6958896	0.6683828
X21	0.2943654	0.350183	0.3413501	0.3677288	0.3831503	0.4122052	0.4445485	0.5057066	0.5342214	0.5634203	0.5638884	0.6085329	0.6479568	0.6812841	0.7435584	0.7807501	0.8408521	0.8653225	0.8994779	0.9338254	0.9681248	1	0.970021	0.9361176	0.9012181	0.8552295	0.8312151	0.7957276	0.7667447	0.7361296	0.7063906
X22	0.2691389	0.3205761	0.3158836	0.3417102	0.3478308	0.3767657	0.4143894	0.4675668	0.4923252	0.5216322	0.5307771	0.5830783	0.620006	0.659778	0.7189439	0.7559556	0.8149055	0.8457126	0.8785719	0.9130753	0.9444104	0.970021	1	0.9651605	0.930273	0.8905371	0.8694749	0.8416922	0.8141134	0.7820102	0.7429103
X23	0.2471407	0.2959736	0.2954095	0.3160756	0.3238164	0.3564716	0.3918392	0.4494209	0.4742634	0.5060786	0.5183078	0.5726373	0.6086649	0.6527453	0.7030063	0.7319052	0.7914191	0.8188659	0.8507651	0.8822082	0.9121152	0.9361176	0.9651605	1	0.9663246	0.9317252	0.9013981	0.8805281	0.8539934	0.8170657	0.7857186
X24	0.2170417	0.2633423	0.2682148	0.2880408	0.2973177	0.3178038	0.3482628	0.4043564	0.4387879	0.4682748	0.4797	0.5342334	0.5718692	0.6142214	0.6672667	0.6888209	0.7415782	0.7752055	0.8117492	0.8425563	0.8692469	0.9012181	0.930273	0.9663246	1	0.9673807	0.9320852	0.9034263	0.8764236	0.8376478	0.8125533
X25	0.1876148	0.2365917	0.2457006	0.2714311	0.2832043	0.3031263	0.3332733	0.3834023	0.4141254	0.4463966	0.4583018	0.5048425	0.5425863	0.5856466	0.6346955	0.6531053	0.7027903	0.7366457	0.7715932	0.7972277	0.8242424	0.8552295	0.8905371	0.9317252	0.9673807	1	0.9605761	0.9285449	0.8984098	0.8608221	0.8288869
X26	0.1565197	0.2048365	0.2153735	0.2453645	0.2607381	0.2893129	0.3255446	0.3707531	0.4064086	0.4351515	0.4517492	0.4936334	0.5245125	0.560384	0.6074047	0.630363	0.6875488	0.7135914	0.7516832	0.7770537	0.8028443	0.8312151	0.8694749	0.9013981	0.9320852	0.9605761	1	0.9630843	0.9383858	0.9026103	0.870267
X27	0.1235164	0.1711971	0.1776778	0.2009961	0.2194179	0.2527333	0.2926493	0.3459826	0.3762376	0.4050885	0.420402	0.4650225	0.5036784	0.5364176	0.5686289	0.5881428	0.6536814	0.6766157	0.7079628	0.7386619	0.759796	0.7957276	0.8416922	0.8805281	0.9034263	0.9285449	0.9630843	1	0.9678728	0.9320852	0.9042544
X28	0.0832643	0.1287249	0.1329973	0.1542394	0.1724092	0.2065287	0.2507651	0.3065227	0.3352295	0.3663366	0.3885269	0.4249265	0.4557096	0.4958896	0.5233603	0.5488269	0.6136334	0.6427723	0.6774557	0.7063186	0.7334533	0.7667447	0.8141134	0.8539934	0.8764236	0.8984098	0.9383858	0.9678728	1	0.9676328	0.9382058
X29	0.0552055	0.0993099	0.099622	0.1203	0.1440984	0.1727093	0.2133213	0.2662466	0.2929493	0.319952	0.3437264	0.3841224	0.4120252	0.459802	0.4853645	0.5181278	0.5748455	0.6069367	0.6405161	0.6646025	0.6958896	0.7361296	0.7820102	0.8170657	0.8376478	0.8608221	0.9026103	0.9320852	0.9676328	1	0.969961
X30	0.0357276	0.0862646	0.0855326	0.1081188	0.1312571	0.1546595	0.1971197	0.2529253	0.2767117	0.3013621	0.3182118	0.3622202	0.389943	0.4436604	0.4687429	0.4910291	0.5462466	0.5756736	0.6063486	0.6379358	0.6683828	0.7063906	0.7429103	0.7857186	0.8125533	0.8288869	0.870267	0.9042544	0.9382058	0.969961	1

Figure 3: Correlation matrix figures