However, when the economic cycle changes to bearish, and due to the previous explanation, the low met debt portfolios suffer an overperformance on the notorious market, specifically, the expected average yield of this portfolio is 1.86%, with an annualized volatility Of 16.59% and a sharpe ratio of 0.11 that can be seen in Table 9.

Figure 13: High and low net debt portfolios weighted about benchmark

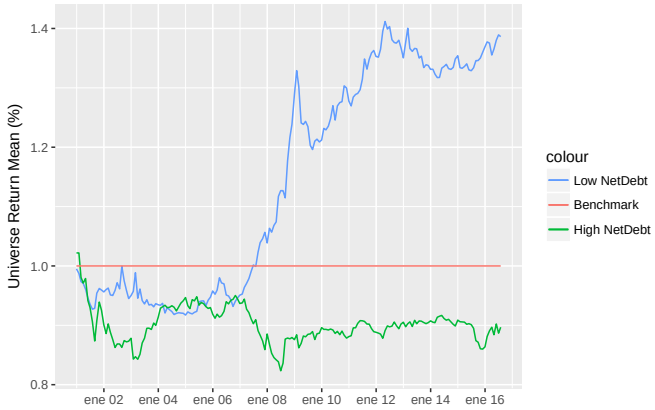
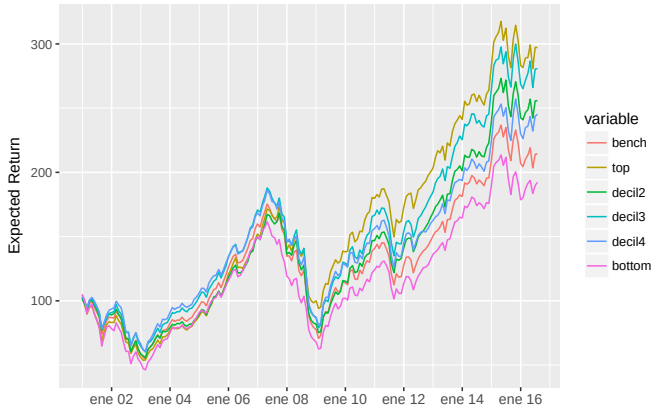


Figure 14:Shows all net debt portfolios and benchmark performance



	Low Net Debt	High Net Debt
Average Excess Return	1.864642	-0.5170789
Annualized Volatility (%)	16.59497	18.91296
Shape's Ratio	0.1123619	-0.02733993
Variation over time	1.386278	0.8967437

Table 9: Performance statistics of Net debt factor

6.1.4.2 ROE factor

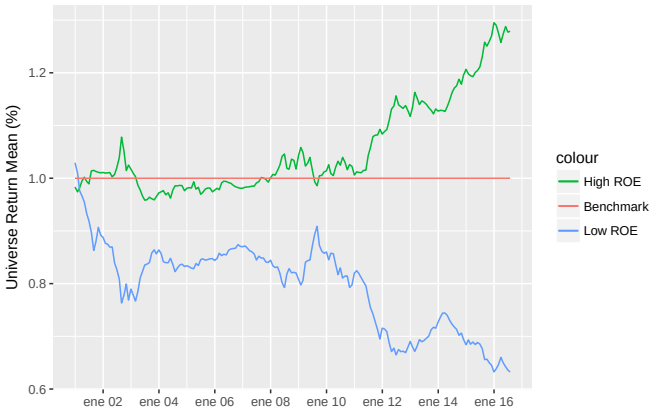
Return on equity (ROE) is the amount of net income returned as a percentage of shareholders equity. Return on equity measures a corporation's profitability by revealing how much profit a company generates with the money shareholders have invested.

ROE is expressed as a percentage and calculated as:

$$ROE = \frac{Net\ \ Income}{Shareholder's\ \ Equity}$$

The results associated with this ROE factor are shown below in Figures 15 and 16. In Figure 15 we can see how the low ROE portfolio has a much more negative cycle changes in the market, whereas in the previous bullish cycle the crisis behaved like the bechmark itself. High ROE portfolio is that prior to the crisis investors did not really pay too much attention to this factor so it remained the same as The benchmark, but during the years after the crisis of 2009 when the cycle changed for the better, investors began to look at this factor more strongly and those companies with greater return of equity (ROE), captured an overperformance that reminds of the own momemtum factor. More specifically, in Table 10 we show the statistical ratios obtained, with respect to the expected average yield obtained by the high ROE portfolio was 1.30% higher than the benchmark, an annualized volatility of 16.35% and a very low Sharpe's ratio of 0.07, on the other hand, the low ROE portfolio obtained an average expected return of 2.22% below the benchmark and the annualized volatility of 21.5%.

Figure 15: High and low ROE portfolios weighted about benchmark



	High ROE	Low ROE
Average Excess Return	1.308806	-2.222208
Annualized Volatility (%)	16.36549	21.50903
Shape's Ratio	0.07997352	-0.1033151
Variation over time	1.27892	0.6323498

Table 10: Performance statistics of ROE factor

6.2 Fundamental factor comparison

In this section, we will make a final comparison between all the fundamental factors explained in the previous section to give visibility to the strengths that are the ones we want to take to elaborate the new factor below. The highest expected average return obtained by the above factors is the high momemtum portfolio of 4.05% higher than the benchmark. Regarding the lowest annualized volatility obtained, the low volatility factor with 12.57% (similar to the low beta portfolio of 12.63%) obviously does honor it is own definition. Regarding the best ratio of Sharpe also refers to the high momemtum portfolio with a 0.26.

But in addition to taking into account the statistics of each factor, we will also try to capture the economical strengths in different cycles ocurrend on.

7. Smart Beta

After carefully analyzing the risk factors in the previous section, their weaknesses and strengths, the relevant question is, therefore, how to better extract the premium of a factor efficiently. Amenc et al. (2014a) present how the Smart Beta approach, whose main idea is to apply an intelligent weighting scheme to a selection of stocks, allows the construction of indexes of factors that are not only exposed to factors risk, but also avoid exposure to unrewarded risks. This approach, called **"smart factor indices"** [17] can be summarized as follows.

The explicit selection of stocks provides the desired slope, the beta, while the intelligent weighting scheme addresses the issues of concentration and it diversifies specific and unrewarded risks. Therefore, "Smart Beta approach" builds indexes of factors that explicitly seek exposures to rewarded risk factors, while unrewarded diversification of distance risks. The results we obtain suggest that factor indices lead to significant improvements in risk adjusted performance.

The flexible index build process used in second generation intelligent beta indexes enables you to take advantage of all the benefits of the smart beta, where the stock slection defines exposure to right (rewarded) risk factors and the intelligent weighting scheme allow unrewarded risks to be reduced.

7.1 Elementary Smart Beta

After presenting the concept of smart beta, we wanted in this research to make a first approximation that consists of constructing a new factor to gather the strengths of all the previous ones under a weighting done by us but in an "elemental" way, trying to leave aside tt is weaknesses with the objective of taking advantage of the upward cycles and avoiding or being able to cushion in bad times during a bearish economic cycle, for this reason we decided to rename it as "elemental smart beta".

7.1.1 Construction of the elemental smart beta factor

The process to construct this new factor is similar to the one used and explained in the chapter of data processing to find the classification in quintiles of the other factors that compose the data set. To achieve this, we first need to calculate the z-score values, normalize the values of each of the factors since it is not appropriate to use the original data since each was on a different scale, therefore, once normalized we can already compare them. We must denote that, in order to carry out this computation and following the structure we have been maintaining since the beginning, the computation has been made in function of the sector to which they belong and within each factor of the period, in other words, each value has been normalized by the period to which it belongs and within the sector to which it belongs.

Once we have the normalized values, and here comes one of the reasons why this new factor we will consider "elemental smart beta" is that we will weigh the values of each factor according to our criterion according to the average yield expected from factors presented in the previous section. As we can see in the following table 11, we present all the previous factors, their average expected performance, and the weighting that we have decided to assign to it according to our "elementary" criterion.

Factor	Expected Return	Weighted Coefficient
Momemtum	4.05%	30%
Earning Yield	3.21%	20%
Dividend Yield	2.04%	15%
NetDebt	1.86%	12%
Volatility	1.85%	10%
Beta	1.80%	8%
ROE	1.30%	5%

Table 11: Expected average yields of each factor and the weights associated with them.

7.1.2 Results obtained by the elemental smart beta factor

In this section we show the results achieved by the new elementary smart beta factor, which we will henceforth denote as ESM. If we look at the chart below, we get a pleasant surprise because at first glance we see how the portfolio built under the high ESM values gives us some rather pronounced profitability and gives us the feeling of having reduced the volatility. Specifically, if we look at the statistical results in Table 12, we see how the high ESM portfolio achieves an expected average return of 5.04% above the benchmark itself, an annualized volatility of 15.02% and a Sharpe ratio of 0.336, While this has generated a variation at the end of the time horizon of 2.36%. On the other hand, the low ESM portfolio has performed quite badly, obtaining an expected return of 6.19% below the benchmark itself, a volatility of more than 23.7% and a Sharpe ratio of -0.26.

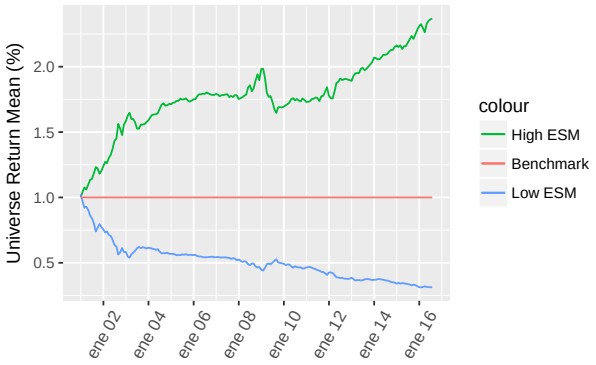
If we try to compare the results with the previous ones in order to check how close we are to achieving the ESM's own objective, we see how this ESM factor has achieved a higher performance than any of the other factors mentioned above. But how does he seem to have collected such fortresses? If we look at the graph we could say how in an earlier period of the crisis of 2009 seems to have captured the overperformance of the value factor (dividend yield and earning yield) that as we explained in the previous chapter tend to have a overperformce to the benchmark in bullish cycles. At the same time, when the cycle changed to bearish due to the crisis itself, this new ESM factor seems to have taken advantage of the strengths of the low risk factor (beta and volatility) to take advantage of the fall of the market itself and get a positive overperformance. Afterwards it suffered a fall next to the market but perhaps due to the quality factor (Netdebt EBITDA and ROE), ending up capturing the upward trend of the momemtum factor whose superior performance to the market is notorious and we have already reasoned previously.

Therefore, we can say that this ESM factor has obtained an increase in the expected average yield higher than the momemtum factor that was the one that previously improved yield, and has reduced annualized volatility compared to other factors, obviously without achieving low volatility factor that by it is own definition would not even make sense. Regarding Sharpe's ratio, a significant improvement has also been achieved in relation to the larger factor. We can say that we have achieved our goal under the construction of this ESM factor, by increasing the expected average yield, the volatility is the smaller, then we have achieved somewhat the objective of to obtain greater profitability with a lower volatility, so that is the objective.

Table 12: statistics performance of ESM factor

	High ESM	Low ESM
Excess Return	5.049712	-6.190875
Volatility	15.02863	23.76965
Ratio Sharpe	0.3360062	-0.260453
Portfolio Variation	2.365957	-0.3122333

Figure 17: Elementary Smart Beta Factor



8. Machine learning approach: next horizont to smart beta

8.1 Preliminaries

Machine learning is one of the fastest growing areas of computer science, with far-reaching applications. The aim of this chapter is the application of advanced statistical techniques such as machine learning is intended to build a new factor that combines the potentially positive characteristics of each of the factors explained above in order to be able to capture or capture the strengths of these factors and to dampen or avoid its weaknesses in a dynamic portfolio capable of knowing in each period how to maximize the average expected performance and, above all, to minimize annualized volatility in order to achieve the fundamental objective of the research.

8.2 Machine Learning Techniques

The term machine learning refers to the automated detection of meaningful patterns in data. By definition, machine learning is a branch of "artificial intelligence" where it is main purpose is converting experience into expertise or knowledge where the input to a learning algorithm is training data, representing experience, and

the output is some expertise. That is, to induce learning through feedback based on information. Roughly, the dynamics could be encircled in 3 large groups: pre-processing of data or information, training and post-processing of the results obtained.

Below are defined the fundamental elements of any learning system: inputs, outputs and types of algorithms. The entries are given in the form of an attribute vector and are called "instances". These attributes can be classified into "nominal attributes" if they take values within any finite set, or "numerical attributes" if they take real values within a finite or infinite set.

There are different types of learning algorithms which are governed by the type of output of the same:

1. Supervised learning: with this algorithm we can create a function that relates or establishes a correspondence between the output variables and the input variables, that is, transforms the input data into the results we expect. The most used supervised learning algorithms are Regression, Decision Tree, Random Forest, KNN, Logistic Regression.
2. Unsupervised learning: in this case, we do not have any objective or variable variables that we have to predict, if not rather, we seek to achieve groupings of the population in different groups. Since we do not have information about the input categories, it treats the input objects as a set of random variables and constructs a density model for that set, the algorithm being able to recognize patterns in order to be able to label the new inputs. Rata the input objects as a set of random variables, a density model being constructed for the data set. The most used non-supervised learning algorithms are: Apriori algorithm, K-means.
3. Reinforcement Learning: Using this algorithm, the machine is trained to make specific decisions. It works this way: the machine is exposed to an environment where it trains itself continually using trial and error. This machine learns from past experience and tries to capture the best possible knowledge to make accurate business decisions. Example of Reinforcement Learning: Markov Decision Process

8.2.1 Decission trees

A decision tree is a predictor, $h : X \to Y$, that predicts the label associated with an instance x by traveling from a root node of a tree to a leaf. For simplicity we focus on the binary classification setting, namely, Y=$\{0,1\}$, but decision trees can be applied for other prediction problems as well. At each node on the root-to-leaf path, the successor child is chosen on the basis of a splitting of the input space. Usually, the splitting is based on one of the features of x or on a predefined set of splitting rules. A leaf contains a specific label.

A popular splitting rule at internal nodes of the tree is based on thresholding the value of a single feature. That is, we move to the right or left child of the node on the basis of $\mathbb{1}_{[x_i<\theta]}$, where $i \epsilon [d]$ is the index of the relevant feature and $\theta \epsilon R$ is the threshold. In such cases, we can think of a decision tree as a splitting of the instance space, $\mathbb{X} = \mathbb{R}^d$, into cells, where each leaf of the tree corresponds to one cell.

8.2.2 Random Forest

In particular, in this research we have applied the technique or algorithm "Random Forest". Another way to reduce the danger of overfitting is by constructing an ensemble of trees. In particular, in the following we describe the method of random forests, introduced by Breiman (2001) [12].This is based on a combination of predictor trees such that each tree depends on the values of a random vector independently tested and with the same distribution for each of these, let's say it is a substantial modification of bagging that builds a long collection of uncorrelated trees and then average them.

The essential idea of bagging is to average many noisy but approximately unbiased models, and therefore reduce variation. But why trees are the ideal candidates for bagging? This is because they can register complex interaction structures in the data, and if they grow deep enough, they have relatively low bias. Since trees are notoriously noisy, they benefit greatly by averaging.

The objective is to create a model that predicts the value of a target variable based on various input variables, each inner node corresponds to one of the input variables and each sheet represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf.

The most known or used decision trees are of two main types:

- Classification trees are when the predicted result is the class to which the data belong.
- Regression trees are when the predicted outcome can be considered a real number (for example, the price of a house, or the length of a patient's stay in a hospital).

8.2.2.1 Random forest: Influence measures

There are two ratios to explain how important the variables of the dataset are when explaining the response that we seek to optimize. In our case, we obtain them through the function of the package "RandomForest" of R-study software that we have been using, and are: **"% IncMSE"** and **"IncNodePutity"**.

The first one is the most robust measure, it offers more information and can be interpreted as follows, if a predictor is important in the current model, then assigning other values to that random predictor should have a negative influence on the prediction, the same model to predict from original data should give worse predictions. Therefore, a predictive measure (MSE) is taken with the original dataset and then with the dataset exchanged, and compared in some way. In a way, especially since we expect the original MSE to always be smaller, we can make a difference. Therefore, the higher the better or more importantly basically.

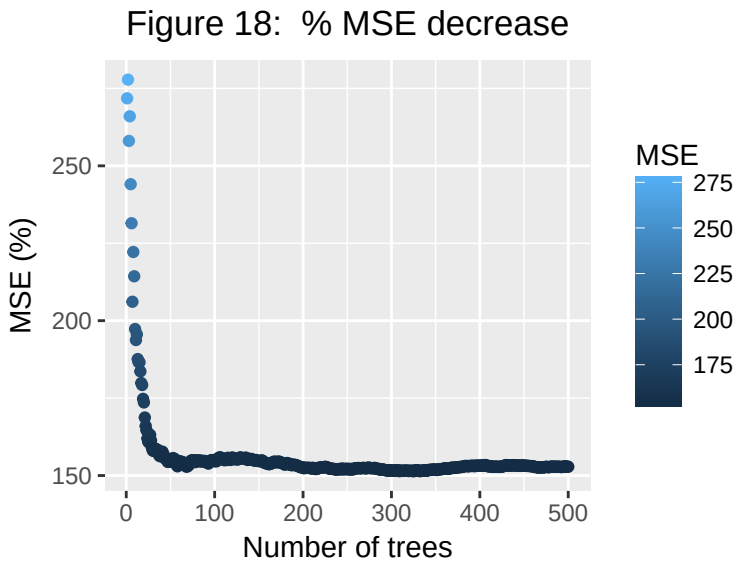
For the second measure, in each division, it is possible to calculate how much reduces the impurity of the node. The most useful variables achieve greater increases in the purity of the node, that is, find a division that has a high variance between nodes and a small intra-nodal variance. "IncNodePurity" is partial and should only be used if the extra computation time to calculate IncMSE% is unacceptable, which in our case is not really relevant since both computations provide a similar calculation time.

8.2.2.2 Random forest: Applications in our research

The procedure used in this research, it is worth pointing out that it will have two objectives. The first objective is to create a function that computes regression trees for each of the monthly periods in order to build a portfolio that we will denotas as *high smart beta* (the continuation of the previous high elementary smart beta ESM), based on the criterion previously explained of the importance of the variables in each period. With this, we want to develop a dynamic portfolio that, depending on it, is able of capturing the power of the most important variable in the immediately previous period, being able to recognize it and invest in the current period according to it.

The second objective or perhaps it can be denoted as a variant of the previous one, is to be able to build a portfolio high smart beta, but in this case, capturing the most important variable during the three periods immediately previous to the present one, and investing in the following according to these. Let is say that we are trying to adjust that procedure to see how well or how badly it works.

It should be noted that the adjustment of the number of trees to be computed in each period has been tested under the criterion of minimizing the mean square error obtained as a function of the number of trees calculated. To give a visual sample of the same, below we can see a figure where the% MSE obtained is plotted as a function of the number of trees computed. It is observed how a smaller number of trees greater MSE obtained, but this percentage is reduced when we increase that number until there are around 300 trees where it seems that this percentage begins to stabilize, for this reason the value of the parameter ntrees to which reference we decided to leave it in this amount.

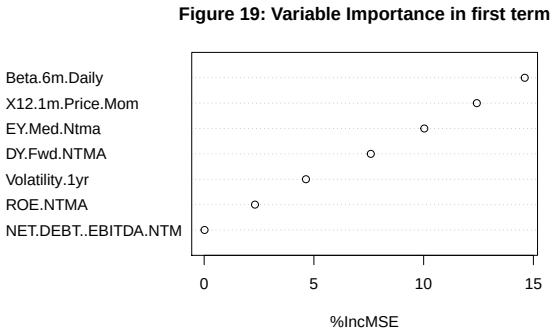


8.2.2.3 First and second term variable importances computation

In order to continue showing the development, in this section, we show the results obtained by the algorithm used during the first two periods in order to contribute the conclusions that we will make for each and every period of the entire time horizon. For period one, table 13 is shown both important measures explained in the previous sections, although as we have clarified the one used in this research has been IncMSE. Most relevant in this table is that the most important variable is the beta factor with 31.75%, which is interpreted as the increase of 31.75% of the MSE in the prediction of the average returns expected by the portfolio if we extract this variable from the model , followed by the momemtum factor with 23.27%. These results are also reflected in figure 18 attached to the table showing the classification of all variables according to their importance.

	IncMSE	IncNodePurity
Beta	31.757874	12141.248
Momemtum	23.276754	12217.477
Earn.Yld	10.919058	8297.780
Divd.Yld	8.406248	5840.536
Volatility	4.960217	8239.339
ROE	2.011630	7284.472
NET.DEBT	1.774	5972.498

Table 13: Importance measures of each factor.

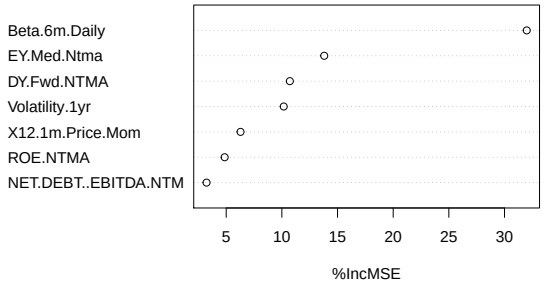


Next we show the results but for the second period as a visual explanation to get a clear the algorithm used. In table 14 we can see how for this second period agrees the most important variable as the beta factor which would contribute an increase of the MSE in predicting the expected average yields of 68.47%, being in this period almost double against the second more important variable than in this case would be the earning yield factor with a 29.51%. This can be explained, and we remember an affirmation made in a previous section, that the beta factor is not a rotational factor as other as the momemtum factor so it would explain better the variable response during periods longer than other factors that suffer more rotation. Again we present Figure 19 where the classification of the table attached to its side is graphed.

	IncMSE	IncNodePurity
Beta	68.4769871	15611.874
Earn.Yld	29.5199103	8229.745
Divd.Yld	11.2605642	5194.761
Volatility	11.1924177	8989.533
Momemtum	6.6287787	6300.367
ROE	2.1791019	3003.160
NET.DEBT	0.8122283	2932.054

Table 14: Importance measures of each factor.

Figure 20: Variable Importance in second term



Finally, we generate the algorithm that period to period stores the most important variable and we build the portfolio high smart beta and low smart beta factor.

8.3 Smart beta factor results

In this section we will show the results obtained by the portfolios built under the procedure explained above for both the portfolio built with one month of training and three months of training. We will show under the same structure that we have maintained throughout the research, a table with the statistical results obtained by both portfolios and a graph where you can clearly visualize the behavior obtained by each of these portfolios.

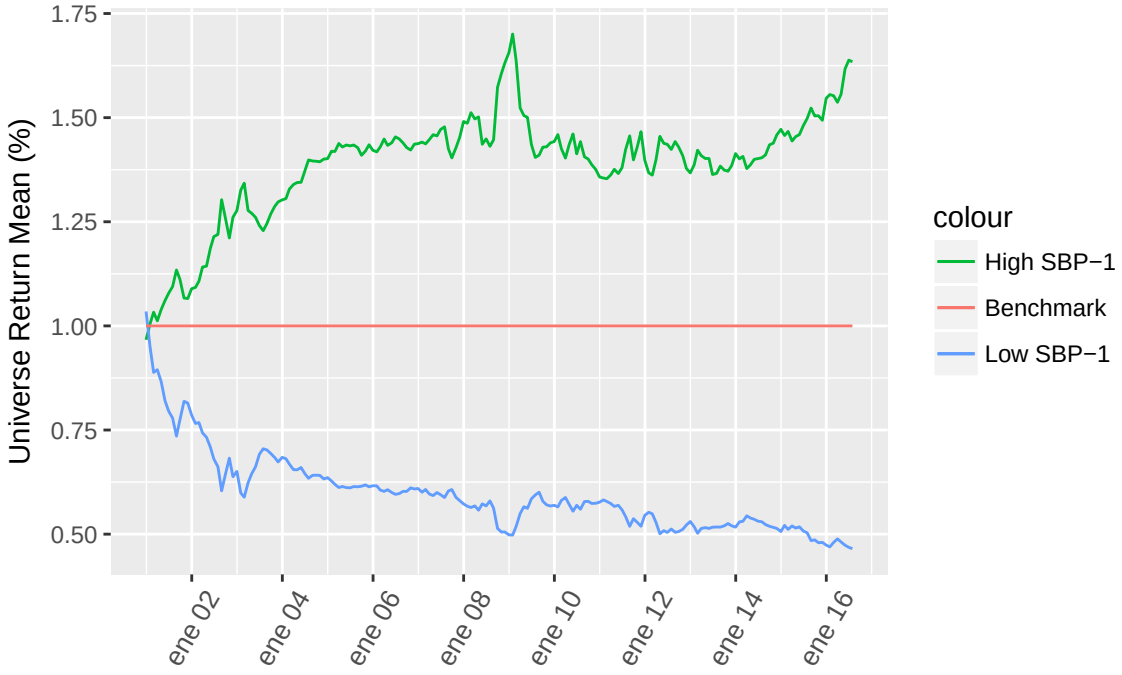
8.3.1 One month training portfolio results

Firstly, we will show the results obtained by the smart beta portfolio under the one-month training period which we will henceforth denote as SBP-1 (Smart Beta Portfolio 1 month). In table 15, the statistical results obtained by both portfolios, we see how the high SMP-1 achieves an average expected return of 2.83% higher than the benchmark, an annualized volatility of 14.15% and a Sharpe's ratio of 0.20, if we compare it with the fundamental factors we see that only the momemtum factor individually surpasses this performance. With regard to volatility, we have achieved relatively low annualized volatility, only surpassed by the low risk factor, which makes sense to us, and Sharpe's ratio could be said to be high compared to other factors, surpassed only by a few tenths earning yield factor, that is, the results at first sight appear to provide an improvement.

If we compare it with the elementary smart beta portfolio (ESB), in terms of average expected performance we are below the previously mentioned 2.83% versus 5.04% of the ESB, it means an expected return average almost half. But in terms of annualized volatility if we experience a decline of the same even below the ESB but without reaching 12.57% of the low risk portfolio. The Sharpe's ratio is also lower than the ESB.

Finally, if we try to compare the behavior of SBP-1 low depending on the strengths of the fundamental factors, we see how in the first cycle, before the crisis, it captured the overperformance of the low risk and high value factors, then the rebound of the high momemtum factor just when the cycle changed during the crisis, to later set the similiar behavior to the benchmark of the high quality factor and finally to overperformance of the high momemtum factor.

Figure 21: Shows the performance of high and low smart beta portfolios



	High SBP-1	Low SBP-1
Average Excess Return	2.831524	-3.845189
Annualized Volatility (%)	14.1567	0.2000129
Shapes Ratio	0.2000129	-0.1565171
Variation over time	1.711089	0.4377229

Table 15: Performance statistics of Smart beta portfolio one month training (SBP-1)

8.3.2 Three months training portfolio results

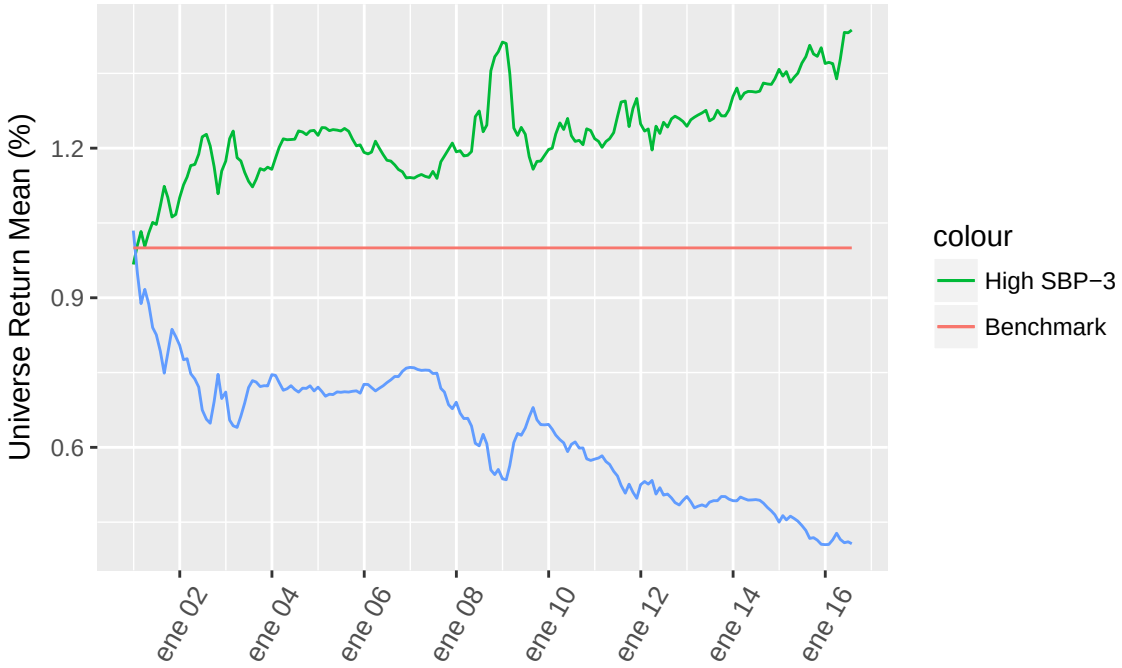
Before presenting the results obtained with the portfolio of high and low smart beta with 3 months of training denoted hereafter as SBP-3, it should be pointed out that the procedure for obtaining the most important variables for each period is the same but taking as training period 3 months instead of one, so we decided not to repeat it in order to speed up the understanding and the visual load in the research.

The results obtained by the high and low SBP-3 can be seen in Table 16 below, the expected average performance of SBP-3 high is 1.88% above the becnhmark, annualized volatility is 15.25% and Sharpe's ratio is 0.12. But how can we verify how good they are? Comparing them with the SBP-1 result. If we look at the statistical results of both tables, we can see how for this variant of the 3-month training the results have worsened at all points, both lower expected performance, higher annualized volatility and a lower Sharpe's ratio, so the training adjustment of 3 months has not given us a better performance than the one month.

However, if we compare it with the factors we see in terms of expected average profitability is not one of the best factors, but in terms of annualized volatility, we managed to reduce it only surpassed by the low risk portfolio.

Finally, trying to give some sense to it is behavior during the time horizon and the possible strengths captured by it, well, we get more or less similar results to SBP-1, where in a first period it seems to capture the overperformance of the low risk Portfolio, while when the cycle turned around due to the crisis also makes sense the high momemtum until the end of the time horizon.

Figure 22: Shows the performance of high and low smart beta portfolios



	High SBP-3	Low SBP-3
Average Excess Return	1.880496	-4.329258
Annualized Volatility (%)	15.25179	24.5535
Shapes Ratio	0.1232968	-0.1763194
Variation over time	1.436896	0.4067427

Table 16: Performance statistics of Smart beta portfolio one month training (SBP-3)

9. Conclusions

To conclude this research, we will provide a final reasoning about the behavior and results obtained under the elementary smart beta and the smart beta one month and it iss variant of three months.

In a first approach to building a factor that brings together the strengths of the fundamental factors low risk, value, momemtum and quality, which we denote as elementary smart beta because the foundation on which we rely for it is construction is to take the simple average of the factors that make it better and worse, we have obtained a very profitable result not only in terms of expected average profitability, and Sharpe's ratio, but in the reduction of annualized volatility, which was therefore successful with the fundamental objective of the research itself.

The next step has been to use more elaborate techniques of advanced machine learning statistics such as random forest. For this, it has been decided to compute a large number of regression trees so as to minimize the %MSE in each period, taking exclusively the most important variable. Under this criterion, we have obtained results that are not as good as the previous ones, say that they do improve or comply with the research's own objective of minimizing variance by maximizing expected returns.

To finish developing a small variant of the previous one, but taking as a training period a somewhat longer period of 3 months, where again even worse or not so good results have been obtained for the period of one month of training, that yes improving even those of each factor individually.

Obviously, you can, moreover, you should go much further, and as a goal for a next research or even the continuation of this should be able to get develop and improve this process even reaching beyond that elementary smart beta that we have obtained here. The calibration of the model is a point where to develop this work, that is to say, which training period is the optimal one at the time of the computation of the regression trees, what is the exact number of trees that manage to minimize the %MSE in each one of those training periods, what number of variables we should extract as important rather than just one or even the average of a number of them the most important and if we use the least significant ones to extract a number of portfolio titles belonging to those variables.

Therefore, the line of research to follow onwards is broad and motivating

9. Bibliografy

References

[1] Markowitz, Harry M. (March 1952), *"Portfolio Selection, The Journal of Finance".*

[2] Willian F. Sharpe: A simplified model for portfolio analysis, *"Portfolio Theory and Capital Markets"*, (McGraw-Hill, 1970);ISBN 0-07-135320-8.

[3] Markowitz, Harry M. (March 1987), *"Mean-variance in portfolio choice and capital Markets"*, Oxford: Blackwell.

[4] Galitz L. (1996), *"Finantial Engineering"*, Prentice Hall.

[5] Knight, Frank. (1921), *"Risk, uncertainly and profit"*, ISBN 978-0-9840614-2-6.

[6] Ruppert, David, *"Statistics and data analysis for financial engineering"*, Springer.

[7] Lintner, John (February 1965), *"The Valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets"*

[8] Jensen, Michael C., Black, Fischer and Scholes, Myron S. (1972), *"The Capital Asset Pricing Model: some empirical tests"*, Praeger Publishers Inc.

[9] Fama, Eugene F. and MacBeth, James D., *"Risk, Return and Equilibrium: Empirical Tests, Journal of Political Economy"*, Vol.81, N3 (May-Jun 1973).

[10] Zhang, Lu. (2005), *"The value premium"*, Journal of finance. Pg:67-103

[11] Frazzini A, Pedersen L H (2010), *"Betting Against Beta"*, NBER Working Paper.

[12] Breiman, L. (1996), *"Bagging Predictors: Machine Learning"*, pp. 123-140.

[13] Financial times handbook of financial engineering, *"Tools and techniques for managing derivaties, options, swaps and risk"*,Prentice Hall.

[14] Baltas, N., Jessop, D., Jones, S., Winter, P., Wu, S., Antrobus, O. and Stoltz, P. (January 2015), *Quantitative Monographs: "Stock selection using Machine Learning"*, UBS Global Research.

[15] Baltas, N., Jessop, D., Jones, C., Lancetti, S., Winter, P. and Holcroft, J (July 2015), *Quantitative Monographs. "Low-Risk Investing: perhaps not everywhere"*, UBS Global Research.

[16] Sefton, J., Jessop, D., De Rossi, G., Zhang, H., Jones, C. (March 2012), *UBS Investment Research. "A fresh look of beta puzzle"*, UBS Global Equity Research.

[17] Baltas, N., Jessop, D., Jones, C., Lancetti, S., Winter, P. and Holcroft, J., Gerken, J., Ivanova, J., Wu, S., Antrobus, O. and Stoltz, P. (September 2016), *Academic Research Monitor. "Combining Smart Beta Factors"*, UBS Global Research.

[18] Jessop, D. and Lancetti, S. (December 2013), *Global Quantitative Research Monographs. "3 Reasons why high-risk underperformed"*, UBS Global Research.

10. Appendix

```
###  Download Data Set Script


# Cargamos ticket STOXX600
stocksLst <- read.csv("E:/Universitat Autónoma Barcelona/Trabajo Fin de Grado/Datos/sp500.csv",
                      header = F, stringsAsFactors = F)
nrow(stocksLst)
stocksLst<-stocksLst[,1]

# Aplicamos el sufijo "NYSE:"
tickets<-sapply(stocksLst, function(x) paste("NYSE:", x), USE.NAMES=FALSE)



###############################################################################
# Load Systematic Investor Toolbox (SIT)
# https://systematicinvestor.wordpress.com/systematic-investor-toolbox/
############################################### ################################
con = gzcon(url('http://www.systematicportfolio.com/sit.gz', 'rb'))
source(con)
close(con)


###############################################################################
# determine date when fundamental data is available
# use 'date preliminary data loaded' when available
# otherwise lag 'quarter end date' 2 months for Q1/2/3 and 3 months for Q4
###############################################################################
date.fund.data <- function(data)
{
  # construct date
  quarter.end.date = as.Date(paste(data['quarter end date',], '/1', sep=''), '%Y/%m/%d')
  quarterly.indicator = data['quarterly indicator',]
  date.preliminary.data.loaded = as.Date(data['date preliminary data loaded',], '%Y-%m-%d') + 1

  months = seq(quarter.end.date[1], tail(quarter.end.date,1)+365, by='1 month')
  index = match(quarter.end.date, months)
  quarter.end.date = months[ iif(quarterly.indicator == '4', index+3, index+2) + 1 ] - 1

  fund.date = date.preliminary.data.loaded
  fund.date[is.na(fund.date)] = quarter.end.date[is.na(fund.date)]

  return(fund.date)
}

#****************************************************************
# Load historical fundamental data
# http://advfn.com/p.php?pid=financials&symbol=NYSE:WMT&mode=quarterly_reports
#****************************************************************
# 1436513D UN
Symbol = data
fund = fund.data(Symbol, 80)


prub<-data.frame(fund)
for(i in 1:length(aaa)){
  prub<-cbind(prub,data.frame(fund.data(aaa[i], 80)))
}
ccc<-row.names(fund)
prub<-cbind(ccc,prub)
ddd<-c("Fundamentals",rep("NYSE:AAPL",81),rep("NYSE:DDD",81),rep("NYSE:FB",81))
prub<-rbind(ddd,prub)
Symbol2 = 'NYSE:AAPL'
fund2 = fund.data(Symbol = Symbol2, 80)
Symbol3 = 'NYSE:DDD'
fund3 = fund.data(Symbol = Symbol3, 80)
Symbol4 = 'NYSE:FB'
fund4 = fund.data(Symbol = Symbol4, 80)
# construct date
fund.date = date.fund.data(fund)


#****************************************************************
# Load historical data
#****************************************************************
load.packages('quantmod')
tickers = 'WMT'
```

```
data <- new.env()
getSymbols(tickers, src = 'yahoo', from = '1980-01-01', env = data, auto.assign = T)
for(i in ls(data)) data[[i]] = adjustOHLC(data[[i]], use.Adjusted=T)

data$WMT = merge(data$WMT, EPS)
# back fill EPS
data$WMT$EPS = ifna.prev(coredata(data$WMT$EPS))

##  Almacenamos los datos en un data set llamado DATA, a partir de ahora los cargaremos directamente.
save(data, file="E:/Universitat Autónoma Barcelona/Trabajo Fin de Grado/datos/SortedData.RData")


## Creación de un vector donde almacenar las posiciones con el cambio de fecha, ya que a partir de ahora
##  necesitamos utilizarlo ya que trabajamos periodo a periodo

# Creamos un vector vacío
v<-c()

# Loop para almacenar dicha posición cuando encuentre diferencias en j y j+1
j=1
for(i in 1:(nrow(data)-1)){
  if(data[i,2]!=data[i+1,2]){
    v[j]<-i+1
    j=j+1
  }

}

# Comprobación
v[1:5]

##  Ahora tenemos un vector con la posición donde cambia el periodo pero necesitamos otro
##  donde se almacene la última fecha para cada periodo

v2=v-1 # Simplemente se obtiene como la resta de la posición anterior al cambio.
v2[1]=1 # indicamos que la primera posición es un uno

##  A partir de este momento almacenamos ambos vectores con el fin de cargarlos
##  cuando se necesite.

save(v, file="E:/Universitat Autónoma Barcelona/Trabajo Fin de Grado/trabajo/v2.RData")
save(v2, file="E:/Universitat Autónoma Barcelona/Trabajo Fin de Grado/trabajo/v2.RData")

## Función base del proyecto

##  A continuación, creamos una función que toma como parámetros de entrada cualquier factor
##  que se desee estudiar: "obj", y además cada uno de los quintiles que se desee calcular y graficar.
##  Por ejemplo, el factor volatility y dibújame el top portfolio (low volatility) contra el bottom
##  portfolio (high volatility).


simulacion<-function(obj,top=F,f2=F,f3=F,f4=F,bottom=F){

  ### Necesitamos la base de datos completa y la base de datos donde solo esta el universe returns
  load(file="E:/Universitat Autónoma Barcelona/Trabajo Fin de Grado/Datos/datos1.RData")
  load(file="E:/Universitat Autónoma Barcelona/Trabajo Fin de Grado/Datos/datos2.RData")
  attach(datos)
  attach(datos2)
  library(data.table)

  ### Amacenamos el nombre de la variable objeto de estudio que ha entrado como parámetro de entrada.
  nombre=deparse(substitute(obj))



  ###   Calcula para cada dia la media del UNIVERSE RETURN y clasifica por fechas
  uniret<-aggregate(datos2[, 2], list(Period..YYYYMMDD.), mean)
  colnames(uniret)<-c("Fecha","Media") #  Renombramos las columnas

  ###   A continuación recogemos los valores de la variable objeto de estudio
  datos3<-data.frame(datos2,obj)
  colnames(datos3)<-c("Fecha","Universe Returns","Factor") #  Renombramos columnas

  ### Problema con los NA's, tenemos que evitarlos a la hora de hacer este calculo (3 NA's)
  ### Metemos un 0 porque los fractiles se mueven en un rango de {1,5} digamos que no los cogeriamos
  ### bajo ningun caso
  datos3$Factor[is.na(datos3$Factor)] <- 0
  summary(datos3)
```

```
### DataSet Quintile 1
F1<-datos3[which(datos3$Factor==1),]

### Agrupamos por fechas y al mismo tiempo se realiza una funcion de media para cada uno de ellos.
F1<-aggregate(F1[, 2], list(F1$Fecha), mean)
colnames(F1)<-c("Fecha","Media") #  Renombramos columnas


### DataSet Quintile 2
F2<-datos3[which(datos3$Factor==2),]

### Agrupamos por fechas y al mismo tiempo se realiza una funcion de media para cada uno de ellos
F2<-aggregate(F2[, 2], list(F2$Fecha), mean)
colnames(F2)<-c("Fecha","Media")#  Renombramos columnas

### DataSet Quintile 3
F3<-datos3[which(datos3$Factor==3),]

##Agrupamos por fechas y al mismo tiempo se realiza una funcion de media para cada uno de ellos

F3<-aggregate(F3[, 2], list(F3$Fecha), mean)
colnames(F3)<-c("Fecha","Media")#  Renombramos columnas

### DataSet Quintile 4
F4<-datos3[which(datos3$Factor==4),]

##Agrupamos por fechas y al mismo tiempo se realiza una funcion de media para cada uno de ellos
 F4<-aggregate(F4[, 2], list(F4$Fecha), mean)
colnames(F4)<-c("Fecha","Media")#  Renombramos columnas

### DataSet Quintile 5
F5<-datos3[which(datos3$Factor==5),]

##Agrupamos por fechas y al mismo tiempo se realiza una funcion de media para cada uno de ellos
F5<-aggregate(F5[, 2], list(F5$Fecha), mean)
colnames(F5)<-c("Fecha","Media")#  Renombramos columnas


### Construyo un data table que reúne todo los anteriores
require(data.table)
datos4<-data.table(F1,F2[,2],F3[,2],F4[,2],F5[,2])
colnames(datos4)<-c("Date","Top","F2","F3","F4","Bottom")

### SIMULACION

### Simulación del BENCHMARK:
BNCHM<-c(100,rep(0,nrow(uniret)))

#En primer lugar la inversion inicial = 100, le debe sumar o restar a esta cantidad,
#el retorno que le corresponda que como es un porcentage...
for(i in 1:nrow(uniret)){
  BNCHM[i+1]<-BNCHM[i]+(BNCHM[i]*uniret[i,2]/100)
}

### Simulación del Quintile 1

SimF1<-c(100,rep(0,nrow(F1)))

for(i in 1:nrow(F1)){
  SimF1[i+1]<-SimF1[i]+(SimF1[i]*F1[i,2]/100)
}


### Simulación del Quintile 2

SimF2<-c(100,rep(0,nrow(F2)))

for(i in 1:nrow(F2)){
  SimF2[i+1]<-SimF2[i]+(SimF2[i]*F2[i,2]/100)
}


### Simulación del Quintile 3

SimF3<-c(100,rep(0,nrow(F3)))

for(i in 1:nrow(F3)){
  SimF3[i+1]<-SimF3[i]+(SimF3[i]*F3[i,2]/100)
}
```

```
### Simulación del Quintile 4

SimF4<-c(100,rep(0,nrow(F4)))

for(i in 1:nrow(F4)){
  SimF4[i+1]<-SimF4[i]+(SimF4[i]*F4[i,2]/100)
}


### Simulación del Quintile 5

SimF5<-c(100,rep(0,nrow(F5)))

for(i in 1:nrow(F5)){
  SimF5[i+1]<-SimF5[i]+(SimF5[i]*F5[i,2]/100)
}


### Una vez contruídas las simulaciones, vamos a graficar los resultados. Mostraremos dos gráficos,
### uno donde se muestran el comportamiento de todos los quintiles de la variable de estudio (portfolios),
### y otro donde ponderaremos contra el benchmark con el fin de observar con detalle cómo se comportan
### ante el mercado.

### Creamos un data set con los resultados de todas las simulaciones
completeGraph<-data.table(date=uniret$Fecha,bench=BNCHM[2:189],top=SimF1[2:189],decil2=SimF2[2:189],
                          decil3=SimF3[2:189],decil4=SimF4[2:189],bottom=SimF5[2:189])
attach(completeGraph)

### Creamos una ventana donde observar ambos gráficos al mismo tiempo.
par(mfrow=c(1,2),bty="n")

### Reordenamos el data set anterior
graf_CTO_long <- melt(completeGraph, id="date")  # convert to long format

### Creamos el primer gráfico

gto<-ggplot(data=graf_CTO_long,
       aes(x=date, y=value, colour=variable)) +
  ggtitle("Figure 10:Shows all momemtum portfolios and benchmark performance")+
  labs(x = "", y = "Expected Return")+
  theme(axis.text.x=element_text( size=10, vjust=0.5))+
  geom_line()+ scale_x_date(date_breaks = "2 years",date_labels = "%b %y")


### Ahora creamos el gráfico donde se ponderan contra el propio benchmark

### Calculamos los resultados para cada portfolio
bnchm<-BNCHM/BNCHM
top<-SimF1/BNCHM
resF2<-SimF2/BNCHM
resF3<-SimF3/BNCHM
resF4<-SimF4/BNCHM
bottom<-SimF5/BNCHM

### De nuevo creamos un data set con todos los resultados para cada simulación
graf<-data.table(date=uniret$Fecha,bench=bnchm[2:189],top=top[2:189],decil2=resF2[2:189],decil3=resF3[2:189],
                 decil4=resF4[2:189],bottom=bottom[2:189])
attach(graf)


### Cogemos el caso donde queremos dar como parámetros de entradas el top and bottom portfolio
if(top==T && bottom==T){

  ###   Creamos el segundo gráfico
  g<- ggplot(graf,aes(x=graf$date,y=top,color="High Momemtum"))+
    geom_line()+
    labs(x = "", y = "Universe Return Mean (%)")
    theme(axis.text.x=element_text(angle=75, size=10, vjust=0.5))

  g<-g+geom_line(data=graf, aes(y = bottom,color="Low Momemtum"))+
    ggtitle("Figure 9:  High and low momemtum portfolios weighted about benchmark")+
    geom_line(data=graf,aes(y=bench,color="Benchmark"))+
    scale_color_discrete(breaks=c("High Momemtum","Benchmark","Low Momemtum"))+
    scale_x_date(date_breaks = "2 years",date_labels = "%b %y")

  ### Devolvemos los dos resultadios de la función (ambos gráficos)
  print(g)
  return(gto)

}}
```

```
###   Como bien hemos dicho esta función es digamos el eje principal del trabajo ya que todos los resultados
###   se obtienen bajo la llamada a la misma.

###   El siguiente paso es normalizar la base de datos bajo el procedimiento explicado en el trabajo


###   Cargamos los Datos (ya ordenados por sector, periodo (mensual) y ISIN)
load(file="E:/Universitat Autónoma Barcelona/Trabajo Fin de Grado/datos/SortedData.RData")

## #  Cargamos los vectores donde se recogen los cambios de fecha (para pasar de un periodo a otro)
load(file="E:/Universitat Autónoma Barcelona/Trabajo Fin de Grado/trabajo/v.RData")
load(file="E:/Universitat Autónoma Barcelona/Trabajo Fin de Grado/trabajo/v2.RData")

###   Creamos un nuevo data set pero esta vez solo con un ID, Fechas, ISIN, Sector y todos los factores
col<-c(1,2,3,4,12,14,16,18,20,22,24)
factores<-data[,col]

###   Creamos una copia de dataset para calcular el z-score de cada factor.
factores1<-factores

###   Vuelvo a crear una copia del data set (aunque no se necesita porque no hemos hecho la media)
factores2<-factores1

###   Cálculo del z-score, pero en función del periodo, es decir, la media y la desviación utilizado es solo por periodo.

##    Especificamos las columnas correspondientes a los fctores
col2<-c(5:11)

###   Loop donde se calculan los coeficientes z-score
i=1
for(i in 1:3479){
  factores2[v[i]:v2[i+1],col2]<-scale(factores2[v[i]:v2[i+1],col2])
  print(i)
}


###   Periodo de comprobación del loop
###   primer periodo:
i=1
factores1[v[i]:v2[i+1],5]
# View(factores1)
factores2[v[i]:v2[i+1],5]
##    Comprobación manual (coinciden los primeros coeficientes)
(0.109096-mean(factores1[v[i]:v2[i+1],5]))/sd(factores1[v[i]:v2[i+1],5])

###   segundo periodo:
i=2
factores1[v[i]:v2[i+1],5]
#View(factores1)
factores2[v[i]:v2[i+1],5]
##    Comprobación manual (coinciden los primeros coeficientes)
(0.104451-mean(factores1[v[i]:v2[i+1],5]))/sd(factores1[v[i]:v2[i+1],5])

###   Ejemplo para el último periodo:
i=3479
factores1[v[i]:v2[i+1],5]
#View(factores1)
factores2[v[i]:v2[i+1],5]
##    Comprobación manual (coinciden los primeros coeficientes)
(0.041651-mean(factores1[v[i]:v2[i+1],5]))/sd(factores1[v[i]:v2[i+1],5])

###   Necesitamos denotar que los valores para las variables beta, volatilidad, netdebt
###   son al contrario que el resto, menores valores, mayor rendimiento.

###   Creamos un data set con los valores normalizados, y damos la vuelta a esas variables
zScore<-factores2
zScore$NET.DEBT..EBITDA.NTM<-zScore[,8]*(-1)
zScore$Beta.6m.Daily<-zScore[,10]*(-1)
zScore$Volatility.1yr<-zScore[,11]*(-1)

###   El siguiente paso es realizar las ponderaciones de cada factor en función de su comportamiento.
###   Explicado en el trabajo.
zScore[,5]<-0.2*zScore[,5]#EY=3.21
zScore[,6]<-+0.15*zScore[,6]#DY=2.04
zScore[,7]<-+0.3*zScore[,7]#MM=4.05
zScore[,8]<-+0.12*zScore[,8]#ND=1.86
zScore[,9]<-+0.05*zScore[,9]#ROE=1.3
zScore[,10]<-+0.08*zScore[,10]#Beta=1.8
zScore[,11]<-+0.1*zScore[,11]#Vol=1.85
```

```
###   Construccción del Elemental Smart beta factor

###   Primera prueba manual solo de los dos primeros periodos
###   Creamos un dataset conde almacenamos el identificador, la fecha, el periodo y la posición
###   donde almacenaremos los valores para el nuevo elemental smart beta factor
fctor<-data.frame(ID=zScore$ID,Fecha=zScore$Period..YYYYMMDD.,NF=zScore$NF,Fractiles=c(rep(0,nrow(zScore))))

###   Calculamos los quintiles, es decir, las probabilidades que hacemos servir para la clasificación
###   de los valores.

###   Como te comentaba utilizo los dos vectores donde tengo almacenado los cambios de fecha: v[1]:v2[2]
a<-quantile(fctor$NF[v[1]:v2[2]],probs = c(0.2,0.4,0.6,0.8),na.rm = T);a

###   Y ahora para las 10 primeras observacines, si es menor que fractil 20, le asigno 5,
###   si está entre el 20 y 40 le asigno un 4, entre en el 4 y 3, le asigno un 3,
###   entre el 2 y el uno le asigno un 2, y si es mayor que el 20 le asigno un 1.
###   Pero de momento solo a las 10 primeras observaciones para ir comprobando.

fctor[which(fctor$NF[1:10] <= (a[1])),4]<-5
fctor[which(fctor$NF[1:10] > a[1] & fctor$NF[1:10] <= a[2]),4]<-4
fctor[which(fctor$NF[1:10] > a[2] & fctor$NF[1:10] <= a[3]),4]<-3
fctor[which(fctor$NF[1:10] > a[3] & fctor$NF[1:10] <= a[4]),4]<-2
fctor[which(fctor$NF[1:10] > a[4]),4]<-1

###   Periodo de comprobación del cálculo

##    Fractiles:
a
##    Asignación:
fctor$NF[1:10]

##    Realmente lo ha hecho bien

###   Realizamos el cálculo para todo el dataset: vamos cogiendo periodo a periodo, y calculamos
###   las probabilidades parra ellos. A continuación cogemos los valores de cada periodo y los
###   clasificamos en función de dichas probabilidades.

for(i in 2:length(v)){

  b<-quantile(fctor$NF[v[i]:v2[i+1]],probs = c(0.2,0.4,0.6,0.8),na.rm = T);b

  fctor[v2[i]+which(fctor$NF[v[i]:v2[i+1]] <= (b[1])),4]<-5
  fctor[v2[i]+which(fctor$NF[v[i]:v2[i+1]] > b[1] & fctor$NF[v[i]:v2[i+1]] <= b[2]),4]<-4
  fctor[v2[i]+which(fctor$NF[v[i]:v2[i+1]] > b[2] & fctor$NF[v[i]:v2[i+1]] <= b[3]),4]<-3
  fctor[v2[i]+which(fctor$NF[v[i]:v2[i+1]] > b[3] & fctor$NF[v[i]:v2[i+1]] <= b[4]),4]<-2
  fctor[v2[i]+which(fctor$NF[v[i]:v2[i+1]] > b[4]),4]<-1

  print(i) #Contador (algo personal que siempre utilizo en loops)
}


###   Periodo de comprobación del cálculo:
#Periodo 1:
quantile(fctor$NF[v[1]:v2[1]],probs = c(0.2,0.4,0.6,0.8),na.rm = T)
fctor$NF[v[1]:v2[2]]
fctor$Fractiles[v[1]:v2[2]]
#Periodo 2:
quantile(fctor$NF[v[2]:v2[3]],probs = c(0.2,0.4,0.6,0.8),na.rm = T)
fctor$NF[v[2]:v2[3]]
fctor$Fractiles[v[2]:v2[3]]
#Periodo 15:
quantile(fctor$NF[v[15]:v2[16]],probs = c(0.2,0.4,0.6,0.8),na.rm = T)
fctor$NF[v[15]:v2[16]]
fctor$Fractiles[v[15]:v2[16]]

###   Podemos comprobar como el looop ha funcionado a la perfección, por lo que el siguiente paso es,
###   almacenar los valores dentro de la base de datos original, y llamar a nuestra función base.

data$NF<-fctor$NF
data$NF.Fractiles<-fctor$Fractiles
attach(data)

###   Llamada a la función aportando como parámetros de entrada, el nombre del factor que es objeto
###   de estudio, y los portfolio que deseeamos comprobar.
simulacion(obj = NF.Fractiles ,top=T,bottom = T)


###   Machine learning and random forest

###   Librerias y/o paquetes utlizados
```

```
library(rpart)
library(lme4)
library(party)
library(partykit)
library(ggplot2)
library(glm2)
library(car)
library(caret)
library(knitr)
library(rpart.plot)
library(randomForest)
library(inTrees)
library(MASS)
library(lmtest)
library(pscl)
library(stargazer)
library(data.table)

###RANDOM FOREST

###   En primer lugar, apuntar que necesitamos los casos completos, es decir, la función de R
###   no acepta NAS

###   Creamos un data set con todos los casos completos del data set original
enteros<-data[complete.cases(data),]

###   Por facilitar la computación extraemos aquellas variables que no son factores quantitativos
columnas<-c(1,3,7,8)
enteros<-enteros[,-columnas]

###   Primer periodo de computación
periodo1<-data[which(data$Period..YYYYMMDD.=="2000-12-29"),]
completeData <- periodo1[complete.cases(periodo1),]
attach(completeData)

###   Calculo del los bosques aleatorios para este periodo
bosque1<-randomForest(Universe.Returns~EY.Med.Ntma+DY.Fwd.NTMA+X12.1m.Price.Mom+NET.DEBT..EBITDA.NTM
                      +ROE.NTMA+Beta.6m.Daily+Volatility.1yr,data=completeData,proximity=TRUE,
                      importance=TRUE,type="classification")

###   Variables importantes
bosque1$importance #Tabla
require(extrafont)
varImpPlot(bosque1,main = "Variable Importance") #Gráfico

##Grafico de MSE VS ntrees para este periodo
MSE<-data.frame(N.trees=seq(1,500,by=1),MSE=bosque1$mse)
grafico<-ggplot(MSE) +
  geom_point(aes(x=N.trees,y=MSE,colour=MSE))

grafico+theme(text = element_text(size=6)) + # Tamaño de fuente del grafico por defecto
  ggtitle("% MSE decrease") + # Título del gráfico
  theme(plot.title = element_text(family="Comic Sans MS",
                                  size=rel(2), #Tamaño relativo de la letra del título
                                  vjust=2, #Justificación vertical, para separarlo del gráfico
                                  hjust=0.5,
                                   face="bold", #Letra negrilla. Otras posibilidades "plain", "italic", "bold" y "bold.italic"
                                  color="black", #Color del texto
                                  lineheight=1.5)) + #Separación entre líneas
  labs(x = "Number of trees",y = "MSE (%)") + # Etiquetas o títulos de los ejes
  #theme(axis.title = element_text(face="italic", colour="brown", size=rel(1.5))) # Tamaño de los títulos de los ejes
  theme(axis.title.x = element_text(face="bold", vjust=-0.5, colour="black", size=rel(1.5))) +
  theme(axis.title.y = element_text(face="bold", vjust=1.5, colour="black", size=rel(1.5)))

###   Ahora almaceno los resultados de las variables importantes
a<-importance(bosque1)
a<-data.frame(a[,1]) #  Como se ha explicado en el trabajo solo la primera

###   Extraigo el nombre del factor mas importante
b<-rownames(a)[apply(a, 2, which.max)];b

###   Segundo periodo de computación (Mismo procedimiento)
periodo2<-data[which(data$Period..YYYYMMDD.=="2001-01-31"),]

completeData <- periodo2[complete.cases(periodo2),]
attach(completeData)
bosque1<-randomForest(Universe.Returns~EY.Med.Ntma+DY.Fwd.NTMA+X12.1m.Price.Mom+NET.DEBT..EBITDA.NTM
                      +ROE.NTMA+Beta.6m.Daily+Volatility.1yr,data=completeData,proximity=TRUE,
                      importance=TRUE,
                      type="classification")
```

```
bosque1$importance
importance(bosque1)
varImpPlot(bosque1,main = "Variable Importance")

#Guardo en A las variables importancia
a<-importance(bosque1)
a<-data.frame(a[,1])
#Extraigo el nombre del factor mas importante
b<-rownames(a)[apply(a, 2, which.max)];b

###   Automatización del proceso para todos y cada uno de los periodos

###   Necesitamos conocer todas las fechas sin repetir que tenemos en el data set, para ello cargamos
###   el vector donde se contienen
load(file="E:/Universitat Autónoma Barcelona/Trabajo Fin de Grado/trabajo/fechas.RData")
fechas[1:4]

### Creación del loop donde se automatiza los procedimientos realizados para cada periodo anteriormente:
i=0

varimp<-c() # Vector donde almacenarempos la variable importante en cada uno de los peroidos
for(i in 1:length(fechas)){

  ### Cogemos fecha a fecha los complete cases
  periodo<-data[which(data$Period..YYYYMMDD.== fechas[i]),]
  completos<-periodo[complete.cases(periodo),]

  ### Calculamos los bosques aleatorios para cada fecha anterior
  bosque1<-randomForest(Universe.Returns~EY.Med.Ntma+DY.Fwd.NTMA+X12.1m.Price.Mom+NET.DEBT..EBITDA.NTM
                        +ROE.NTMA+Beta.6m.Daily+Volatility.1yr,data=completos,proximity=TRUE,  importance=TRUE,
                        type="classification")

  ### Guardamos los resultados (ambas medidas de importanci)
  importancia<-importance(bosque1)
  ### Especificamos que deseamos la INCMSE (explicado trabajo)
  importancia<-data.frame(importancia[,1])
  ### Almacenamos solo el nombre
  varimp[i]<-rownames(importancia)[apply(importancia, 2, which.max)]

  ### Comprobación in-situ del funcionamiento del loop (personal)
  print(varimp[i])
  print(i)
}

### Con el fin de no repetir el proceso las almacenamos
save(varimp1, file="E:/Universitat Autónoma Barcelona/Trabajo Fin de Grado/trabajo/varimp1.RData")

###   A partir de ahora solo tendremos que cargarlas.
load(file="E:/Universitat Autónoma Barcelona/Trabajo Fin de Grado/trabajo/varimp1.RData")

###   Por saber qûe factores y el número han tenido cada una hacemos lo siguiente (curiosidad)
class(varimp1)
prueba<-factor(varimp1)
summary(prueba)

###   Comprobación primeras 5 variables importancia
varimp1[1:5]

###   Ahora hay que coger periodo a periodo los fractiles de las variables mas importantes
###   para cada uno de dichos periodos.

###   Creamos un data set inicial
prueba<-data.frame(data)

###   Creamos la posición donde almacenaremos los resultados para el mismo
prueba$RF<-c(rep(0,nrow(prueba)))

###   Como siempre hacemos, calculamos los dos primeros periodos de manera manual antes
###   de automatizar el procesp.

###   En el nuevo data set creado donde el periodo  coincida con la fecha almacenada en el vector
###   de fechas, almacenas los resultados del periodo anterior de la variable más importante
###   almacenada por el loop anterior.

###   Es decir, en el periodo siguiente, invertimos en función del factor más importante en el periodo
###   inmediatamente anterior.
prueba[which(prueba$Period..YYYYMMDD.==fechas[2]),28]<-prueba[which(prueba$Period..YYYYMMDD.==fechas[1]),varimp1[1]]


###   Automatización del proceso para todos los periodos
```

```
i=2
for(i in 2:length(varimp1)){

    prueba[which(prueba$Period..YYYYMMDD. == (fechas[i])),28]<-prueba[which(prueba$Period..YYYYMMDD.==fechas[i]),varimp1[i-1]]


}

###   Una vez calculados, de nuevo guardamos los resultados en la base de datos original, y hacemos
###   la llamada a la función base.
data$RF.Fractiles<-prueba$RF
attach(data)

####    Llamada a la función base con el parámetro de entrada esta vez del nuevo factor, y los
###   portfolio que deseeamos observar
simulacion(obj = RF.Fractiles ,top=T,bottom = T)

###   Finalmente ampliamos el rolling windows a 3 meses tal y como se explica en el trabajo, es decir,
###   la inversión en este periodo, se hará en función de la variable más importante durante los 3 periodos
###   inmeediantamente anteriores a este

###   Como el metodo de computación es exactamente el mismo, pasamos directamente a la automatizacion
###   del proceso para todos los periodos

###   Cargo las fechas sin repetir
load(file="E:/Universitat Autónoma Barcelona/Trabajo Fin de Grado/trabajo/fechas.RData")

### LooP

i=1
varimp3<-c() #Vector donde almacenaremos la variable importante en los 3 periodos
for(i in 1:(length(fechas)-2)){

  ###   Ahora el periodo de estudio reune 3 periodos en sí
  periodo<-subset(data, data$Period..YYYYMMDD. == fechas[i] | data$Period..YYYYMMDD. == fechas[i+1] |
                    data$Period..YYYYMMDD. == fechas[i+2])

  ###   Calculamos los casos completos
  completos<-periodo[complete.cases(periodo),]

  ###   Calculamos los bosques aleatorios
  bosque1<-randomForest(Universe.Returns~EY.Med.Ntma+DY.Fwd.NTMA+X12.1m.Price.Mom+NET.DEBT..EBITDA.NTM
                        +ROE.NTMA+Beta.6m.Daily+Volatility.1yr,data=completos,proximity=TRUE,
                        type="classification")

  ###   Obtenemos ambas medidas de importancia
  importancia<-importance(bosque1,type = 2)

  ###   Pero solo almacenamos la que nos interesa
  varimp3[i]<-rownames(importancia)[apply(importancia, 2, which.max)]

  ###   Comprobación personal del proceso
  print(i)
}

###   De nuevo una vez calculados procedemos a almacenar dicho vector
save(varimp3, file="E:/Universitat Autónoma Barcelona/Trabajo Fin de Grado/trabajo/varimp3.RData")

###   Después solo debemos cargar dicho vector cuando se necesite con la siguiente función
load(file="E:/Universitat Autónoma Barcelona/Trabajo Fin de Grado/trabajo/varimp3.RData")

###   De nuevo comprobamos la frecuencia de los factores más importantes durante el horizonte
class(varimp3)
proba<-factor(varimp)
summary(proba)

###   Ahora necesitamos decirle que lo que queremos son los quintiles asociados a cada factor importante
###   no su valor. Se diferencian por el nombre

##Decimos que son los fractles no los coeficientes
varimp3<-c(paste(varimp3,"..Fractiles.",sep=""))
varimp3[1:5]

###   Creamos un data set nuevo de prueba
prueba3<-data.frame(data)

###   Creamos la posición donde almacenar los resultados
prueba3$RF3<-c(rep(0,nrow(prueba3)))

###   Computo del primer periodo manualmente
```

```
## Ha metido en los 3 primeros periodos, los correspondientes a la variable mas importante 1 (por tener algo)

prueba3[which(prueba3$Period..YYYYMMDD.==fechas[1]|prueba3$Period..YYYYMMDD.
              ==fechas[1+1]|prueba3$Period..YYYYMMDD. == fechas[1+2]),29]<-
prueba3[which(prueba3$Period..YYYYMMDD.==fechas[1]| prueba3$Period..YYYYMMDD.
              == fechas[1+1]|prueba3$Period..YYYYMMDD. == fechas[1+2]),varimp3[1]]

###   Automatizacion del proceso para cada periodo: Loop
i=3
for(i in 3:length(varimp3)){

  ### Es el mismo proceso anterior, solo que ahora dejamodos periodos por el medio en lugar de uno.
  prueba3[which(prueba3$Period..YYYYMMDD. == (fechas[i+1])),29]<-
    prueba3[which(prueba3$Period..YYYYMMDD. ==fechas[i+1]),varimp3[i-2]]
}

###   Una vez obtenidos los resultados seguimos el procedimiento habitual de guardarlos en el dataset
###   original, y haremos la llamada a la función.
data$RF3.Fractiles<-prueba3$RF3
attach(data)
simulacion(obj = RF3.Fractiles ,top=T,bottom = T)


#######     FIN   ##########
```