MASTER IN BUSINESS ENGINEERING PROFESSIONAL FOCUS IN DATA SCIENCE

Doe the publicly available news allow to refine ESG ratings in order to create better hedged portfolios against climate change news?

de SAUVAGE VERCOUR, Grégoire

Award date:
2022

Awarding institution:
University of Namur

Doe the publicly available news allow to refine ESG ratings in order to create better hedged portfolios against climate change news?

Grégoire de Sauvage Vercour

Directeur: Prof. Béreau

Mémoire présenté en vue de l'obtention du titre de
Master 120 - Ingénieur de Gestion
Finalité Spécialisée en Data Science

ANNÉE ACADÉMIQUE : 2021-2022

Summary/Résumé

Summary

Engle et al. (2019)[15] developed an approach to construct portfolios hedged against the climate change news. Their approach is based on ESG ratings, but these scores are criticised among other things for their lack of transparency. In this paper, we therefore decide to use Data Science through techniques such as word embedding or Sentiment Analysis in order to refine these ratings on the basis of scraped ESG related news. We then adapt the methodology of Engle et al. (2019)[15] by integrating our refined score in attempt to improve the portfolio hedging against climate risk. The hedging of the two approaches are similar, so our methodology does not allow for better results, but this work lays the foundation for further research.

Résumé

Engle et al. (2019)[15] ont élaboré une approche permettant de constituer des portefeuilles couverts contre le risque lié à la fréquence de parution de news sur le changment climatique. Ils ont pour cela utilisé les scores ESG, mais ces derniers sont controversés, surtout pour leur manque de transparence. Dans ce papier nous utilisons donc des techniques de Data Science comme la vectorisation de texte ou l'Analyse de Sentiments pour ajuster les scores ESG existants en analysant des news. Nous intégrons ensuite ces scores ajustés dans la méthodologie d'Engle et al. (2019)[15] pour tenter de l'améliorer. A travers les résultats, nous constatons que les deux approches donnent des taux de couverture similaires. On ne peut donc pas dire que notre méthodologie permet d'obtenir de meilleurs résultats, mais notre travaille pose les bases pour de futures recherches.

Acknowledgments

I would like to thank all the people who helped me during the writing of this thesis. Mrs Sophie Béreau, Professor at UNamur, for her help and advice throughout this process, but also Mrs Camille Baily, Assistant at UNamur, who not only helped me to collect the data I needed, but was also available to answer many of my questions. Finally, Mr Dragomir Radev and Kathan Roberts, respectively Professor and student at Yale University, for answering my questions on their paper and letting me access their code. All of them have greatly contributed to the completion of this project and I wanted to thank them through this little note.

Contents

Doe the publicly available news allow to refine ESG ratings in order to create better hedged portfolios against climate change news?

Grégoire de Sauvage Vercour

August 2022

1 Introduction

Pushed by numerous events, people developed their ethical awareness over time. The environmental context is the main driver of this change, but not the only one. Indeed, the social context is also of growing concern, and the governance of some countries and companies raises several questions too. These concerns form ESG (Environmental, Social and Governance) factors and affect everyone and every sector. Companies have understood this and have started to adapt by incorporating these factors into their strategies. They may be motivated by their morals or by consumer demands, but another reason for firms to become more responsible is the risk posed by bad ESG events and the losses associated with the realisation of these risks. The financial sector has also followed the trend, as can be seen from the increase in the value of sustainable assets under management since 2016 (Statista, 2021[17]). The asset managers that already try to handle a number of risk factors that may affect the market, must now take a new one into account in their management strategy, the ESG risk factor. The rating companies then diversified and established a score to evaluate the sustainability of firms in order to help investors in their quest for responsible investment. This score is called the ESG rating and Engle et al. (2019)[15] went further by formalising an approach based on it to construct portfolio hedging climate risk. They first created climate risk indices on the basis of the frequency of ESG articles in the press and used them as a target to hedge. They then adopted the mimicking portfolio approach with stock ESG ratings to proxy the firm climate risk

exposure. However, as we know from Borms et al. (2021)[8], several papers point out that these scores lack transparency (Berg, Koelbel & Rigobon, 2019; Amel-Zadeh & Serafeim, 2018; and Olmedo, Torres & Izquierdo, 2010)[7, 2, 32]. The same paper also shows that news information can be a useful tool for investors.

Based on these two assumptions, this work attempts to see whether the incorporation of news in ESG ratings allows a better approximation of the exposure to climate risk of funds, and therefore to improve the climate change news hedging. To tackle this question, we adopt the hedged portfolio construction methodology developed by Engle et al. (2019)[15] that we complement by implementing an algorithm based on news analysis to refine the ESG ratings. We retrieve fund and ESG related news on which we perform a sentiment analysis to establish a refining score that we incorporate into baseline ESG scores. We then use this new rating as climate risk factor in the mimicking portfolio approach to construct a portfolio hedged against climate risk indices. We finally perform in- and out-of-sample estimations. The in-sample results show that, depending on the climate risk index used, the portfolio based on the refined score has worse/better performance in terms of climate risk hedging rate than the portfolio constructed with the baseline ESG rating. However, the differences in performance are very small. The out-of-sample results lead us to the same conclusions. We cannot therefore say that the incorporation of news in ESG ratings improves the hedging against the risk induced by climate change news. Nevertheless, the methodology used remains simple and can be improved to check whether these results hold up.

Several papers have already focused on climate risk hedging, automatic news processing to help investors or ESG score improvement, but to our knowledge, our work is the first to combine all of these. We are thus laying the foundations for a new field of literature. Moreover, by pointing out our limitations and suggesting ways to alleviate

them, we hope to help for future research.

The following section defines the important concepts and reviews what has been done in the literature. Section 3 presents the data and their processing. Section 4 details the methodology from the fund selection for the portfolio to the estimates. Section 5 gives the in- and out-of-sample results and interprets them. Section 6 identifies the weaknesses of the study and proposes solutions. Finally, the section 7 draws conclusions.

2 Literature Review

As this work aims to adjust ESG ratings thanks to news articles to see whether it allows to hedge better the climate risk in a financial portfolio, it mixes a number of concepts. It is thus important to explain and define them, but also to give an overview of the different works that have been done in the literature on the four main points discussed in this paper, namely: Climate Risk, ESG ratings, Text Mining and Sentiment Analysis. Then the end of this section provides a tour of the works that have brought together several of these areas and that inspired our work.

2.1 Climate Risk in Finance

The climate risk can be seen as the uncertainty caused by the climate changes, these changes affect companies and therefore they have an influence on the market (Lemoine, 2021)[27]. The impact of the climate on market can be of two kinds, physical or non-physical. The non-physical effect is the one that is caused by market adaptation to climate change while physical climate risks are caused by the direct impact of climate events on companies (Le Guenedal & Roncalli, 2022)[26]. For a non-physical effect example, a local climate change can decrease the returns of local and foreign firms. It can also affect the behavior of individual investors, and they might sell their stocks of high-carbon

emission companies and want to turn to low-carbon emission companies when their local temperature is abnormally high (Choi, Gao & Jiang, 2020)[10]. Another evidence of the impact of the climate events on the stock markets and a good example of physical climate risks is provided by Hong, Li and Xu (2019)[20]. The authors show that high drought risk in a country leads to low returns for the food companies in that country.

The climate risk must therefore be taken into account in the management of a portfolio. Nevertheless, this risk is not always well handled by the managers. They sometimes misestimate the impact of a natural disaster based on their position relative to where the event occurred, and investors closer to the disaster therefore tend to underweight stocks located in the disaster area more than investors further away (Alok, Kumar & Wermers, 2020)[1]. A good starting point to tackle this issue is to measure climate risk, and Le Guenedal and Roncalli (2022)[26] propose an overview of some different existent indicators. First indicator is the carbon footprint that is the companies' share of responsibility for the amount of CO2 produced. The second one is the carbon pathway that represents the variations in the CO2 emission to be achieved in order to meet different targets such as a certain temperature. The measurement of these two indicators is based on the carbon emissions of companies. On the other hand, the two following measures depend on the carbon prices. The first one assesses how much a firm may be affected by a change in carbon price depending on its capital structure. The second one is called the carbon beta and incorporates a risk factor into a return evaluation model to understand the extent to which this factor influences the stock return. There are several method to build this risk factor such as a brown-minus-green factor (Görgen, Nerlinger, & Wilkens, 2020)[18] or climate risk news indices (Engle et al., 2019)[15] that is described later in this study. Then, the penultimate indicator is the climate physical risk as described above and that aims to measure the risks of climate disasters. Finally, They give some other metrics such as ESG ratings or some KPI's.

Despite several leads, an formal approach to enable managers to make rational decisions about climate risk is missing.

2.2 ESG Ratings

With the increasing number of responsible investments, investors needed a metric to assess the sustainability of companies. That's why ESG ratings appeared (Olmedo et al., 2019)[34]. This score is one of the most used indicator to manage the climate risk and to have an idea of the propensity of the firms to be affected by climate change, but not only. More broadly, it provides an indication of whether a company acts according to good environmental, ethical and governance practices. This means that Environmental pillar refers to the way in which the company deals with environmental issues (waste treatment, CO2 emissions, etc.), the Social pillar is linked to the way in which the firm treats people (equality, working conditions, etc.), and the Governance pillar is about the way in which the company is managed (management of fiscal aspects, corruption, executive compensation, etc.). These ratings are calculated by famous rating companies such as Morgan Stanley Capital International (MSCI), Morningstar, Refinitiv or Bloomberg. Each rating agency has its own way to establish a sustainability score and some are really clear about it but for some other agencies there are grey areas in their methodology, especially about the data sources. For instance, Refinitiv uses sustainability reports, news sources and NGO websites (Krappel, Bogun & Borth, 2021)[23]. Meanwhile, Morningstar explains that their score is determined by the exposure of a company to an ESG risk and its ability to manage these risks, and that they rely on data for this (Garz & Volk, 2018)[16], but they don't precise what kind of data and their sources. This comparison is one example, but one can say that the lack of transparency about the methodology is present in every rating company because it is not totally publicly available (Chatterij & Levine, 2006; Olmedo et al., 2010; Saadaoui & Soobaroyen, 2018; Scalet & Kelly, 2018)[9, 32, 36, 37]

and its not the only weakness of ESG ratings. Actually, Delmas & Blass (2010), Olmedo et al. (2013) and Windolph (2011)[14, 33, 40] point out that companies may be very bad in one domain, but it may be compensate by good scores in other domains. Therefore, these points raise the question of whether the scores accurately reflect the responsible nature of the company.

2.3 Text Mining and Sentiment Analysis in Finance

Data Science is now present in a lot of areas (e.g. healthcare (Consoli, Reforgiato Recupero & Petkovic, 2019)[11], supply chain management (Wisetsri et al., 2022)[41] or construction industry (Baduge et al., 2022)) with the Machine Learning (ML) and the Artificial Intelligence (AI), and Finance is one of these fields. For instance, Consoli, Recupero & Saisana (2021)[12] offer a good overview of the use of data science in finance through their book entirely dedicated to the subject. We find also Bartram, Branke and Motahari (2020)[6] that took stock of the AI techniques used in asset management and they emphasize the utility of ML and AI to handle new data formats.

Actually, until now the most used data were numerical data because a computer can process them as they are, but with the evolution of data processing techniques, new data sources such as textual data appeared. Especially, the discipline of text processing and that aims to extract information and discover useful patterns is called Text Mining (TM). To achieve these goals, it mixes techniques of information retrieval, information extraction and Natural Language Processing (NLP) (Hotho, Nürnberger & Paaß, 2005)[21] that is defined as follow by Liddy (2001)[28]: "NLP is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications.". In these applications we find e.g., text classification, sentiment analysis or translation. Moreover, again according to Bartram

et al. (2020)[6] NLP allows to go further in the textual data analysis than the "naive" approaches such as dictionary-based technique, and for all these reasons, very useful insights may be extracted from textual data.

Regarding the application of TM in Finance, Baker, Bloom and Davis (2016)[5] used a simple dictionary-based approach to build economic policy uncertainty indexes ranging from 1985 to 2015 on the basis of the newspaper coverage frequency by selecting articles from multiple sources on the condition that they contain certain terms. In more advanced works, one can find Lavrenko et al. (2000)[25] that developed a system that predict trends on the stock market over a five-hour horizon on the basis of the news released. To achieve that, they collected news and then used a language model trained to be able to say that some words are linked to the appearance of a trend thanks to probabilistic approaches. The system predicts thus the coming trend depending on the words present in the gathered news. Another application of advanced TM in Finance is given by Kraus & Feuerriegel (2017)[24] that performed financial materials analysis thanks to long-short term memory neural networks[1] (LSTM) to forecast the short-term stock prices. Finally, the last example of textual data processing for a financial purpose is the use of NLP made by Luccioni, Baylor & Duchene (2020)[30] to find pertinent answers to questions provided by Task Force on Climate-related Financial Disclosures to companies in the aim of providing good sustainable reports. All these works give a good overview of the TM techniques, the uses of them, and their usefulness in Finance.

Nevertheless, there is a domain of TM widely used in Finance and not mentioned above, that is the Sentiment Analysis (SA). According to Liu (2012)[29] "Sentiment

[1]The LSTM is a variant of classical deep neural networks that is able to treat sequences and is therefore useful for processing textual data because it allows context to be taken into account by keeping into memory some words before and after the current word.

analysis and opinion mining is the field of study that analyzes people's opinions, sentiments, evaluations, attitudes, and emotions from written language. It is one of the most active research areas in natural language processing and is also widely studied in data mining, Web mining, and text mining.". To extract opinion from textual data SA uses TM and NLP techniques, and especially may be achieved through two main approaches: dictionary-based (lexicon-based) or ML. SA with ML employs techniques such as Naïve Bayes Classifier (probabilistic approach) or Support Vector Machines (Gupta et al., 2020)[19]. Regarding the use of sentiment analysis in Finance, Day & Lee (2016)[13] predicted Taiwanese stock price trend by analyzing sentiments in financial news thanks to a mix of lexicon-based and Deep Learning (DL) approaches. Tajmazinani et al. (2022)[38] proposed a similar approach but by blending two data types, i.e. prices indicators and news sentiments. Still in the area of stock price prediction, Jing, Wu & Wang (2021)[22] have also taken textual data and numerical data as input, but, for their part, they extract investors' sentiment from a Chinese stock forum. They also went further in the sense they used only DL. Actually, they extract sentiment thanks to a convolutional neural network[2] and they predicted stock prices with an LSTM. SA is therefore a good way to extract signals from qualitative data which can be used as leverage for all kinds of financial tasks.

2.4 Machine Learning, ESG Ratings and Climate Risk Hedging

While all the above works treats one of the areas covered in this study, the following papers make use of several of these fields and come closer to this work.

First of all, Moniz (2016)[31] demonstrated that ESG news can be predictive of both companies' performance and the returns on their subsequent stock. Actually, he found

[2]Convolutional neural network is a type of deep neural network that is mainly used for image processing or classification. It works by detecting patterns thanks to linear algebra methods. More information on https://www.ibm.com/cloud/learn/convolutional-neural-networks

a significant relation between the fact that a firm is linked to bad ESG news and that earnings surprises occurs. His results also show that abnormal stock returns decrease when the amount of negative news realeased about the firm increases. Therefore, it may be reasonable to assume that taking news information into account leads to a gain of information compared to a purely quantitative analysis. Regarding its methodology, Moniz first gathered non-financial news from the Dow Jones Newswires corpus, and identified Corporate Social Responsibility (CSR) issues in them thanks to the Latent Dirichlet Allocation (LDA) algorithm that aims to extract topics from textual data. The advantage of the LDA compared to term counting methods, such as term frequency-inverse document frequency (*tf-idf*), is that it is able to take into account several topics and it understands synonymy (some words does not appear in a document but their synonyms do) and polysemy (some words have several meanings depending on the context). After identifying three topics which are ethical behaviour, corporate issues and illegal behaviour, Moniz used SA to measure the intensity of media pessimism about a company. He computed two sentiment scores thanks to the negative terms counting and tf-idf. Finally, he performed two regressions, one to see if the negative media coverage of a firm and the intensity of media pessimism about this firm influence the unexpected earnings of it, and the other to check whether these same characteristics have an impact on the excess returns of the company. Thus, in his paper, Moniz combined several NLP techniques to process ESG news articles.

Borms et al. (2021)[8] go further by constructing an index from ESG news articles that anticipates changes in ESG scores provided by Sustainalytics[3]. The ESG ratings are indeed not continuously updated, but at best monthly. To obtain such a tool, the authors proceed as follows, they gather ESG news related to firms of interest, from this news

[3]Sustainalytics is a Morningstar company that establishes the ESG scores.

corpus, they compute a sentiment score for each article, and with this score they build their index. To form their news corpus they first manually instantiate a base of keywords related to the topics of interest, that is each ESG pillar and negative sentiments, and with this set of keywords they query a database to retrieve related documents that are used to estimate a word embedding after being preprocessed[4]. A word embedding model is used to transform words into numerical features (i.e. vectors) to be able to process them. There are several methods for word embedding but Borms et al. choose the one called GloVE that models a word according to the words whith which it co-occurs to take the context into account. Once their model is estimated, they compute a similarity score between the basic set of keywords and the words retrieved from the corpus to keep the 25 most similar words for each keyword in order to expand their ESG and negative sentiments related set of keywords. Then to obtain a news corpus with news related to company and sustainability, the authors query a news database and keep articles that contain at least the name of the company and one keyword from the ESG keyword set. After gathering news articles, they compute sentiment score to see how negative the news releases are and a frequency to realize the amount of released news. Borms et al. (2021)[8] use therefore advanced techniques to retrieve relevant ESG news and extract sentiment score from them. At the same time, they also show that news articles can provide real added value for portfolio managers.

However, the work most related to this study is proposed by Engle et al. (2019)[15]. Actually, they develop a method to hedge portfolio against the climate risk. To do so, they first construct two indices that are related to ESG concerns, they then use these

[4]Preprocessing aims first to split the text into words, then all words are changed to lower case, after the non-informative words such as "a", "this", "and", ... are removed, and finally the words are stemmed (studies becomes studi) or lemmatized (studies becomes study). All these steps to keep only informative words and avoid duplicates.

indices as hedge targets for their portfolio and select assets to take in it, and finally they construct a mimicking portfolio to see how well their approach hedges climate risk. Their two indices are constructed on the basis of news articles collected from two sources that are the Wall Street Journal (WSJ), and Crimson Hexagon's (CH) (a data analytics vendor) that provided to the authors a corpus of news from more than 1000 newspaper suppliers. The WSJ refers to the intensity of the media climate news coverage, so to build it Engle et al. (2019)[15] first gather climate change papers to have a climate change glossary, then they preprocess both the news corpus and the glossary for transforming them into numerical features thanks to the *tf-idf* technique. Finally, they compute a similarity score between the daily news and the ESG glossary, and that give them the WSJ index. For the CH index, now, it is different because CH provide news and statistics according to search terms provided by the client. The authors search term is "climate change" and their index reflects the proportion of negative climate-related news among all news. Then to approximate the asset climate risk exposure in their mimicking portfolio, they use ESG ratings both from MSCI and Sustainalytics to make their estimations with two different sources. They include some other control variables in their regression and then make in- and out-sample fits. Their results show that their approach allows to get a better hedge against climate risk than ETF's. Nevertheless, they point out the fact that the hedging could be better by replacing the ESG scores by another climate risk exposition measure. That's precisely the aim of this paper, that is attempt to improve the climate risk hedging obtained by Engle et al. (2019)[15] by adjusting ESG scores through the incorporation of qualitative information contained in the news.

For this purpose, Roberts, Radev & Kelly (2019)[35] provide avenues to explore through their approach of establishing ESG scores by analysing 10k reports[5] using NLP

[5]The 10k reports are reports mandated by the U.S. Securities and Exchange Commission (SEC) that

techniques. They first collect reporting standards from the Global Reporting Inintiative[6] that give guidelines to organisations for reporting their ESG impact. The authors extract then vectors from both the 10k reports and the GRI standards thanks to the *tf-idf* method. After vectorisation their system then calculates a similarity score between each sentence in the reports and each sentence in the standards, and for similar sentences it uses sentiment analysis to derive a positive or negative score. Therefore, after aggregating the scores for the whole document they obtain a score for the company's performance against each GRI ESG standard. In this paper, we adapt this approach to analyse news and derive a score to adjust existing ESG ratings.

3 Data

This work is based on multiple data sources. This section is dedicated to the description of these data but also to the way we collect and process them to perform our analysis. Since our analysis isn't performed on stocks but well on funds, the first subsection is about them. The second subsection describes all the textual data used in the paper. In this category, we speak about the news database that is used for the sentiment analysis and the development of the new ESG score. This subsection explains also the sustainable vocabulary glossaries which allow to filter out the news to keep only ESG news, and the sentiment dictionary which serves as a basis for sentiment analysis. Finally, the last data we present in the section are those used to estimate the hedged portfolio.

provide details of a company's financial performance.

[6] "The Global Reporting Initiative (GRI) is an independent international standard-setting body for sustainability performance and disclosure of information by companies, government and non-governmental organisations." https://fr.wikipedia.org/wiki/Global_Reporting_Initiative

3.1 ESG Fund Data

Since the objective of this paper is to refine ESG scores to use them as a proxy for climate risk exposure, it is not enough to have one-off data, but over a period of time. To determine this period we based ourselves on the work of Engle et al. (2019)[15] as they provide two climate risk indices from 1984 to 2017 and from 2008 to 2018, that we use in this work. Therefore the only data we have access to covered part of these periods is quarterly data on 2042 U.S. equity funds from Morningstar[7]. These data were retrieved and provided to us by Baily and Gnabo (2022)[4], start in the first quarter of 2012 and run until the fourth quarter of 2018. Nevertheless, the coverage of ESG scores is too low to be exploited before the first quarter of 2013, so we keep the data from the first exploitable quarter.

The database contain panel data and has 28 variables but we need only the following:

- Sustainability Score: this score is calculated on the basis of two components, the portfolio ESG score and a Controversy score. The first one is first established by Sustainalytics[8] for each firm in the portfolio. To do so they assess the ability of companies to face ESG issues and they normalize the score relative to the other firms in the same industry. Morningstar then aggregates these scores at the portfolio level by weighting the scores by the weight of the assets in the portfolio. Finally, they deduce a controversy score from this score. The controversy score is calculated by Sustainalytics on the basis of ESG-related incidents for each asset in the portfolio and then as for the ESG score it is aggregated by Morningstar at the portfolio level.

- Id of the funds: the unique fund id's that allows to link the funds with their related

[7]Morningstar is one of the biggest rating agency.
[8]Sustainalytics is a subsidiary of Morningstar, dedicated to the treatment of sustainable data.

data in other databases

- Quarter: indicate the quarter of the data row

We add the excess return by fund and by quarter to the 28 variables because it was not there. The excess return is the difference between the return and the risk free rate.

In addition of these data on funds, we have the investments of funds by quarter as well as a file with the market names of the companies present in funds.

By exploring the data we notice that 45 funds are not present for every quarter in fund investments data and 120 have missing quarterly data in ESG fund data. We decide to remove these 165 funds from the the list of candidates for inclusion in the hedged portfolio. Indeed, as we make estimates for each quarter, we need continuity in the data.

3.2 Textual Data

First of all, the news. It is required to have data for the same period as the ESG Data. Therefore we need to retrieve news archives. To do so we use a python package called *news-please*[9] that allows to scrap news on a website called Common Crawl[10]. This website stores billions of web archives including newspaper archives. It allowed us to gather 60398 news from January 2013 to December 2018. These news come from Wall Street Journal, Washington Post, New York Times, Financial Times, NBC News, Bloomberg and Reuters. We took newspapers providers that are likely to reach investors and that are known to be serious.

[9]The package is available on Github: https://github.com/fhamborg/news-please#news-archive-from-commoncrawlorg

[10]https://commoncrawl.org

Part of the filtering of these news is done thanks to 19 glossaries collected on the internet. For most of them they are retrieved from web sources thanks to web scraping techniques, and the others from PDF documents thanks to the python package *PyMuPDF*[11]. Sources include the Government of Canada, United Nations Climate Change, BBC, and others. The full list of these sources can be found in Appendix A.1. These glossaries allow to constitute an ESG vocabulary corpus that is used to retrieve the related articles.

Finally, to perform the SA we use the Harvard IV-4 Dictionnary (HIV) that contains terms and the sentiment associated with them. It is also used by Moniz (2016)[31] and Roberts et al. (2019)[35]. It was in fact the latter who provided it to us.

3.3 Climate Risk Indices and Fama-French Factors

The Climate Risk Indices have been developed by Engle et al. (2019)[15]. There are two indices, one based on the Wall Street Journal (Wall Street Journal Index, WSJI) and the other built by Crimson Hexagon (Crimson Hexagon Index, CHI)[12]. These indices represent the coverage of the climate change in the media between 1984 and 2018 for the first one and between 2008 and 2018 for the second one, and give thus an idea of the amount of occurrences of climate events. The two indicators differ in the sens that the WSJ index takes into account all the news whether it is good or bad, while the CH index only incorporates bad news and is based on multiple sources. These measures are used as target for the hedge portfolio. To do so, we follow Egle et al. (2019)[15] and we take as target the residuals from an AR(1) model perform on each of the two indices. We then aggregate these values at a quarterly level. Therefore, the hedge target is the innovation in each index between two quarter. All these data are provided by Engle et al (2019)[15].

[11] https://pymupdf.readthedocs.io/en/latest/index.html

[12] Crimson Hexagon was a company of data analysis. They have merged with their rival and are now called Brandwatch

Moreover, in their paper, they use three additional risk factor to ensure the portfolio only hedges climate risk. One accounting for the size of the firms, one for their value (based on book-to-market ratio), and one for their market value. However, these factors are calculated on the basis of firm level historical data that we don't have. We therefore replace them by Fama-French factors from the 3-factor model. They are monthly and are retrieved from the Kenneth French's website[13]. We thus aggregate them into quarterly data to match the funds data. In these 3 factors we find the Market (MKT) factor that represents the excess return of the market portfolio, the Small-Minus-Big (SMB) factor that is the difference between the expected return of a low-capitalisation portfolio and of a high-capitalisation portfolio, and finally the High-Minus-Low (HML) factor that is the difference between expected return of a portfolio with high book-to-market ratio and with a low book-to-market ratio.

4 Methodology

The goal of this paper is to check whether the use of news articles allows to refine ESG ratings in order to obtain a better hedge against climate risk for a portfolio. The first step is to chose the target to hedge and in this paper it is the two indices constructed by Engle et al. (2019)[15] as their two indices are indicators of climate risk and as our aim is to compare our results with their. We thus also follow their methodology to construct the hedged portfolio. In their methodology, they adopt the mimicking portfolio approach to hedge the climate risk target CRT_t, that gave them the following regression equation:

$$CRT_t = \xi + w_{SUS} Z_{t-1}^{SUS} r_t + w_{SIZE} Z_{t-1}^{SIZE} r_t + w_{HML} Z_{t-1}^{HML} r_t + w_{MKT} Z_{t-1}^{MKT} r_t + e_t \quad (1)$$

[13] http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html

where $w_{SUS}, w_{SIZE}, w_{HML}, w_{MKT}$ are the weights of the funds in the mimicking portfolio, r_t are the excess returns of the funds over the risk-free rate, Z_{t-1}^{SIZE} is a vector of standardized market value of funds, Z_{t-1}^{HML} is a vector of standardized book-to-market ratio of the funds, Z_{t-1}^{MKT} is a vector of the fund share of total market value, e_t is the error vector, and Z_{t-1}^{SUS} is the vector of ESG ratings that stands for the proxy of climate risk exposure of the funds, it will be designated as the climate risk factor in the rest of the paper. The three additional risk factors are used as control factors to ensure to hedge the climate risk because there could be a correlation between these three factors and the climate risk factor. As explained in subsection 3.3, since we do not have access to market data at fund level, we replace the three additional risk factors by the Fama-French factors. In our work, regression 1 becomes thus

$$CRT_t = \xi + w_{SUS} Z_{t-1}^{SUS} r_t + w_{SIZE} Z_{t-1}^{SIZE} + w_{HML} Z_{t-1}^{HML} + w_{MKT} Z_{t-1}^{MKT} + e_t \qquad (2)$$

where Z_{t-1}^{SIZE}, Z_{t-1}^{HML}, Z_{t-1}^{MKT}, becomes vectors that contains respectively the SMB, the HML and the MKT factor of the Fama-French 3-factor model.

The mimicking portfolio approach is interesting because it allows to construct a portfolio that follows at most the variations of a target indicator. It is therefore adequate for the construction of a hedging portfolio because if it is positively correlated with the climate risk index it will produce higher excess returns when the climate risk increase.

Also, to try to improve the climate risk hedging our approach is to refine the ESG scores of the fund thanks to SA on news articles related to stocks in the fund. With this method, it is hoped to obtain a proxy for climate risk exposure that better reflects actual exposure than conventional ESG ratings. To do so, we first have to select the funds to put in the portfolio and collect all the names of the stocks present in the funds for the purpose of retrieving related news. Once the news has been gathered, we filter them out

to keep only the ESG related ones. We perform then the SA that allows to compute a score for each news that is aggregated at stock level before being aggregated at the fund level. These scores are then added or deduced from the base ESG scores. Finally, the updated ESG rating is used in regression 2 and we estimate it to get the hedging capacity of the portfolio. This methodology is detailed in the following subsections.

4.1 Fund Selection

To select the funds for the portfolio, we proceed as follows. We group funds by their Id and take the average TNA and ESG rating over all quarters. Then we detect outliers in terms of mean ESG score over the whole period because they could influence the further estimates. In this set of funds there is no abnormal score, the list of candidates therefore remains unchanged after this stage. After that, to maximise the chances of finding related news we keep only the top quartile of the largest funds in terms of TNA because our news base is not so large. To check that this does not influence our results, we test later by taking all the funds. We also withdraw funds for which we do not have the investments for each quarter or where ESG data are missing for some quarters as explained in subsection 3.1.Code used for Fund Selection can be found in Appendix A.2.

4.2 Stock Names Retrieval

There are a number of issues to be raised when collecting the names of assets in the funds. Actually, fund investments are not constant, so the list of stocks by fund change every quarter. That is why we collect this list from the fund investments files again every quarter and we drop duplicates to get a list of unique stocks. From this list we retrieve the legal names of the firms (e.g. Apple Inc, Tesla Inc or Exxon Mobil Corp). Nevertheless, in newspapers it is rare to find the legal names of companies due to the suffixes such as "Inc", "Ltd", "SA", etc. It is therefore required to clean the names to optimize the

research of related news as explained by Borms et al. (2021)[8], so we remove the suffixes from legal names to have both the "common" and the legal name. Code used for Stock Names Retrieval can be found in Appendix A.3.

4.3 News Gathering and Filtering

First of all it is required to gather the news to build up a newsbase. To do so, one solution could be to do web scraping. Nevertheless, we need articles from 2013 to 2018 and the sites that offer them are only available with a paid subscription, so we need to find another solution. As described in section 3.2, we use the *news-please* python package to grab news on Common Crawl. The resultant newsbase contains 60398 articles in English.

Then, for each quarter we retrieve the appropriate news before applying filters. Actually, since our goal is to refine the ESG ratings, we chose to keep only stocks and ESG related news, and calculate the amount of good and bad news to establish a score. In other words, we want to see if companies have experienced good/bad ESG events that would have escaped the rating companies. As the sustainable theme may be large and as the process to detect ESG related news is more cumbersome than the one to retrieve companies related news, it is more convenient to start by filtering the newsbase by firm name in order to reduce its size. The method we adopt here is similar to a "naive" search engine. Indeed, we do a simple search in the texts of the articles to see if either the legal name of the company or its name cleaned of its suffix is present. Borms et al. (2021)[8] used a similar approach. Despite its simplicity, this method allows a good number of search results to be obtained because it gives more results than only searching for the legal name. However, there is also more noise since some company names become common nouns without their suffix. For example, "Apple Inc" becomes "Apple" and although the system is case sensitive, sometimes the common nouns begin a sentence. Nevertheless, in our case it is more interesting to have more relevant news even if it means having some

waste because there is a second filter to recover ESG related news which will eliminate some of these irrelevant news.

This ESG filter is built as follows. We were inspired by Roberts et al. (2019)[35] who calculated a similarity score between the Global Reporting Initiative (GRI) Standards[14] sentences and those of the 10-k filings[15] reports to perform SA on similar sentences and see how frequently the reports speak positively or negatively about ESG subjects. We also were inspired by Engle et al. (2019)[15] that establish a similarity measure too between some sustainable glossaries and news articles to build up their two climate risk indices. This measure used in the two papers is called cosine similarity and it gives an idea of the distance between two vectors by calculating the cosine of the angle between them.

$$cos\theta = \frac{A.B}{\|A\|\|B\|} \tag{3}$$

This is the cosine of the θ angle between vector A and B, and this value is between $[-1, 1]$. -1 means the two vector are opposite, 0 means they are independent and 1 they are the same. So, to compute this value it is required to have vectors and thus to transform textual data into vectors. For our part, we calculate the similarity between a corpus of several ESG glossaries and our news articles, and to transform them into vectors we use the *tf-idf* method as in the two works cited above. First of all, tf means "Term Frequency" and is computed as follows:

$$tf = \frac{n_{i,j}}{N} \tag{4}$$

[14]GRI Standards provide guidelines to make good reports about sustainable topics

[15]According to Investopedia, "A 10-K is a comprehensive report filed annually by a publicly-traded company about its financial performance and is required by the U.S. Securities and Exchange Commission (SEC)."

$n_{i,j}$ is the number of occurrences of word i in document j and N is the total number of word in document. Secondly, Inverse Document Frequency (*idf*) is more complicated to understand, it indicates how representative a word is of the topic matter of a document. Actually, it is calculated thanks to this equation:

$$idf = \log \frac{D}{nd_i} \tag{5}$$

where D is the total number of document in the Corpus and nd_i is the number of documents containing the word i. Therefore, the more a word is present in texts, the more it is common and thus the less it is informative on the nature of a document. The *tf-idf* is simply the product between tf and *idf*, so we can say that the more a word is representative of the nature of a document, the higher its *tf-idf* score will be. Once the *tf-idf* has been calculated, to vectorise the document, simply replace each word with its score and it gives the document vector.

However, before vectorising the corpus, it is required to preprocess it. There are five steps in preprocessing. The tokenization, that means splitting the text into words. Then the Lower casing, that is putting all words in lower case. After that, the Stop Words Removal which consists of deleting all words like "the", "and", ... because they have no informative capacity about the nature of the text. Afterwards, the Stemming that reduces word to its root form (e.g., "machine" becomes "machin", "learning" becomes "learn"). Finally, the Lemmatization is an alternative to Stemming and it transforms words into their word of origin (e.g., "machine" stays "macine", "learning" becomes "learn").

In our paper, we build the documentbase that serves as the basis for practicing *tf-idf* by gathering the newsbase and the sustainable glossaries corpus. We then perform the preprocessing and the *tf-idf* transformation thanks to the *TfidfVectorizer* Python module

implemented by Scikit Learn[16]. That allows us to calculate the cosine similarity between the glossaries and each news and we fix a threshold above which the news are considered ESG related. To fix our threshold we take the median of the similarity matrix, that is the matrix $S(n \text{ x } n)$ with n the number of documents in the aggregated corpus and where $s_{i,j}$ is the cosine similarity between text i and j.

In addition, this matrix allows us to detect near-duplicates news, i.e. news that are not entirely the same but tell the same story. Actually, since the articles come from different sources and the same information may be used by several media, it is important to check if we don't have near-duplicates so that the same information is not taken into account several times in the calculation of the new ESG score. As the level of similarity needs to be higher than simply detecting sustainable news, we set this time an arbitrary threshold based on our observations. This threshold is set at 0.5 and we remove news whose similarity score with another news exceeds 0.5. This is the last step of the news processing part. All the code used for News Gathering and Filtering can be found in Appendix A.4.

4.4 Sentiment Analysis

The SA is the beginning of the scoring phase. Actually, a SA is needed to know if the events reported in the news collected are positive or negative for the company and therefore it is needed to know if the score must be added to the initial ESG rating or deduced from it. To establish the refining score for each news, we adapt the methodology of Roberts et al. (2019)[35]. In their study, they count the frequency of positive and negative sentences in the document on the basis of the HIV to have a positive and a

[16] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

negative score. They combine this sentiment with a similarity score between each sentence and each GRI reporting standard to determine to which ESG category the sentence is related. In our paper we do not distinguish between the different ESG categories, so we just use the sentiment score. For each news, we proceed sentence by sentence and calculate a positive and a negative sentiment score. The sentence scores are calculated in the following way:

$$PS_s = \sum_{pw=1}^{PW} 1 \tag{6}$$

for the positive score of sentence s and for its negative score,

$$NS_s = \sum_{nw=1}^{NW} 1 \tag{7}$$

where PW and NW are respectively the total number of positive and negative words in the sentence, and we determine whether a word is positive or negative using the HIV. We calculate then the total positive and negative scores for the whole document with

$$PS_d = \frac{\sum_{p=1}^{P} PS_p}{S_d} \tag{8}$$

$$NS_d = \frac{\sum_{n=1}^{N} NS_n}{S_d} \tag{9}$$

where P and N are respectively the total number of positive and negative sentences in document d, and S_d is the total number of sentences in d. Finally, we calculate a global score that is simply the difference between the positive and negative score of the document.

$$GS_d = PS_d - NS_d \tag{10}$$

Therefore, we have a global score for each news which we will use to calculate the fund refining score. The code used for Sentiment Analysis is available in Appendix A.5.

4.5 Score Refining

Before calculating the refining score for the funds, we need to aggregate the news global score at the stock level. For each stock, we sum thus the global scores of the news related to it.

$$SS_s = \sum_{d=1}^{D} GS_d \tag{11}$$

where D is the total number of news related to the stock s and GS_d is the global score of the news d. That gives us a score for each stock. Then we aggregate it at the fund level. To do so, we take the investments of each fund to have the weight of each stock in it, and we use this to weight the stock score in the fund score. That gives us the following refining score for the fund f:

$$RS_f = \sum_{s=1}^{S} w_{s,f} SS_s \tag{12}$$

where S is the total number of stocks in fund f, $w_{s,f}$ is the weight of stock s in f, and SS_s is the score of stock s. We then standardise this score to avoid having too high values and ending up with negative final ESG scores

$$SRS_f = \frac{RS_f - \mu_F}{\sigma_F} \tag{13}$$

where μ_F is the mean refining score of the fund portfolio and σ_F is its standard deviation. Now we have a standardised refining score for each fund we can adding it to the baseline ESG rating

$$ESG_f^{new} = ESG_f + SRS_f \tag{14}$$

The baseline ESG score of the fund is thus increased or decreased depending on the sign of SRS_f. With this new ESG rating we can launch the estimate phase to compare hedging performance against the climate risk of a portfolio constructed using the baseline ESG score as climate risk factor against one constructed using the refined ESG score.

The steps described in subsections 4.2 to 4.5 are performed for each quarter as fund investments and related-stocks articles vary every quarter.

The code used for Score Refining can be found in Appendix A.6.

4.6 Estimation

For the estimates as explained in the subsection 3.1 we start from the database obtained through the previous processing, we add excess returns and the three Fama-French factors, and we construct the climate risk factor. Actually, we make two different climate risk factors based on the approach of Engle et al. (2019)[15] to compare their results with ours. These are:

- Absolute score: for each quarter we demean the ESG rating of each firm and then take its absolute value ($Z_{t-1}^{SUS\_A}$)
- Ranked score: for each quarter we rank the companies by their ESG rating and then normalise their ranking to be in the range -0.5 to 0.5 ($Z_{t-1}^{SUS\_R}$)

We then break down these two factors by also calculating them on the basis of our refined ESG score, and that gives the "refined absolute score" ($RZ_{t-1}^{SUS\_A}$) and the "refined ranked score" ($RZ_{t-1}^{SUS\_R}$). For the climate risk targets as explained in subsection 3.3, we use residuals of AR(1) models applied on respectively the WSJI (WSJI_AR1) and the CHI (CHI_AR1). Therefore, we have all the needed variables to estimate regression 2. We perform eight different estimates, i.e. four by climate risk target. Indeed, one is needed per climate risk factor. That gives us the following eight regressions to estimate:

$$WSJI\_AR1_t = \xi + w_{SUS} Z_{t-1}^{SUS\_A} r_t + w_{SIZE} Z_{t-1}^{SIZE} r_t + w_{HML} Z_{t-1}^{HML} r_t + w_{MKT} Z_{t-1}^{MKT} r_t + e_t \quad (15)$$

$$WSJI\_AR1_t = \xi + w_{SUS} Z_{t-1}^{SUS\_R} r_t + w_{SIZE} Z_{t-1}^{SIZE} r_t + w_{HML} Z_{t-1}^{HML} r_t + w_{MKT} Z_{t-1}^{MKT} r_t + e_t \quad (16)$$

$$WSJI\_AR1_t = \xi + w_{SUS} RZ_{t-1}^{SUS\_A} r_t + w_{SIZE} Z_{t-1}^{SIZE} r_t + w_{HML} Z_{t-1}^{HML} r_t + w_{MKT} Z_{t-1}^{MKT} r_t + e_t \quad (17)$$

$$WSJI\_AR1_t = \xi + w_{SUS} RZ_{t-1}^{SUS\_R} r_t + w_{SIZE} Z_{t-1}^{SIZE} r_t + w_{HML} Z_{t-1}^{HML} r_t + w_{MKT} Z_{t-1}^{MKT} r_t + e_t \quad (18)$$

$$CHI\_AR1_t = \xi + w_{SUS} Z_{t-1}^{SUS\_A} r_t + w_{SIZE} Z_{t-1}^{SIZE} r_t + w_{HML} Z_{t-1}^{HML} r_t + w_{MKT} Z_{t-1}^{MKT} r_t + e_t \quad (19)$$

$$CHI\_AR1_t = \xi + w_{SUS} Z_{t-1}^{SUS\_R} r_t + w_{SIZE} Z_{t-1}^{SIZE} r_t + w_{HML} Z_{t-1}^{HML} r_t + w_{MKT} Z_{t-1}^{MKT} r_t + e_t \quad (20)$$

$$CHI\_AR1_t = \xi + w_{SUS} RZ_{t-1}^{SUS\_A} r_t + w_{SIZE} Z_{t-1}^{SIZE} r_t + w_{HML} Z_{t-1}^{HML} r_t + w_{MKT} Z_{t-1}^{MKT} r_t + e_t \quad (21)$$

$$CHI\_AR1_t = \xi + w_{SUS} RZ_{t-1}^{SUS\_R} r_t + w_{SIZE} Z_{t-1}^{SIZE} r_t + w_{HML} Z_{t-1}^{HML} r_t + w_{MKT} Z_{t-1}^{MKT} r_t + e_t \quad (22)$$

The code used for Estimates is available in Appendix A.8

5 Results

This section is dedicated to the discussion of the results obtained through the different phases described above and to the comparison of the approach with the baseline ESG rating and the one with the refined ESG rating.

First of all, the fund selection phase described in subsection 4.1 allows to restrict the number of funds used in the portfolio from 2042 to 347.

For these 347 funds we retrieve on average 414 news per quarter and this results in an average of 115 fund score being updated. A first observation we can make is that few funds are updated on average per quarter. These results depend on the method used, it would therefore be interesting to try with other methods for the document vectorisation such as Word2Vec or the use of Topic Modelling based on deep learning models to retrieve ESG-related articles. These alternatives and the reasons for not using them here are detailed in the next section.

This obtained list of funds is thus used to construct the different hedge portfolios.

5.1 In-Sample Estimates

For the in-sample estimates we run all the regressions presented in subsection 4.6 over the period for which we have both ESG data and data on the two climate risk indices. Therefore, the estimation period for portfolios hedging the WSJI is from the second quarter of 2013 to the second of 2017, and that for portfolios hedging the CHI is also from the second quarter of 2013 but run until the first one of 2018. Table 1 shows the results of regressions 15, 16, 17, 18. The coefficient of the climate risk factor in regression 15 indicates a positive and significant correlation with the WSJ climate risk index, that means that portfolios with more funds performing better in ESG factors has higher excess returns. The portfolio constructed on the basis of the refined ESG ratings has also a positive and significant relationship with the climate risk index but it performs worse than the first one because its coefficient is less important ($0.001 > 0.00093$). The portfolio obtained from the regression 17 is also less efficient in the hedging of variation of the WSJI as shown by its R^2 which is 48.06% against 48.18% for the regression 15. Concerning the two portfolios based on the ESG ranked score, i.e. regression 16 and 18, they both have an insignificant coefficient between their climate risk factor and the climate risk target. It would therefore be incorrect to infer any conclusions. However, their R^2 are equal to within 0.01%, so we can say that they hedge climate risk as well as each other.

Table 2 shows the results of regressions 19, 20, 21, 22 that hedge the measure of innovations of the CHI. Such as for the previous regressions, the two ones that capture relationships between the absolute score and CHI_AR1 have positive and significant coefficient for the climate risk factor. Therefore these results also demonstrate that portfolios investing in ESG funds have higher excess returns. However, here the difference between the portfolio based on the baseline ESG score and the one based on the refined score is less marked, but still to the advantage of the former. In terms of hedging of the variation

Table 1: Regression from 2013 Q2 to 2017 Q2 on WSJI_AR1

	Reg. 14	Reg. 15	Reg. 16	Reg. 17
$Z_{t-1}^{SUS\_A} r_t$	0.001^{***}			
	(0.000062)			
$Z_{t-1}^{SUS\_R} r_t$		-0.00068		
		(0.00043)		
$RZ_{t-1}^{SUS\_A} r_t$			0.00093^{***}	
			(0.000057)	
$RZ_{t-1}^{SUS\_R} r_t$				-0.00076
				(0.00044)
$Z_{t-1}^{SIZE} r_t$	-0.00011^{***}	-0.00011^{***}	-0.00011^{***}	-0.00011^{***}
	(0.0000025)	(0.0000026)	(0.0000025)	(0.0000026)
$Z_{t-1}^{HML} r_t$	0.0000066^{***}	0.0000077^{***}	0.0000065^{***}	0.0000078^{***}
	(0.0000016)	(0.0000016)	(0.0000016)	(0.0000016)
$Z_{t-1}^{MKT} r_t$	-0.000062^{***}	-0.000064^{***}	-0.000062^{***}	-0.000064^{***}
	(0.0000022)	(0.0000022)	(0.0000022)	(0.0000022)
Constant	0.00078^{***}	0.00079^{***}	0.00078^{***}	0.00079^{***}
	(0.000011)	(0.000011)	(0.000011)	(0.000011)
R^2	0.4818	0.4576	0.4806	0.4577

This table shows results from regressions 15, 16, 17 and 18. The dependent variable for each regression is the measure of innovation of the WSJI (WSJI_AR1). Standard errors are in parentheses. *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0$

of the CHI, it is less high than for the WSJI but the portfolio based on baseline ESG rating is still better with 18.67% against 17.24%. When it comes to the portfolios based on the ranked score, the coefficients of their climate risk factor are again insignificant, and the difference between their R^2's is still negligible.

5.2 Out-of-Sample Estimates

For the out-of-sample estimates we also follow the methodology of Engle et al.(2019) [15]. For each quarter q, we estimate the eight regressions presented in subsection 4.6 from the period q_{min}, i.e. the first quarter for which we have both ESG data and climate risk indices data, to q-1. These estimates gives coefficients that we multiply by the various associated risk factors to get the fund weights in the portfolio for quarter q. Thanks to these weights, we then calculate the excess returns of the different portfolios for every

Table 2: Regression from 2013 Q2 to 2018 Q1 on CHI_AR1

	Reg. 18	Reg. 19	Reg. 20	Reg. 21
$Z_{t-1}^{SUS\_A} r_t$	0.00049***			
	(0.000014)			
$Z_{t-1}^{SUS\_R} r_t$		−0.000008		
		(0.00009)		
$RZ_{t-1}^{SUS\_A} r_t$			0.00043***	
			(0.000013)	
$RZ_{t-1}^{SUS\_R} r_t$				−0.00003
				(0.000096)
$Z_{t-1}^{SIZE} r_t$	−0.000018***	−0.000013***	−0.000017***	−0.000013***
	(0.00000098)	(0.000001)	(0.00000098)	(0.000001)
$Z_{t-1}^{HML} r_t$	0.0000066***	0.0000082***	0.0000066***	0.0000082***
	(0.00000065)	(0.000007)	(0.00000065)	(0.000007)
$Z_{t-1}^{MKT} r_t$	0.0000063***	0.0000013	0.0000056***	0.0000013
	(0.00000087)	(0.00000093)	(0.00000088)	(0.00000093)
Constant	−0.000043***	−0.0005***	−0.000043***	−0.0005***
	(0.0000045)	(0.0000048)	(0.0000045)	(0.0000048)
R^2	0.1867	0.04338	0.1724	0.04339

This table shows results from regressions 19, 20, 21 and 22. The dependent variable for each regression is the measure of innovation for the CHI (CHI_AR1). Standard errors are in parentheses. *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0$

quarter q. Finally, we establish the correlations between these returns and the climate risk indices that are shown in Table 3. The first thing that can be noticed is that the portfolios are negatively linked to the CHI, so they have less important returns during periods when negative climate news release. Moreover these correlations are relatively strong, so the decrease in excess returns of the portfolios are also relatively intense when there are positive innovations in CHI. On the other hand, the portfolios are all positively correlated with the WSJI, so they have higher excess returns when there are positive innovations in the index. Nevertheless, intensity of these correlations is less strong than for the CHI.

If we compare the results between the portfolios constructed with the baseline ESG score and the ones constructed on the basis of the refined score, we see that the refined portfolio with absolute score is better in hedging the WSJI because it has a higher positive

correlation ($0.0273 > 0.0266$), but it is worse to hedge the CHI because it has a smaller negative correlation ($-0.5375 < -0.5347$).For the portfolios based on the ranked score, the refined one is better than the baseline one when the CHI is used as climate risk target because it has a higher negative correlation ($-0.1330 > -0.1714$). However this portfolio performs worse than the other to hedge the WSJI.

Table 3: Correlations between the different portfolios and the climate risk indices

	WSJI	CHI
P^{SUS_A}	0.0266	−0.5347
RP^{SUS_A}	0.0273	−0.5375
P^{SUS_R}	0.1182	−0.1714
RP^{SUS_R}	0.1180	−0.1330

This table shows the correlation between portfolios constructed thanks to equations 15, 16, 17, 18, 19, 20, 21 and 22 and their respective climate risk target.

From the previous two sections, it can be seen that the approaches based on the two different ESG scores fare more or less well in relation to each other. Nevertheless, the differences in performance are small and this may be because the adjustments to the scores are themselves small. More marked results could be obtained with a larger newsbase or more advanced methods to filter out and score these news. These limits are discussed in the following section.

5.3 Sensibility of results

To check whether our results are sensitive to our methodology we test an alternative. Despite the small size of our newsbase, we try to use a less restrictive list of funds by removing the step where we only select funds from the top quartile in terms of TNA. As the list is larger, there are more funds that are not present for every quarter in fund investments data and have missing quarterly data in ESG fund data. In the two categories there are respectively 589 and 530 funds. It results in a final selection of 924 funds to construct the portfolio. The rest of the methodology remains the same.

The Table 4 shows the results of the in-sample estimates for the WSJI and Table 5 presents the ones for the CHI. With this new list of selected funds, the results for the portfolios hedging the WSJI are slightly different. Actually, for the absolute score, the refined portfolio is still worse than the baseline one, but for the ranked score the refined portfolio is now also worse than the baseline one. This is different from previous estimates, but the R^2's are still very close, so it cannot be said that this approach makes any real difference. Regarding Table 5, the main difference is for the absolute score.

Table 4: Regression from 2013 Q2 to 2017 Q2 on WSJI

	Reg. 14	Reg. 15	Reg. 16	Reg. 17
$Z_{t-1}^{SUS\_A} r_t$	0.00091***			
	(0.000036)			
$Z_{t-1}^{SUS\_R} r_t$		−0.0016***		
		(0.000063)		
$RZ_{t-1}^{SUS\_A} r_t$			0.00083***	
			(0.000033)	
$RZ_{t-1}^{SUS\_R} r_t$				−0.0016***
				(0.000064)
$Z_{t-1}^{SIZE} r_t$	−0.00011***	−0.00011***	−0.00011***	−0.00011***
	(0.0000015)	(0.0000016)	(0.0000015)	(0.0000016)
$Z_{t-1}^{HML} r_t$	0.0000068***	0.0000069***	0.0000067***	0.0000069***
	(0.00000099)	(0.00000098)	(0.00000098)	(0.00000098)
$Z_{t-1}^{MKT} r_t$	−0.000062***	−0.000062***	−0.000063***	−0.000062***
	(0.0000013)	(0.0000014)	(0.0000013)	(0.0000014)
Constant	0.00079***	0.00078***	0.00079***	0.00078***
	(0.0000069)	(0.0000069)	(0.0000069)	(0.0000069)
R^2	0.4786	0.4787	0.478	0.4784

This table shows results from regressions 15, 16, 17 and 18. The dependent variable for each regression is the measure of innovation of the WSJI (WSJI_AR1). Standard errors are in parentheses. *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0$

Indeed compared to previous estimates, the refined portfolio hedges better the climate risk than the portfolio constructed with the baseline ESG ratings. However, as for the former table the R^2's remain close. We can conclude that for the in-sample estimates there is no huge difference, the results are relatively robust to the size of portfolios.

The observation is different for the out-of-sample results presented in Table 6. The

Table 5: Regression from 2013 Q2 to 2018 Q1 on CHI

	Reg. 18	Reg. 19	Reg. 20	Reg. 21
$Z_{t-1}^{SUS\_A} r_t$	0.00046***			
	(0.0000084)			
$Z_{t-1}^{SUS\_R} r_t$		−0.00074***		
		(0.000015)		
$RZ_{t-1}^{SUS\_A} r_t$			0.00043***	
			(0.0000078)	
$RZ_{t-1}^{SUS\_R} r_t$				−0.00074***
				(0.000015)
$Z_{t-1}^{SIZE} r_t$	−0.000019***	−0.000019***	−0.000018***	−0.000019***
	(0.0000006)	(0.00000061)	(0.0000006)	(0.00000061)
$Z_{t-1}^{HML} r_t$	0.0000068***	0.0000069***	0.0000066***	0.0000069***
	(0.0000004)	(0.0000004)	(0.0000004)	(0.0000004)
$Z_{t-1}^{MKT} r_t$	0.000006***	0.0000056***	0.0000059***	0.0000057***
	(0.00000054)	(0.00000054)	(0.00000054)	(0.00000054)
Constant	−0.000042***	−0.0005***	−0.00004***	−0.000044***
	(0.0000028)	(0.0000048)	(0.0000028)	(0.0000028)
R^2	0.1763	0.1588	0.1776	0.1598

This table shows results from regressions 19, 20, 21 and 22. The dependent variable for each regression is the measure of innovation for the CHI (CHI_AR1). Standard errors are in parentheses. *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0$

baseline portfolio based on the absolute score becomes better for hedging the WSJI and the gap is a bit more marked between the two portfolio than in subsection 5.2. It is the opposite for the portfolios based on the ranked score and hdging the WSJI, the refined portfolio is now better but not by much. The latest change from the estimates made on the first selection of funds is in the absolute score for the CHI where the refined portfolio becomes better. Finally, the refined portfolio is still better in hedging the ranked score for the CHI. In contrast to the in-sample estimates, we observe a more marked difference

Table 6: Correlations between the different portfolios and the climate risk indices

	WSJI	CHI
P^{SUS_A}	0.0345	−0.5374
RP^{SUS_A}	0.0287	−0.5353
P^{SUS_R}	0.1312	0.1189
RP^{SUS_R}	0.1383	0.1451

This table shows the correlation between portfolios constructed thanks to equations 15, 16, 17, 18, 19, 20, 21 and 22 and their respective climate risk target.

between the two portfolio than with the initial funds, but only for the absolute score when hedging the WSJI. For the others, even if the balance of power is reversed, the differences remain as small as in subsection 5.2.

The conclusions of the estimates based on the first fund selection remain. The portfolios based on the two ESG scores get pretty similar results. Based on our methodology, it is therefore difficult to say whether one is better than the other.

6 Limitations

There are a number of areas for improvement in this work. It is worth discussing them to highlight them for future research. This section not only presents these various limitations, but also proposes solutions that may address them.

The first element that could lead to limitations is the data. We indeed base our analysis on the Sustainalytics ESG ratings because they are the only ones we have access to, but it would be interesting to test the approach with ESG scores from other providers such as MSCI, Refinitiv, Bloomberg or other in order to check if results are sensitive to the provider. However, the two key areas of improvement for the data are in the newsbase and the three additional risk factors. The newsbase for its part is relatively small for the period covered. Actually, although the website from which we collect the articles stores a large number of web archives, it is certainly not as comprehensive as the official newspaper sites. By scraping these websites it would probably be possible to increase the number of retrieved news, but some don't provide access to their archives and others are only accessible via paid subscriptions. The problem with the three control risk factors (Z^{SIZE}, Z^{HML}, Z^{MKT}) is that we don't have access to fund level data, so we have to use the Fama-French factors that are global. Therefore, if those factors can be calculated at the fund level as did Engle et al. (2019)[15], estimates will be more accurate.

The methodology also has a number of limitations, starting with the cleaning of company names. A better cleaning could be obtained by training a neural network to retrieve the common names. We had the time and the computing power to do it, but unfortunately to train this kind of model it is required to have a significant amount of labelled data (that is a database with pairs of legal names, and common names) that we don't have. With exact names for more firms, the stock related news gathering would be better. The names of the companies may also be names of firm groups that are not necessarily mentioned in the articles about their subsidiaries. This could be overcome by retrieving all the subsidiary names of each group. Another issue linked to the company names is the way to collect the stock related news since we lose a number of articles with the exact matching approach. There is no perfect method but one can try with vectorisation techniques such as *tf-idf*. The common names have a last problem that we have already pointed out in the subsection 4.3, they may be common nouns. Borms et al. (2021)[8] propose to remove from the portfolio the firms for which this is the case, but in our case it is not possible because as we work with funds, we have to consider all the stocks in the funds. By working with stocks, it would therefore be possible to improve the methodology.

The phase for retrieving ESG news also has weaknesses. The *tf-idf* is a good approach but it doesn't take semantics into account. Therefore, documents with similar words are considered similar when they may not be using these words in the same context and therefore may not be talking about the same topic. There exist techniques of word embedding that fix this issue such as GloVE or Word2Vec, but they require data and training, so it is technically difficult to use them for this paper. Another technique to explore for gathering ESG related news is to use a topic modeling model like Latent Dirichlet Allocation (LDA) and then check the obtained text clusters to label it, but it is not guaranteed to get an ESG related news group.

The LDA can also be used for the SA as shown by Moniz (2016)[31] and it is more advanced than the term counting approach used in this study because it takes semantic into accounts. The SA can also be achieved by Deep Learning that allows to model more complex relation like Day and Lee (2016)[13] explain in their paper. Another way to improve SA by keeping the same approach is to use an adapted sentiment lexicon since some words can be neutral in a certain context and positive/negative in another. For example the sentence "the company uses child labour" is negative in the ESG context but with a general dictionary as HIV it is not because both "child" and "labour" are neutral. However such a lexicon is not accessible for free and is really long to implement.

Finally, limitations that are beyond our control, but that must be mentioned, concern news issuers. It is indeed not impossible that they relay rumours, information that is not proven or that they are not objective in writing their articles, and all this will be taken into account in the scoring. Risks can be reduced by selecting reliable suppliers or implementing analyses using advanced NLP techniques, but they cannot be eliminated.

7 Conclusion

Through this work we attempt to see whether the incorporation of news in ESG ratings allows a better approximation of the exposure of funds to climate risk.

In this aim, we apply the methodology detailed above to refine the baseline ESG score of the funds using Data Science techniques. Among these techniques we find exact matching research based on company names to collect stock related news. We then filter them out thanks to the word embedding method *tf-idf* and the cosine similarity measure between articles and an aggregated sustainable glossary to keep only the articles that concern environmental, social and governance matters. Once the relevant newsbase is built up, we perform a sentiment analysis by term counting that gives a score for each

news. These scores are aggregated at the stock level and finally at the fund level to adjust their ESG rating.

To check if this refined score is a better climate risk proxy, we integrate it in the methodology developed by Engle et al. (2019)[15] that build a portfolio aimed at hedging climate risk using the mimicking portfolio approach. Although the approach based on the baseline ESG score provides globally a better hedge against climate risk in in-sample estimates, the finding is less clear in the out-of-sample estimates. Indeed, in this case, we find that the portfolio constructed thanks to the absolute refined score gets higher excess returns when the Wall Street Journal climate risk index increases. Also, the excess returns of the one based on the ranked refined score decrease less when there are more innovations in Crimson Hexagon index. When the portfolios are built on the basis of the ranked score or the absolute score and hedges respectively the WSJI and the CHI, the baseline ESG rating approach is better.

We can therefore say that the results really depend on the used climate risk index. However, despite the differences in performance, the results of the two approaches are very similar. This is probably due to the fact that the score adjustments are low. It could therefore be interesting to adapt the methodology in order to check whether the results persist. That is why we test ourselves a first alternative which does not fundamentally change the results and we propose several others. Among others, better results are expected by using fund level data to establish the three control risk factors, collecting more news to create the newsbase, or achieving the news filtering and the sentiment analysis using deep learning techniques.

References

[1] Alok, S., Kumar, N., & Wermers, R. (2020). Do Fund Managers Misestimate Climatic Disaster Risk. The Review of Financial Studies, 33(3), 1146–1183. https://doi.org/10.1093/rfs/hhz143

[2] Amel-Zadeh, A., & Serafeim, G. (2018). Why and How Investors Use ESG Information: Evidence from a Global Survey. Financial Analysts Journal, 74(3), 87–103. https://doi.org/10.2469/faj.v74.n3.2

[3] Baduge, S. K., Thilakarathna, S., Perera, J. S., Arashpour, M., Sharafi, P., Teodosio, B., Shringi, A., & Mendis, P. (2022). Artificial intelligence and smart vision for building and construction 4.0: Machine and deep learning methods and applications. Automation in Construction, 141, 104440. https://doi.org/10.1016/j.autcon.2022.104440

[4] Baily, C., & Gnabo, J. (2022). How Different Are ESG Mutual Funds? Evidence and Implications. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.4048577

[5] Baker, S. R., Bloom, N., & Davis, S. J. (2016). Measuring Economic Policy Uncertainty*. The Quarterly Journal of Economics, 131(4), 1593–1636. https://doi.org/10.1093/qje/qjw024

[6] Bartram, S. M., Branke, J., & Motahari, M. (2020). Artificial Intelligence in Asset Management. CFA Institute Research Foundation.

[7] Berg, F., Kölbel, J., & Rigobon, R. (2019). Aggregate Confusion: The Divergence of ESG Ratings. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3438533

[8] Borms, S., Boudt, K., Holle, F. V., & Willems, J. (2021). Semi-supervised text mining for monitoring the news about the ESG performance of companies. In Data science for economics and finance (pp. 217-239). Springer, Cham.

[9] Chatterji, A., & Levine, D. (2006). Breaking down the Wall of Codes: Evaluating Non-Financial Performance Measurement. California Management Review, 48(2), 29–51. https://doi.org/10.2307/41166337

[10] Choi, D., Gao, Z., & Jiang, W. (2020). Attention to Global Warming. The Review of Financial Studies, 33(3), 1112–1145. https://doi.org/10.1093/rfs/hhz086

[11] Consoli, S., Reforgiato Recupero, D., & Petkovic, M. (2019). Data science for healthcare Methodologies and applications. Berlin: Springer Nature.

[12] Consoli, S., Recupero, R. D., & Saisana, M. (2021). Data Science for Economics and Finance: Methodologies and Applications (1st ed. 2021 ed.). Springer.

[13] Day, M. Y., & Lee, C. C. (2016). Deep learning for financial sentiment analysis on finance news providers. 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). https://doi.org/10.1109/asonam.2016.7752381

[14] Delmas, M., & Blass, V. D. (2010). Measuring corporate environmental performance: the trade-offs of sustainability ratings. Business Strategy and the Environment, 19(4), 245–260. https://doi.org/10.1002/bse.676

[15] Engle, R. F., Giglio, S., Lee, H., Kelly, B. T., & Stroebel, J. (2019). Hedging Climate Change News. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3422236

[16] Garz, H., Volk, C., & Morrow, D. (2018). The ESG risk ratings: Moving up the innovation curve. Sustainalytics White Paper.

[17] Global Sustainable Investment Alliance. (July 15, 2021). Value of sustainable assets under management (AUM) and total assets under management worldwide from 2016 to 2020 (in billion U.S. dollars) [Graph]. In Statista. Retrieved August 14, 2022, from https://www.statista.com/statistics/948492/value-sustainable-total-aum-worldwide/

[18] Görgen, M., Nerlinger, M., & Wilkens, M. (2017). Carbon Risk. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.2930897

[19] Gupta, A., Dengre, V., Kheruwala, H. A., & Shah, M. (2020). Comprehensive review of text-mining applications in finance. Financial Innovation, 6(1). https://doi.org/10.1186/s40854-020-00205-1

[20] Hong, H., Li, F. W., & Xu, J. (2019). Climate risks and market efficiency. Journal of Econometrics, 208(1), 265–281. https://doi.org/10.1016/j.jeconom.2018.09.015

[21] Hotho, A., Nürnberger, A., & Paaß, G. (2005). A brief survey of text mining. In Ldv Forum (Vol. 20, No. 1, pp. 19-62).

[22] Jing, N., Wu, Z., & Wang, H. (2021). A hybrid model integrating deep learning with investor sentiment analysis for stock price prediction. Expert Systems with Applications, 178, 115019. https://doi.org/10.1016/j.eswa.2021.115019

[23] Krappel, T., Bogun, A., & Borth, D. (2021). Heterogeneous ensemble for ESG ratings prediction. arXiv preprint arXiv:2109.10085.

[24] Kraus, M., & Feuerriegel, S. (2017). Decision support from financial disclosures with deep neural networks and transfer learning. Decision Support Systems, 104, 38–48. https://doi.org/10.1016/j.dss.2017.10.001

[25] Lavrenko, V., Schmill, M., Lawrie, D., Ogilvie, P., Jensen, D., & Allan, J. (2000, August). Mining of concurrent text and time series. In KDD-2000 Workshop on text mining (Vol. 2000, pp. 37-44). University Park, PA, USA: Citeseer.

[26] le Guenedal, T., & Roncalli, T. (2022). Portfolio Construction with Climate Risk Measures. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3999971

[27] Lemoine, D. (2021). The Climate Risk Premium: How Uncertainty Affects the Social Cost of Carbon. Journal of the Association of Environmental and Resource Economists, 8(1), 27–57. https://doi.org/10.1086/710667

[28] Liddy, E.D. 2001. Natural Language Processing. In Encyclopedia of Library and Information Science, 2nd Ed. NY. Marcel Decker, Inc.

[29] Liu, B. (2012). Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies, 5(1), 1–167. https://doi.org/10.2200/s00416ed1v01y201204hlt016

[30] Luccioni, A., Baylor, E., & Duchene, N. (2020). Analyzing sustainability reports using natural language processing. arXiv preprint arXiv:2011.08073.

[31] Moniz, A. (2016). Inferring the Financial Materiality of Corporate Social Responsibility News. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.2761905

[32] Olmedo, E. E., Torres, M. J. M., & Izquierdo, M. A. F. (2010). Socially responsible

investing: sustainability indices, ESG rating and information provider agencies. International Journal of Sustainable Economy, 2(4), 442. https://doi.org/10.1504/ijse.2010.035490

[33] Olmedo, E. E., Muñoz-Torres, M. J., Fernández-Izquierdo, M. N., & Rivera-Lirio, J. M. (2013). Lights and shadows on sustainability rating scoring. Review of Managerial Science, 8(4), 559–574. https://doi.org/10.1007/s11846-013-0118-0

[34] Olmedo, E. E., Fernández-Izquierdo, M., Ferrero-Ferrero, I., Rivera-Lirio, J., & Muñoz-Torres, M. (2019). Rating the Raters: Evaluating how ESG Rating Agencies Integrate Sustainability Principles. Sustainability, 11(3), 915. https://doi.org/10.3390/su11030915

[35] Roberts, K., Radev, D., & Kelly, B. (2019). An ESG Performance Metric Based on Text Analysis. Yale LILY Lab.

[36] Saadaoui, K., & Soobaroyen, T. (2018). An analysis of the methodologies adopted by CSR rating agencies. Sustainability Accounting, Management and Policy Journal, 9(1), 43–62. https://doi.org/10.1108/sampj-06-2016-0031

[37] Scalet, S., & Kelly, T. F. (2009). CSR Rating Agencies: What is Their Global Impact? Journal of Business Ethics, 94(1), 69–88. https://doi.org/10.1007/s10551-009-0250-6

[38] Tajmazinani, M., Hassani, H., Raei, R., & Rouhani, S. (2022). Modeling stock price movements prediction based on news sentiment analysis and Deep Learning. Annals of Financial Economics, 17(01). https://doi.org/10.1142/s2010495222500038

[39] Time. (September 23, 2019). Change in average temperature by decade world-

wide from 1910 to 2019, by region (in degrees Celsius)* [Graph]. In Statista. Retrieved June 25, 2022, from https://www.statista.com/statistics/1054149/difference-temperature-decade-worldwide-by-region/

[40] Windolph, S. E. (2011). Assessing Corporate Sustainability Through Ratings: Challenges and Their Causes. Journal of Environmental Sustainability, 1(1), 1–22. https://doi.org/10.14448/jes.01.0005

[41] Wisetsri, W., Donthu, S., Mehbodniya, A., Vyas, S., Quiñonez-Choquecota, J., & Neware, R. (2022). An Investigation on the Impact of Digital Revolution and Machine Learning in Supply Chain Management. Materials Today: Proceedings, 56, 3207–3210. https://doi.org/10.1016/j.matpr.2021.09.367

A Appendix

A.1 Sustainable Glossaries Sources

Here are all the sources from which the glossaries were collected.

There is one pdf glossary from Global Reporting Initiative: Global Reporting Initiative

Other glossaries come from website: BBC, Wikipedia, Government of Canada, University of Miami, California Air Ressources Board, Fresh Air. The Scent of Pine., U.K. Climate Impacts Programme (UKCIP), CBC News, Auburn University, U.S. Climate Resilience Toolkit, Nitric Acid Climate Action Group, National Geographic, Conservation in a Changing Climate, South West Climate Change Impacts Partnership, International Petroleum Industry Environmental Conservation Association, Agricultural Marketing Ressource Center, Mekong River Comission for Sustainable Development, Irin

A.2 Python code used for Fund Selection described subsection 4.1

A.2.1 Tools for processing DataFrame

```
# Code taken from Professor Mickaël Tits' Data Analysis course at UNamur
import pandas as pd
import numpy as np

def is_cat(series, relative_threshold = 0.5, absolute_threshold = 20):
    """
    Cette fonction permet de vérifier si une
    colonne d'un dataframe est catégorielle
    """
    nvalues = len(series)
    ncats = len(series.unique())
    #On considère la variable comme catégorielle si le nombre de valeurs uniques
    # et plus petit qu'un seuil relatif ou absolu
    return (ncats/nvalues <= relative_threshold) and (ncats <= absolute_threshold)

def is_num(series):
    """
    Cette fonction permet de vérifier si une colonne d'un dataframe est numérique
    """
    try:
        series.astype(float)
        return True
    except:
```

```
        return False

#Détection d'outliers numériques
def detect_numeric_outliers_series(series, h = 2, left = 0.25, right = 0.75):
    """
    Cette fonction permet de détecter les outliers des fonctions numériques
    """
    Q1 = series.quantile(left)
    Q3 = series.quantile(right)
    IQR = Q3 - Q1
    #Les données s'écartant fortement des quantiles sont potentiellement
    # des anomalies (outlier en anglais)
    is_outlier = (series < (Q1 - h * IQR)) | (series > (Q3 + h * IQR))
    if is_outlier.sum() > 0:
        print(is_outlier.sum(), "outliers numériques trouvés pour", series.name)
    return is_outlier

# Détection d'outliers catégoriels: on identifie une catégorie rare
# si elle représente moins de 1% (seuil relatif) des données
def detect_rare_cat_series(series, relative_threshold = 0.01, nmin = 2):
    """
    Cette fonction permet de détecter les outliers de variables catégorielles
    """
    counts = series.value_counts()
    #on garde la contrainte la plus souple (< rel_th% des données, ou < nmin)
    relative_nmin = relative_threshold*len(series)
```

```
    tot_nmin = max(nmin, relative_nmin)
    is_rare = (counts <  tot_nmin)
    rare_cats = counts[is_rare]
    is_outlier = series.isin(rare_cats.index)
    if is_outlier.sum() > 0:
        print(is_outlier.sum(), "outliers catégoriels trouvés pour", series.name)
    return is_outlier

#Détection générique des outliers dans une Series
def detect_outlier_series(series):
    """
    Cette variable détecte automatiquement le type de colonne et les
    outliers leur appartenant
    """
    if is_cat(series):
        return detect_rare_cat_series(series)
    elif is_num(series):
        return detect_numeric_outliers_series(series)
    else:
    #return a Series containing only False
        return pd.Series(False, series.index)
```

A.2.2 Code used for Fund Selection

```
import os
import sys
import pandas as pd
import numpy as np
```

```
# Local imports
import df_tools as dt


def load_funds_investments(dir_path):
    """
    Loads all the files containing the distribution of the
    funds' investments as Pandas DataFrames.
    Parameters
    ----------
    dir_path: the path to the directory containing all
    the file of distribution of funds' investments
    Return
    ------
    investments_distribution: a dictionnary containing the
    DataFrames with the weights of the funds. The keys are the
    numbers of the quarter (dict of Pandas DataFrame)
    """
    investments_distribution = {}
    for filename in os.listdir(dir_path):
        # Retrieve the file as a Pandas DataFrame
        filepath = os.path.join(dir_path, filename)
        fund_investments = pd.read_csv(filepath)
        # Retrieve the number of the quarter
        quarter_nb = ''
        for s in filename:
            if s.isdecimal():
```

```
                quarter_nb += s
        # Storing the dataframe of the current quarter
        # in the dict
        if quarter_nb != '':
            investments_distribution[quarter_nb] = fund_investments
    return investments_distribution


def detect_outliers(funds_df):
    """
    Detects outliers in a Pandas DataFrame with funds Id as index,
    the total net assets of the funds in the column 'TNA', and
    the number of stocks in the fund in the column 'Holdings'.
    Parameters
    ----------
    funds_df: a Pandas DataFrame with funds Id as index,
    the total net assets of the funds in the column 'TNA', and
    the number of stocks in the fund in the column 'Holdings'
    Returns
    -------
    outliers_indices: list of outliers' indices in the DataFrame
    """
    # Detection of the outliers for each column
    outliers = funds_df.apply(dt.detect_outlier_series)
    # Retrieve the indices of the funds that are outliers
    # in the two columns
    outlier_sum = outliers.sum(axis = 1)
```

```
    n_cols = 1
    outliers_indices = funds_df[outlier_sum>= n_cols].index.tolist()
    return outliers_indices


def selects_funds(funds_aggregated_data, funds_investments_distribution):
    """
    Selects the appropriate funds for the portfolio and
    retrieves their Id's. To select the funds, first it takes
    the 25% of the funds that invested the most on average
    in the period. Then it takes in those funds the 25%
    of the funds with fewest assets. So, it remains 1/4
    of the original number of funds.
    Parameters
    ----------
    funds_agreggated_data: the DataFrame that contains the
    agreggated fund data (Pandas DataFrame)
    funds_investments_distribution: the dictionnary containing
    the DataFrames with the quarterly fund data (dict of Pandas DataFrames)
    Returns
    -------
    selected_funds_id: list of the fund Id's (list of str)
    """
    # Group the data of different quarters of a fund by fund and
    # by taking the mean to have the mean TNA
    funds_size = funds_aggregated_data[
                ['Id', 'TNA']].groupby('Id').mean()
```

```
print('nombre de fonds au départ: ', funds_size.count())
# Detect outliers on PFSustScore
fund_score = funds_aggregated_data[
            ['Id', 'PFSustScore']].groupby('Id').mean()
outliers_indices = detect_outliers(fund_score)
# Drop the outliers
funds_size = funds_size.drop(labels=outliers_indices, axis=0)
print('nombre de fonds après outliers: ', funds_size.count())
# Select funds according to their quantiles.
Q3_TNA = funds_size['TNA'].quantile(0.75)
# Keep the 25% of funds that invested the biggest
# amount on average during the period
selected_funds = funds_size[funds_size['TNA'] >= Q3_TNA]
# Retrieve Id of the selected funds
selected_funds_id = selected_funds.index.to_list()
# Uncoment the line below and comment the line above
# to select all the funds for the portfolio
#selected_funds_id = funds_size.index.to_list()
# Adding Unnamed: 0 (the column of funds ids)
# so that the column is taken into account when
# selecting the funds in funds_investments
selected_funds_id.append('Unnamed: 0')
#print('selected funds number:', len(selected_funds_id))
# Clean the indices list by removing the funds are not
# present every quarter
not_in_df = [i for quarter in funds_investments_distribution
```

```
                for i in selected_funds_id if i not in
                funds_investments_distribution[quarter].columns]
    not_in_df = list(set(not_in_df))
    print('number of funds not present in every quarter investments distribution: ',
          len(not_in_df))
    for fund in not_in_df:
        if fund in selected_funds_id:
            selected_funds_id.remove(fund)
    return selected_funds_id


def main(funds_investments_dir, aggregated_fund_data_path):
    """
    The main function that takes the paths to the directory
    containing the quarterly fund data and to the agreggated
    fund data file, and returns the DataFrame containing
    the agreggated data for the funds that have been
    selected for the portfolio.
    Parameters
    ----------
    funds_investments_dir: path to the directory
    containing the quarterly fund data (str)
    agreggated_fud_data_path: path to the agreggated
    fund data file(str)
    Returns
    -------
    aggregated_selected_funds: the DataFrame containing
```

```
    the agreggated data for the funds that have been
    selected for the portfolio (Pandas Dataframe)
    selected_funds_id: list of id of the
    selected funds (list of str)
    """
    funds_investments_distribution =  load_funds_investments(funds_investments_dir)
    aggregated_funds = pd.read_csv(aggregated_fund_data_path)
    selected_funds_id = selects_funds(aggregated_funds,
                                      funds_investments_distribution)
    # Remove funds that have missing quarters in aggregated data
    not_every_quarter = []
    for fund in selected_funds_id:
        quarter_list = aggregated_funds.loc[
        aggregated_funds[aggregated_funds['Id']==fund].index, 'Quarter']
        if len(quarter_list) < 24 and fund != 'Unnamed: 0':
            not_every_quarter.append(fund)
    for fund in not_every_quarter:
        selected_funds_id.remove(fund)
    print('number of funds having missing quarters in aggregated data: ',
         len(not_every_quarter))
    print('selected funds number after drop:', len(selected_funds_id))
    # Retrieve the indices of the selected funds in the agreggated data
    selected_funds_indices = [idx for fund in selected_funds_id
                              for idx in aggregated_funds
                              [aggregated_funds['Id']==fund].index.tolist()]
    # Build the DataFrame with agreggated data on the selected funds
```

```
    aggregated_selected_funds = aggregated_funds.loc[selected_funds_indices]
    return aggregated_selected_funds, selected_funds_id
```

A.3 Python code used for Stock Names Retrieval described in subsection 4.2

```
import os
import sys
import re
import pandas as pd

def get_needed_stocks(funds_id, selected_quarter):
    """
    Makes a list of all the stocks present in
    selected funds for the selected quarter.
    Parameters
    ----------
    funds_id: a list of ids of the funds whose
    investments are to be retrieved (list of str)
    selected_quarter: the number of the quarter
    for which the investments of the funds are to
    be retrieved (int)
    Returns
    -------
    selected_stocks: list of all the stocks ids present
    in a fund for the selected quarter
    stocks_by_funds: funds investments repartition for
    each stock
    """
    funds_investments_path ='/Users/gregoiredesauvage/Documents/UNamur/
```

```
Mémoire/Ressources/Données Fonds/Quarters_investments/Quarter_%d.csv'%selected_quar
funds_investments = pd.read_csv(funds_investments_path)
# Selecting funds
stocks_by_funds = funds_investments[funds_id]
# Setting index on the funds ids
stocks_by_funds = stocks_by_funds.set_index('Unnamed: 0')
stocks_in_funds = {}
# Sometimes, there are funds in the stocks, so we
# need to filter the funds list
for fund in stocks_by_funds:
    fund_in = False
    non_null_stocks = stocks_by_funds[fund].dropna()
    stocks_in_fund = non_null_stocks.index.to_list()
    for stock in stocks_in_fund:
        # Checks if it is a fund (funds id start with a F)
        if stock.startswith('F'):
            fund_in = True
    # If there is a fund in the current fund, it isn't
    # added to the dictionnary stocks by fund
    if fund_in:
        continue
    stocks_in_funds[fund] = stocks_in_fund
# Makes a list of all stocks retrieved
selected_stocks = [stock for fund in stocks_in_funds
                   for stock in stocks_in_funds[fund]]
# Drops duplicates
```

```
    selected_stocks = list(set(selected_stocks))
    return selected_stocks, stocks_by_funds


def clean_stocks_names(stocks_names_file, selected_stocks_list):
    """
    This function retrieves the names of the selected stocks and
    cleans the names of all suffixes.
    Parameters
    ----------
    stocks_names_file: path to the file containing all the stocks
    names
    selected_stocks_list: list of selected stocks ids
    Retruns
    -------
    names: a Dataframe containing the original names and
    the cleaned names of the stocks
    """
    stocks_names = pd.read_csv(stocks_names_file, sep=';')
    stocks_names = stocks_names.set_index('Id')
    names = stocks_names.loc[selected_stocks_list]
    strings_to_replace = r''' Inc\b| Corp\b| Ltd| Co\b| Co | PLC|
                              LLC| NV| /NV| LP| SA| S.A.| SE| DR|
                              L.P.| N.V.| NA| AG| A[/|\]S| SpA|ADR|
                              Class [A-Z]| [A-Z]\b|\.|,| | Holdings|
                              Holding'''
    cleaned_names = [re.sub(strings_to_replace, '', name)
```

```
                    for name in names['Name'].to_list()]
    names['Cleaned Name'] = pd.Series(cleaned_names,
                                      index=names.index.to_list())
    return names


def main(funds_id, selected_quarter, stocks_names_file):
    """
    Takes a list of fund id's, the desired quarter for
    which to retrieve stock names and the stock present
    in each fund, and the file with stock id's and their
    names.
    Parameters
    ----------
    funds_id: a list of id of the
    selected funds (list of str)
    selected_quarter: the quarter to which apply the
    function (int)
    stocks_names_file: path to the file containing
    the stock id's and their names
    Returns
    -------
    selected_stocks: a list of id's of all the stocks
    present in selected funds for the selected quarter
    (list of str)
    stocks_by_funds: a dictionary with fund id's as keys
    and a list of id's of stocks present in this fund
```

```
    (dict of list of str)
    stocks_names: a DataFrame with stock id's as indices
    and two columns: one with stock legal names and another
    with their common names (Pandas DataFrame)
    """
    selected_stocks, stocks_by_funds = \
    get_needed_stocks(funds_id, selected_quarter)
    stocks_names = clean_stocks_names(stocks_names_file,
                                      selected_stocks)
    return selected_stocks, stocks_by_funds, stocks_names
```

A.4 Python code used for News Gathering and Filtering described in subsection 4.3

```
import os
import sys
import pandas as pd
import numpy as np
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.tokenize import sent_tokenize
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import math
import json

def newsbase_creation(news_dir):
    """
    Creates the news Dataframe from a directory
    containing all the news files.
    Parameters
    ----------
    news_dir: the path to the directory that
    contains all the news files (str)
    Returns
    -------
    newsbase: Dataframe containing all the news text,
    their date of publication, their title and
```

```
    their url (Pandas Dataframe)
    """
    news_dict = {}
    maintexts = []
    date_publish = []
    titles = []
    urls = []
    for directoryname in os.listdir(news_dir):
        if 'DS_Store' not in directoryname:
            dir_path = os.path.join(news_dir, directoryname)
            for filename in os.listdir(dir_path):
                if 'DS_Store' not in filename:
                    file_path = os.path.join(dir_path, filename)
                    myfile = open(file_path)
                    article = json.load(myfile)
                    maintexts.append(article['maintext'])
                    date_publish.append(article['date_publish'])
                    titles.append(article['title'])
                    urls.append(article['url'])
    news_dict['Article text'] = maintexts
    news_dict['Date published'] = date_publish
    news_dict['Headline'] = titles
    news_dict['Url'] = urls
    newsbase = pd.DataFrame.from_dict(news_dict)
    newsbase = newsbase.drop_duplicates(subset='Headline')
    newsbase = newsbase.drop_duplicates(subset='Article text')
```

```
    return newsbase


def news_by_quarter(newsbase, selected_quarter):
    """
    Retrieve the news of a certain quarter.
    Parameters
    ----------
    newsbase: the newsbase in which to collect
    the news of the selected quarter (Pandas Dataframe)
    selected_quarter: the number of the quarter
    for which the investments of the funds are to
    be retrieved (int)
    Returns
    -------
    quarter_newsbase: Dataframe containing the
    news of the selected quarter (Pandas Dataframe)
    """
    newsbase_2 = newsbase.copy()
    newsbase_2 = newsbase_2.dropna(subset=['Date published', 'Article text'])
    if selected_quarter < 9:
        y = 3
    elif selected_quarter >= 9 \
    and selected_quarter < 13:
        y = 4
    elif selected_quarter >= 13 \
    and selected_quarter < 17:
```

```
            y = 5
        elif selected_quarter >= 17 \
        and selected_quarter < 21:
            y = 6
        elif selected_quarter >= 21 \
        and selected_quarter < 25:
            y = 7
        else:
            y = 8
        if selected_quarter in [5,9,13,17,21,25]:
            m1, m2, m3 = '01','02', '03'
        elif selected_quarter in [6,10,14,18,22,26]:
            m1, m2, m3 = '04','05', '06'
        elif selected_quarter in [7,11,15,19,23,27]:
            m1, m2, m3 = '07','08', '09'
        elif selected_quarter in [8,12,16,20,24,28]:
            m1, m2, m3 = '10','11', '12'
        quarter_newsbase = \
        newsbase_2[newsbase_2['Date published'].str.contains('201%d-%s'%(y, m1))
        |newsbase_2['Date published'].str.contains('201%d-%s'%(y, m2))
        |newsbase_2['Date published'].str.contains('201%d-%s'%(y, m3))]\
        .sort_values(by='Date published')
        return quarter_newsbase

    def stocks_name_filter(newsbase, names_df):
        """
```

```
    Filters the news to keep only those containing
    the name of a selected firm.
    Parameters
    ----------
    newsbase: a Dataframe containing the news
    to be treated (Pandas Dataframe)
    names_df: a Dataframe containing two columns
    of names (Pandas Dataframe)
    Returns
    -------
    news_indices_by_stock: a dict with the complete
    name of stocks as keys and a list of indices of
    the related news (dict of list of int)
    news_by_stock: dictionnary with indices of stocks
    in the newsbase as keys and a list of articles
    texts as value (dict of lists of str)
    pertinent_news: list of all pertinent news
    retrieved (list of str)
    """
    # Retrieval of news containing company names
    # N.B.: itertuplres() returns tuples,
    # so to acces to a data use int indices
    news_indices_by_stock = {}
    for stock in names_df.itertuples():
        news_current_stock = newsbase[newsbase['Article text']\
                             .str.contains('%s |%s '%(stock[1],stock[2]),
```

```
                                regex=True)].index.to_list()
            news_current_stock = list(set(news_current_stock))
            news_indices_by_stock[stock[1]] = news_current_stock
        news_by_stock = {}
        pertinent_news = []
        for stock in news_indices_by_stock:
            news_for_stock = newsbase.loc[news_indices_by_stock[stock]]\
                            ['Article text'].to_list()
            news_by_stock[stock] = news_for_stock
            pertinent_news += news_for_stock
        pertinent_news = list(set(pertinent_news))
        return news_by_stock, pertinent_news


def calc_similarity_matrix(texts_corpus):
    """
    Takes a corpus of texts as input, vctorizes
    it, and computes a similarity matrix.
    Parameters
    ----------
    texts_corpus: a list containing texts (list of str)
    Returns
    -------
    similarity_matrix: a matrix of similarity scores
    between each doc (ndarray of shape (nb_texts, nb_texts))
    """
    vectorizer = TfidfVectorizer(stop_words='english')
```

```
    X = vectorizer.fit_transform(texts_corpus)
    doc_term_matrix = X.todense()
    vectors_df = pd.DataFrame(doc_term_matrix,
                              columns=vectorizer.get_feature_names())
    similarity_matrix = cosine_similarity(vectors_df, vectors_df)
    return similarity_matrix


def esg_filter(newsbase, glossaries_path):
    """
    This function allows to retrieve indices
    of ESG related articles in a newsbase. It
    is just required to provide the newsbase
    to be analysed, and the path to the ESG
    glossaries corpus.
    Parameters
    ----------
    newsbase: the newsbase to be filtered
    (Pandas Dataframe)
    glossaries_path: the path to the glossaries
    corpus file (str)
    Returns
    -------
    unique_esg_docs: list of the ESG related
    news inidces in the newsbase (list of int)
    """
    with open(glossaries_path,
```

```
            'rb') as f:
        glossary_corpus = json.load(f)
f.close()
# Global corpus is the corpus containing the glossaries and
# all the pertinent news retrieved so far
global_corpus = glossary_corpus + newsbase
sim_matrix = calc_similarity_matrix(global_corpus)
esg_doc = []
# Retrieving pertinent docs
sim_matrix_median = np.quantile(sim_matrix, 0.5)
for doc in sim_matrix:
    # If the similarity with the ESG glossary
    # (i.e. doc[similarity_matrix.shape[1]-1])
    # is above the median of the sim scores
    if doc[sim_matrix.shape[1]-1] > sim_matrix_median:
        # Appending the corpus news index of the news
        # in the list of pertinent docs
        esg_doc.append(np.where(sim_matrix==doc)[0][0])
# Detecting near-duplicate news thanks
# to the cosine similarity
duplicates_esg_doc = []
for i in esg_doc:
    for j in esg_doc:
        # If the similarity between 2 docs is above 0.5
        # and the same pair is not in the duplicates
        # list yet
```

```
            if i != j and sim_matrix[i][j] > 0.5\
            and (j,i,sim_matrix[i][j]) not in duplicates_esg_doc:
                duplicates_esg_doc.append((i,j,sim_matrix[i][j]))
    # Making the list of uniques pertinent ESG news
    unique_esg_docs = []
    for doc in esg_doc:
        doc_in_dup = False
        for dup in duplicates_esg_doc:
            # If the news is in the duplicates list or
            # if its duplicate is already in the unique
            # pertinent docs list the doc is considered
            # as a duplicate
            if doc in dup and dup[0] in unique_esg_docs:
                doc_in_dup = True
        # If the doc isn't a duplicate and isn't the last
        # element of the ESG docs list
        # (that is the ESG glossary)
        if doc_in_dup == False and esg_doc.index(doc) != len(esg_doc)-1:
            unique_esg_docs.append(doc)
    return unique_esg_docs


def main(news_dir, glossaries_path, names_df,
         selected_quarter):
    """
    Retrieves the news and filters them to keep only
    stocks and ESG related news.
```

```
    Parameters
    ----------
    news_dir: the path to the directory that
    contains all the news files (str)
    glossaries_path: the path to the glossaries
    corpus file (str)
    names_df: a Dataframe containing the original names and
    the cleaned names of the stocks (Pandas Dataframe)
    selected_quarter: the number of the quarter
    for which the investments of the funds are to
    be retrieved (int)
    Returns
    -------
    news_indices_by_stock: a dict with the complete
    name of stocks as keys and a list of indices of
    the related news (dict of list of int)
    news_by_stock: dictionnary with indices of stocks
    in the newsbase as keys and a list of articles
    texts as value (dict of lists of str)
    pertinent_news: list of all pertinent news
    retrieved (list of str)
    esg_news_indices: list of the ESG related
    news inidces in the newsbase (list of int)
    """
    newsbase = newsbase_creation(news_dir)
    quarter_newsbase = news_by_quarter(newsbase,
```

```
                                    selected_quarter)
news_by_stock, pertinent_news = \
stocks_name_filter(quarter_newsbase, names_df)
esg_news_indices = esg_filter(pertinent_news, glossaries_path)
print('Pour ce trimestre, il y a %d news'%len(esg_news_indices))
return news_by_stock, pertinent_news, esg_news_indices
```

A.5 Python code used for Sentiment Analysis described in subsection 4.4

All the functions of this code except "main" were retrieved and adapted from the poster of Roberts et al. (2019)[35]. The authors kindly let us access and reuse their code.

```
import os
import sys
import pandas as pd
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.tokenize import sent_tokenize
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import math
import csv


def load_sentiment_data(filepath):
    """
    Reads dictionary entries from a tab-delimited CSV text
    file into a python dictionary object.
    Filters out entries not in the Harvard IV-4 dictionary
    depending on the config parameter 'restrict_harvard'.
    Returns a tuple of the full dictionary and a dictionary
    that maps the names of the column headers to column indices.
    """
    restrict_harvard = True
```

```
    fp = open(filepath, 'r')
    dict_reader = csv.reader(fp, delimiter='\t')
    sentiment_dict = {}
    header_i = {} # Column header indices, keyed by name
    first_line = True # Whether 'row' is the first line
    source_i = 1 # Column index of word source
    for row in dict_reader:
        if first_line:
      # Create a dict of header names
            for i, name in enumerate(row):
                header_i[name] = i
            first_line = False
        elif (not restrict_harvard) or \
        (row[header_i['Source']] == 'H4' or \
         row[header_i['Source']] == 'H4Lvd'):
          # Add word to dictionary
            sentiment_dict[row[0]] = row
    fp.close()
    return (sentiment_dict, header_i)


def count_category_freq(sentence, category, sentiment_data):
    """
    Counts the number of words of a given category that occur
    in the sentence.
    Parameters:
    - sentence: string
```

```
    - category: string
    - sentiment_data: tuple of
        (sentiment dictionary, header index dictionary).
        The return value of load_sentiment_data().
    """
    sentiment_dict, header_i = sentiment_data
    category_i = header_i[category]
    freq = 0
    for word in word_tokenize(sentence):
        key = word.upper()
        if key in sentiment_dict and \
        sentiment_dict[key][category_i] == category:
            freq += 1
    return freq

def _calc_sentiment_scores_helper(sentences, category, sentiment_data):
    """
    Helper function 'for calc_sentiment_scores()'.
    """
    total_freq, n_sentences = 0.0, len(sentences)
    for sentence in sentences:
        total_freq += count_category_freq(sentence,\
                                          category, sentiment_data)
    return (total_freq / n_sentences if n_sentences != 0 else 0.0)

def calc_sentiment_scores(text, category, sentiment_data):
```

```
    """
    Calculates a sentiment score for the 10K
    toward the specified reporting standard.
    Returns the total sentiment score of the text
    """
    sentences = sent_tokenize(text)
    score = _calc_sentiment_scores_helper(sentences,\
                                category, sentiment_data)
    return score


def main(sentiment_data_path, newsbase, esg_news_list):
    """
    Calculates the sentiment score for the news in the
    newsbase and return a dictionnary with indices of
    the news as keys, and a positive, a negative and a
    global score as values.
    Parameters
    ----------
    sentiment_data_path: the path to the sentiment data
    file (str)
    newsbase: Dataframe containing the news to be
    analysed (Pandas Dataframe)
    esg_news_list: list of the ESG related
    news inidces in the newsbase (list of int)
    Returns
    -------
```

```
    scores: dictionnary with news indices as keys
    and a dictionnary with Positive, Negative and Global
    as keys that contain a positive, a negative and a
    global score as values.
    (dict of dict of float)
    """
    sentiment_data = load_sentiment_data(sentiment_data_path)
    scores = {}
    scores_list = []
    for i in esg_news_list:
        scores[i] = {}
        scores[i]['Positive'] =\
        calc_sentiment_scores(newsbase[i], 'Positiv', sentiment_data)
        scores[i]['Negative'] =\
        calc_sentiment_scores(newsbase[i], 'Negativ', sentiment_data)
        scores[i]['Global'] =\
        scores[i]['Positive'] - scores[i]['Negative']
        scores_list.append(scores[i]['Global'])
    return scores
```

A.6 Python code for Score Refining described in subsection 4.5

```
import os
import sys
import pandas as pd
import numpy as np

def stocks_score_calc(scores_dict, news_by_stock, newsbase):
    """
    Calculates the total score for each stock.
    Parameters
    ----------
    scores_dict: a dict of dict with news indices
    as first keys, Positive, Negative and Global
    as second keys and score as values (dict of dict of int)
    Returns
    -------
    stocks_score: dict with stock indices as keys
    and standardized score as value (dict of float)
    """
    # Calcul du score global par stock car
    # une stock peut avoir plusieurs news
    stocks_score = {}
    for i in scores_dict:
        for stock in news_by_stock:
            if newsbase[i] in news_by_stock[stock]\
            and stock not in stocks_score.keys():
```

```
                stocks_score[stock] = scores_dict[i]['Global']
            elif stock not in stocks_score.keys()\
            and stock in stocks_score.keys():
                stocks_score[stock] += scores_dict[i]['Global']
    return stocks_score


def funds_score_calc(stock_by_funds, stock_names, stocks_score):
    """
    Calculates the refining score for each fund.
    Parameters
    ----------
    stock_by_funds: Dataframe with funds Ids as
    columns names, stocks Ids as index and the weight
    of each stock in the fund as values
    stock_names: Dataframe with stocks Ids as index
    and at least a column 'Name' containing the names
    of stocks
    Returns
    -------
    funds_score: dict with funds Ids as keys and
    standardized refining score as values
    (dict of float)
    """
    # Calcul des scores d'ajustement par fond
    # Initialisation d'un dict avec chaque fond comme clé
    # et une valeur de 0 pour le score
```

```
    funds_score = {fund:0.0 for fund in stock_by_funds}
    # Pour chauqe fond
    for fund in funds_score:
        # Pour chauqe actif
        for stock in stocks_score:
            # Je récupère l'Id de l'actif
            stock_id = stock_names[stock_names['Name'] == stock].index[0]
            # Si la part de l'actif dans le portefeuille n'est pas nan
            # = si le fond a investi dans l'actif
            if not np.isnan(stock_by_funds.loc[stock_id, fund]):
                # J'ajoute le score de l'actif pondéré par le poids
                # de ce dernier dans le fond
                funds_score[fund] += \
                ((stock_by_funds.loc[stock_id, fund]/100)*stocks_score[stock])
    # Standardisation du score par fond
    scores_list = [funds_score[fund] for fund in funds_score]
    mean = np.mean(scores_list)
    if mean != 0:
        for fund in funds_score:
            funds_score[fund] = (funds_score[fund]\
                                 -np.mean(scores_list))/np.std(scores_list)
    else:
        print('Mean score is 0')
    return funds_score

def scores_update(selected_funds_df, score_by_fund, quarter):
```

```
    """
    Updates the score of the selected funds for the
    quarter you want. Saves the updated Dataframe
    as a csv in the path tou give.
    Parameters
    ----------
    selected_funds_df: a dataframe containing all the data
    of the funds you want to update (Pandas DataFrame)
    score_by_fund: a dictionnary with fund_Id: adjustment_score (dict{str: float})
    quarter: the quarter for which you want to update the score (int)
    """
    selected_funds_df.to_csv('/Users/gregoiredesauvage/Documents/UNamur/
    Mémoire/Ressources/Données Fonds/funds_data_before_Q%d.csv'%quarter)
    print('###### File before update saved ! #######')
    non_zero_fund_nb = 0
    for fund in\
    selected_funds_df.loc[selected_funds_df['Quarter']==quarter, 'Id'].tolist():
        if score_by_fund[fund] > 0:
            selected_funds_df.loc[(selected_funds_df['Quarter']==quarter)&
                                  (selected_funds_df['Id']==fund),
                                  'PFSustScore'] += score_by_fund[fund]
            non_zero_fund_nb += 1
    print('%d Fonds ont été mis à jour'%non_zero_fund_nb)
    return selected_funds_df

def main(scores_dict, news_by_stock, newsbase, stock_by_funds,
```

```
    stock_names, selected_funds_df, quarter):
"""
Calculates and updates the ESG scores of the funds data.
Parameters
----------
scores_dict: a dict of dict with news indices
as first keys, Positive, Negative and Global
as second keys and score as values (dict of dict of int)
news_by_stock: dictionnary with indices of stocks
in the newsbase as keys and a list of articles
texts as value (dict of lists of str)
stock_by_funds: Dataframe with funds Ids as
columns names, stocks Ids as index and the weight
of each stock in the fund as values
stock_names: Dataframe with stocks Ids as index
and at least a column 'Name' containing the names
of stocks
selected_funds_df: a dataframe containing all the data
of the funds you want to update (Pandas DataFrame)
quarter: the quarter for which you want to update the score (int)
Returns
-------
updated_funds_df: the updated Dataframe with all the funds data
(Pandas Dataframe)
"""
stocks_score = stocks_score_calc(scores_dict, news_by_stock, newsbase)
```

```
funds_score = funds_score_calc(stock_by_funds, stock_names, stocks_score)
updated_funds_df = scores_update(selected_funds_df, funds_score, quarter)
return updated_funds_df
```

A.7 Main script containing all the steps

```
import funds_selection
import stocks_by_fund_retrieval
import news_retrieval
import sentiment_analysis
import score_refining

my_funds_investments_dir = '/Users/gregoiredesauvage/Documents/UNamur/Mémoire
                            /Ressources/Données Fonds/Quarters_investments'
my_agreggated_funds_data_path = '/Users/gregoiredesauvage/Documents/UNamur/Mémoire
                                /Ressources/Données Fonds/Fonds_donnees_agregees.csv'
my_stocks_names_file = '/Users/gregoiredesauvage/Documents/UNamur/
                        Mémoire/Ressources/Données Fonds/Stocks_name.csv'
my_news_dir = '/Users/gregoiredesauvage/Documents/UNamur/
            Mémoire/Ressources/Données News/cc_download_articles'
my_glossaries_path = '/Users/gregoiredesauvage/Documents/UNamur/
                    Mémoire/Ressources/Glossaries/glossary_corpus.json'
my_sentiment_data_path = '/Users/gregoiredesauvage/dev/
                        Mémoire/roberts_cs490_code/data/inqtabs.txt'
my_path_for_save = '/Users/gregoiredesauvage/Documents/UNamur/
                    Mémoire/Ressources/Données Fonds/updated_funds_data.csv'

def main(funds_investments_dir, agreggated_funds_data_path,
         stocks_names_file, news_dir, glossaries_path,
         sentiment_data_path, path_for_save):
    """
```

```
    Parameters
    ----------
    path_for_save: the path where you want to save the updated funds data (str)
    funds_investments_dir: path to the dir containing all the fund investments
    file (str)
    agreggated_funds_data_path: path to the fund data (str)
    stocks_names_file: path to the file containing stock names (str)
    news_dir: path to the directory containing all the news (str)
    glossaries_path: path to the file containing the ESG glossaries (str)
    sentiment_data_path: path to the file containing sentiment dictionary (str)
    """
    aggregated_selected_funds, selected_funds_id =\
    funds_selection.main(funds_investments_dir, agreggated_funds_data_path)
    aggregated_selected_funds.to_csv('/Users/gregoiredesauvage/Documents/UNamur/
                                    Mémoire/Ressources/Données Fonds/
                                    funds_before.csv')
    updated_funds_df = aggregated_selected_funds.copy()
    for quarter in range(5,29):
        selected_stocks, stocks_by_funds, stocks_names =\
        stocks_by_fund_retrieval.main(selected_funds_id, quarter,
                                      stocks_names_file)
        news_by_stock, pertinent_news, esg_news_indices =\
        news_retrieval.main(news_dir, glossaries_path, stocks_names, quarter)
        news_scores = sentiment_analysis.main(sentiment_data_path,
                                              pertinent_news, esg_news_indices)
        updated_funds_df = score_refining.main(news_scores, news_by_stock,
```

```
                                                    pertinent_news, stocks_by_funds,
                                                    stocks_names, updated_funds_df,
                                                    quarter)
        updated_funds_df.to_csv('/Users/gregoiredesauvage/Documents/UNamur/Mémoire
                                /Ressources/Données Fonds/update_q%d.csv'%quarter)
        print('######### Scores Updated #########')
        if quarter == 28:
            updated_funds_df.to_csv(path_for_save)
        print('######### Updated data saved #########')


##### Launch The Function If The File Is Executed #####
if __name__ == "__main__":
    main(my_funds_investments_dir, my_agreggated_funds_data_path,
         my_stocks_names_file, my_news_dir, my_glossaries_path,
         my_sentiment_data_path, my_path_for_save)
```

A.8 R code used for the estimates of subsection 4.6 and section 5

```
#### Libraries import ####
library(haven)
library(PerformanceAnalytics)


#### Data import ####
funds <- read_dta('<path to your fund data file.dta>')
funds_data <- read.csv('<path to your selected fund
                        data file before score refining.csv>')
updated_funds_data <- read.csv('<path to your selected fund
                                data file after score refining.csv>')
funds_returns <- read.csv('<path to your fund returns file.csv>')
funds_returns = na.omit(funds\_returns)
ff_factors <- read.csv('<path to the Fama-French factors file.csv>')
climate_indices <- read.csv('<path to the climate indices file.csv>')


#### Adding Returns to the data ####
for (fund in updated_funds_data$Id) {
  # Retrieve quarters of the current fund
  present_quarters <-
  c(updated_funds_data[updated_funds_data$Id == fund, ]$Quarter)
  # Retrieve returns and add them to data
  updated_funds_data[updated_funds_data$Id == fund, ]$Returns <-
  c(funds_returns[funds_returns$SecId == fund, present_quarters])
  funds_data[funds_data$Id == fund, ]$Returns <-
```

```
  c(funds_returns[funds_returns$SecId == fund, present_quarters])
}
# Returns are in % so convert them into float
funds_data$Returns = as.numeric(updated_funds_data$Returns)/100
updated_funds_data$Returns = as.numeric(updated_funds_data$Returns)/100

#### Processing the Fama-French factors data ####
     #### to be able to aggregate them ####
colnames(ff_factors)[1] = "Date"
ff_factors$Date <- as.Date(strptime(ff_factors$Date, format = "%Y%m%d"))
ff_factors_m <- ts(ff_factors, start=c(2012,4), frequency=12)
# Aggregate the factors at a quarterly level
ff_factors_q <- aggregate(ff_factors_m, nfrequency = 4)
View(ff_factors_q)
# Replace the date by the number of the quarter
for (row in c(4:27)) {
  ff_factors_q[row, 'Date'] = row+1
}
# Removing the first 3 quarters because we need only from the 4th
ff_factors_q = ff_factors_q[-c(1:3), ]

#### Calculate the excess returns ####
for (fund in updated_funds_data$Id) {
  present_quarters <-
  c(updated_funds_data[updated_funds_data$Id == fund, ]$Quarter)-4
  # Excess return is return - risk free rate
```

```
  updated_funds_data[updated_funds_data$Id == fund, ]$Excess_Returns <-
  c(as.numeric(updated_funds_data[updated_funds_data$Id == fund, ]$Returns)
  -as.numeric(ff_factors_q[present_quarters, 'RF']))

  funds_data[updated_funds_data$Id == fund, ]$Excess_Returns <-
  c(as.numeric(funds_data[funds_data$Id == fund, ]$Returns)-
  as.numeric(ff_factors_q[present_quarters, 'RF']))
}

#### Retrieve the only columns needed ####
cleaned_funds_data <-
funds_data[, c('Quarter', 'Id', 'PFSustScore', 'Excess_Returns')]
cleaned_updated_funds_data <-
updated_funds_data[, c('Quarter', 'Id', 'PFSustScore', 'Excess_Returns')]

#### Add climate indices and risk factors to data ####
for (quarter in c(5:25)) {
  cleaned_funds_data[cleaned_funds_data$Quarter == quarter, 'WSJ_AR1'] <-
  as.numeric(climate_indices[climate_indices$Quarter == quarter, 'WSJ_AR1'])

  cleaned_funds_data[cleaned_funds_data$Quarter == quarter, 'CHNEG_AR1'] <-
  as.numeric(climate_indices[climate_indices$Quarter == quarter,
  'CHNEG_AR1'])

  cleaned_updated_funds_data[cleaned_updated_funds_data$Quarter == quarter,
  'WSJ_AR1'] <- as.numeric(climate_indices[climate_indices$Quarter ==
```

```
quarter, 'WSJ_AR1'])

cleaned_updated_funds_data[cleaned_updated_funds_data$Quarter == quarter,
'CHNEG_AR1'] <- as.numeric(climate_indices[climate_indices$Quarter ==
quarter, 'CHNEG_AR1'])

# Retrieve the q-1 factors (+1 to begin at quarter 6, and
# -4 because the indexes are shifted)
cleaned_funds_data[cleaned_funds_data$Quarter == quarter+1, 'Z_hml'] <-
as.numeric(ff_factors_q[quarter-4, 'HML'])

cleaned_funds_data[cleaned_funds_data$Quarter == quarter+1, 'Z_size'] <-
as.numeric(ff_factors_q[quarter-4, 'SMB'])

cleaned_funds_data[cleaned_funds_data$Quarter == quarter+1, 'Z_mkt'] <-
as.numeric(ff_factors_q[quarter-4, 'Mkt.RF'])

cleaned_updated_funds_data[cleaned_updated_funds_data$Quarter == quarter+1,
'Z_hml'] <- as.numeric(ff_factors_q[quarter-4, 'HML'])

cleaned_updated_funds_data[cleaned_updated_funds_data$Quarter == quarter+1,
'Z_size'] <- as.numeric(ff_factors_q[quarter-4, 'SMB'])

cleaned_updated_funds_data[cleaned_updated_funds_data$Quarter == quarter+1,
'Z_mkt'] <- as.numeric(ff_factors_q[quarter-4, 'Mkt.RF'])
```

```
## Calculate the Absolute Score from ESG ratings
quarter_mean1 <-
mean(c(as.numeric(cleaned_funds_data[cleaned_funds_data$Quarter == quarter,
'PFSustScore'])))

cleaned_funds_data[cleaned_funds_data$Quarter == quarter, 'Absolute_Score'] <-
abs(c(as.numeric(cleaned_funds_data[cleaned_funds_data$Quarter == quarter,
'PFSustScore']))-quarter_mean1)

quarter_mean2 <-
mean(c(as.numeric(cleaned_updated_funds_data
[cleaned_updated_funds_data$Quarter==quarter, 'PFSustScore'])))

cleaned_updated_funds_data[cleaned_updated_funds_data$Quarter == quarter,
'Absolute_Score'] <- abs(c(as.numeric(cleaned_updated_funds_data
[cleaned_updated_funds_data$Quarter==quarter, 'PFSustScore']))-quarter_mean2)

## Calculate the Ranked Scores form ESG ratings
quarter_data <- cleaned_funds_data[cleaned_funds_data$Quarter==quarter, ]
# Order the funds for the quarter
ranked_quarter_data <- quarter_data[order(quarter_data$PFSustScore,
decreasing = TRUE), ]
# Assign a rank to the funds for the quarter
ranking <- seq(from = 1, to = nrow(ranked_quarter_data), by = 1)
ranked_quarter_data$Ranking <- ranking
# Retrieve indices of the funds to be able to find them
```

```
  # in the whole dataset
  indices <- rownames(ranked_quarter_data)
  for (i in indices) {
    # For each fund add its ranking for the quarter in the whole dataset
    cleaned_funds_data[i, 'Ranking'] = ranked_quarter_data[i, 'Ranking']
    # Range the ranking to be between [-0.5, 0.5]
    cleaned_funds_data[i, 'Ranked_Score'] =
    (((ranked_quarter_data[i, 'Ranking']-346)/(1-346))*(0.5+0.5))-0.5
  }
  # Do the same for the updated data
  quarter_updated_data <-
  cleaned_updated_funds_data[cleaned_updated_funds_data$Quarter == quarter,]
  ranked_quarter_updated_data <-
  quarter_updated_data[order(quarter_updated_data$PFSustScore,
  decreasing = TRUE), ]
  ranking <- seq(from = 1, to = nrow(ranked_quarter_updated_data), by = 1)
  ranked_quarter_updated_data$Ranking <- ranking
  indices <- rownames(ranked_quarter_updated_data)
  for (i in indices) {
    cleaned_updated_funds_data[i, 'Ranking'] =
    ranked_quarter_updated_data[i, 'Ranking']
    cleaned_updated_funds_data[i, 'Ranked_Score'] =
    (((ranked_quarter_updated_data[i,'Ranking']-346)/(1-346))*(0.5+0.5))-0.5
  }
}
# Above we calculated the scores for the quarter but we
```

```
# use the score of q-1
# Retrieve the score of the previous quarter and assign
# it to the current quarter
for (quarter in c(6:25)) {
  cleaned_funds_data[cleaned_funds_data$Quarter==quarter,
  'Absolute_Score_t_1'] = cleaned_funds_data[cleaned_funds_data$Quarter
  ==quarter-1, 'Absolute_Score']

  cleaned_funds_data[cleaned_funds_data$Quarter==quarter,
  'Ranked_Score_t_1'] = cleaned_funds_data[cleaned_funds_data$Quarter
  ==quarter-1, 'Ranked_Score']

  cleaned_updated_funds_data[cleaned_updated_funds_data$Quarter==quarter,
  'Absolute_Score_t_1'] = cleaned_updated_funds_data[
  cleaned_updated_funds_data$Quarter==quarter-1, 'Absolute_Score']

  cleaned_updated_funds_data[cleaned_updated_funds_data$Quarter==quarter,
  'Ranked_Score_t_1'] = cleaned_updated_funds_data[
  cleaned_updated_funds_data$Quarter==quarter-1, 'Ranked_Score']
}

#### Create the appropriate dataset for estimates with WSJ and CH ####
wsj_data <- cleaned_funds_data[cleaned_funds_data$Quarter %in% c(6:22), ]
wsj_updated_data <-
cleaned_updated_funds_data[cleaned_updated_funds_data$Quarter %in% c(6:22),]
chneg_data <- cleaned_funds_data[cleaned_funds_data$Quarter %in% c(6:25),]
```

```
chneg_updated_data <-
cleaned_updated_funds_data[cleaned_updated_funds_data$Quarter %in% c(6:25),]


#### Calculate the climate risk factors for regressions ####
wsj_data$Z_susa <- wsj_data$Absolute_Score_t_1*wsj_data$Excess_Returns
wsj_data$Z_susr <- wsj_data$Ranked_Score_t_1*wsj_data$Excess_Returns
wsj_updated_data$Z_susa <-
wsj_updated_data$Absolute_Score_t_1*wsj_updated_data$Excess_Returns
wsj_updated_data$Z_susr <-
wsj_updated_data$Ranked_Score_t_1*wsj_updated_data$Excess_Returns
chneg_data$Z_susa <-
chneg_data$Absolute_Score_t_1*chneg_data$Excess_Returns
chneg_data$Z_susr <- chneg_data$Ranked_Score_t_1*chneg_data$Excess_Returns
chneg_updated_data$Z_susa <-
chneg_updated_data$Absolute_Score_t_1*chneg_updated_data$Excess_Returns
chneg_updated_data$Z_susr <-
chneg_updated_data$Ranked_Score_t_1*chneg_updated_data$Excess_Returns


#### In-Sample Estimates ####
# Regress the climate risk indices onto risk factors
# of all the firm over all the quarters
wsj_a_reg <- lm(WSJ_AR1~Z_susa+Z_size+Z_hml+Z_mkt, data=wsj_data)
updated_wsj_a_reg <- lm(WSJ_AR1~Z_susa+Z_size+Z_hml+Z_mkt,
data=wsj_updated_data)
wsj_r_reg <- lm(WSJ_AR1~Z_susr+Z_size+Z_hml+Z_mkt, data=wsj_data)
updated_wsj_r_reg <- lm(WSJ_AR1~Z_susr+Z_size+Z_hml+Z_mkt,
```

```
data=wsj_updated_data)
chneg_a_reg <- lm(CHNEG_AR1~Z_susa+Z_size+Z_hml+Z_mkt, data=chneg_data)
updated_chneg_a_reg <- lm(CHNEG_AR1~Z_susa+Z_size+Z_hml+Z_mkt,
data=chneg_updated_data)
chneg_r_reg <- lm(CHNEG_AR1~Z_susr+Z_size+Z_hml+Z_mkt,data=chneg_data)
updated_chneg_r_reg <- lm(CHNEG_AR1~Z_susr+Z_size+Z_hml+Z_mkt,
data=chneg_updated_data)
summary(wsj_a_reg)
summary(updated_wsj_a_reg)
summary(wsj_r_reg)
summary(updated_wsj_r_reg)
summary(chneg_a_reg)
summary(updated_chneg_a_reg)
summary(chneg_r_reg)
summary(updated_chneg_r_reg)

#### Out-of-sample estimates ####
## For every quarter q we estimate the regression with data from q_min
## (the first quarter) to the quarter q-1
## We then use these estimates to construct the hedged portfolio in q,
## get its returns and calculate the correlation with the climate risk
## index

# Create vectors to store quarter returns
wsji_returns_a <- vector()
wsji_updated_returns_a <- vector()
```

```
wsji_returns_r <- vector()
wsji_updated_returns_r <- vector()
chneg_returns_a <- vector()
chneg_updated_returns_a <- vector()
chneg_returns_r <- vector()
chneg_updated_returns_r <- vector()
# We begin the estimates at quarter 10 because regression
# needs a certain amount of data to be estimated
quarters <- c(10:25)
for (q in quarters) {
  if (q <= 22) {
    # Retrieve the data from first quarter q_min to quarter q-1
    before_q_wsj_data <- wsj_data[wsj_data$Quarter %in% c(6:q-1), ]
    before_q_updated_wsj_data <-
    wsj_updated_data[wsj_updated_data$Quarter %in% c(6:q-1), ]
    q_wsj_data <- wsj_data[wsj_data$Quarter==q, ]
    q_updated_wsj_data <- wsj_updated_data[wsj_updated_data$Quarter==q, ]
    # Estimate the regressions for period q_min to q-1
    oos_wsj_a_reg <-
    lm(WSJ_AR1~Z_susa+Z_size+Z_hml+Z_mkt, data=before_q_wsj_data)
    oos_updated_wsj_a_reg <-
    lm(WSJ_AR1~Z_susa+Z_size+Z_hml+Z_mkt, data=before_q_updated_wsj_data)
    oos_wsj_r_reg <-
    lm(WSJ_AR1~Z_susr+Z_size+Z_hml+Z_mkt, data=before_q_wsj_data)
    oos_updated_wsj_r_reg <-
    lm(WSJ_AR1~Z_susr+Z_size+Z_hml+Z_mkt, data=before_q_updated_wsj_data)
```

```
# Calculate weights of funds in the portfolio for quarter q
# For the absolute score
w_a <- oos_wsj_a_reg$coefficients[2]*q_wsj_data$Z_susa +
oos_wsj_a_reg$coefficients[3]*q_wsj_data$Z_size +
oos_wsj_a_reg$coefficients[4]*q_wsj_data$Z_hml+
oos_wsj_a_reg$coefficients[5]*q_wsj_data$Z_mkt

updated_w_a <-
oos_updated_wsj_a_reg$coefficients[2]*q_updated_wsj_data$Z_susa +
oos_updated_wsj_a_reg$coefficients[3]*q_updated_wsj_data$Z_size +
oos_updated_wsj_a_reg$coefficients[4]*q_updated_wsj_data$Z_hml +
oos_updated_wsj_a_reg$coefficients[5]*q_updated_wsj_data$Z_mkt
# For the ranked score
w_r <- oos_wsj_r_reg$coefficients[2]*q_wsj_data$Z_susa +
oos_wsj_r_reg$coefficients[3]*q_wsj_data$Z_size +
oos_wsj_r_reg$coefficients[4]*q_wsj_data$Z_hml+
oos_wsj_r_reg$coefficients[5]*q_wsj_data$Z_mkt

updated_w_r <-
oos_updated_wsj_r_reg$coefficients[2]*q_updated_wsj_data$Z_susa +
oos_updated_wsj_r_reg$coefficients[3]*q_updated_wsj_data$Z_size +
oos_updated_wsj_r_reg$coefficients[4]*q_updated_wsj_data$Z_hml +
oos_updated_wsj_r_reg$coefficients[5]*q_updated_wsj_data$Z_mkt

q_wsj_data$Weights_a <- w_a
```

```
  q_updated_wsj_data$Weights_a <- updated_w_a
  q_wsj_data$Weights_r <- w_r
  q_updated_wsj_data$Weights_r <- updated_w_r
  # Calculate excess returns of portfolio for quarter q
  r_a <- sum(q_wsj_data$Excess_Returns*q_wsj_data$Weights_a)
  updated_r_a <-
  sum(q_updated_wsj_data$Excess_Returns*q_updated_wsj_data$Weights_a)
  r_r <-
  sum(q_wsj_data$Excess_Returns*q_wsj_data$Weights_r)
  updated_r_r <-
  sum(q_updated_wsj_data$Excess_Returns*q_updated_wsj_data$Weights_r)
  # Add return of the portfolio
  wsji_returns_a = append(wsji_returns_a, r_a)
  wsji_updated_returns_a = append(wsji_updated_returns_a, updated_r_a)
  wsji_returns_r = append(wsji_returns_r, r_r)
  wsji_updated_returns_r = append(wsji_updated_returns_r, updated_r_r)
}
# Retrieve the data from first quarter q_min to quarter q-1
before_q_chneg_data <- chneg_data[chneg_data$Quarter %in% c(6:q-1), ]
before_q_updated_chneg_data <-
chneg_updated_data[chneg_updated_data$Quarter %in% c(6:q-1), ]
q_chneg_data <- chneg_data[chneg_data$Quarter==q, ]
q_updated_chneg_data <- chneg_updated_data[chneg_updated_data$Quarter==q,]
# Estimate the regressions for period q_min to q-1
oos_chneg_a_reg <-
lm(WSJ_AR1~Z_susa+Z_size+Z_hml+Z_mkt, data=before_q_chneg_data)
```

```
oos_updated_chneg_a_reg <-
lm(WSJ_AR1~Z_susa+Z_size+Z_hml+Z_mkt, data=before_q_updated_chneg_data)
oos_chneg_r_reg <-
lm(WSJ_AR1~Z_susr+Z_size+Z_hml+Z_mkt, data=before_q_chneg_data)
oos_updated_chneg_r_reg <-
lm(WSJ_AR1~Z_susr+Z_size+Z_hml+Z_mkt, data=before_q_updated_chneg_data)

# Calculate weights of funds in the portfolio for quarter q
# For the absolute score
w_a <- oos_chneg_a_reg$coefficients[2]*q_chneg_data$Z_susa +
oos_chneg_a_reg$coefficients[3]*q_chneg_data$Z_size +
oos_chneg_a_reg$coefficients[4]*q_chneg_data$Z_hml+
oos_chneg_a_reg$coefficients[5]*q_chneg_data$Z_mkt

updated_w_a <-
oos_updated_chneg_a_reg$coefficients[2]*q_updated_chneg_data$Z_susa +
oos_updated_chneg_a_reg$coefficients[3]*q_updated_chneg_data$Z_size +
oos_updated_chneg_a_reg$coefficients[4]*q_updated_chneg_data$Z_hml +
oos_updated_chneg_a_reg$coefficients[5]*q_updated_chneg_data$Z_mkt
# For the ranked score
w_r <- oos_chneg_r_reg$coefficients[2]*q_chneg_data$Z_susa +
oos_chneg_r_reg$coefficients[3]*q_chneg_data$Z_size +
oos_chneg_r_reg$coefficients[4]*q_chneg_data$Z_hml +
oos_chneg_r_reg$coefficients[5]*q_chneg_data$Z_mkt

updated_w_r <-
```

```
  oos_updated_chneg_r_reg$coefficients[2]*q_updated_chneg_data$Z_susa +
  oos_updated_chneg_r_reg$coefficients[3]*q_updated_chneg_data$Z_size +
  oos_updated_chneg_r_reg$coefficients[4]*q_updated_chneg_data$Z_hml +
  oos_updated_chneg_r_reg$coefficients[5]*q_updated_chneg_data$Z_mkt

  q_chneg_data$Weights_a <- w_a
  q_updated_chneg_data$Weights_a <- updated_w_a
  q_chneg_data$Weights_r <- w_r
  q_updated_chneg_data$Weights_r <- updated_w_r
  # Calculate excess returns of portfolio for quarter q
  r_a <- sum(q_chneg_data$Excess_Returns*q_chneg_data$Weights_a)
  updated_r_a <-
  sum(q_updated_chneg_data$Excess_Returns*q_updated_chneg_data$Weights_a)
  r_r <- sum(q_chneg_data$Excess_Returns*q_chneg_data$Weights_r)
  updated_r_r <-
  sum(q_updated_chneg_data$Excess_Returns*q_updated_chneg_data$Weights_r)
  # Add return of the portfolio
  chneg_returns_a = append(chneg_returns_a, r_a)
  chneg_updated_returns_a =
  append(chneg_updated_returns_a, updated_r_a)
  chneg_returns_r = append(chneg_returns_r, r_r)
  chneg_updated_returns_r =
  append(chneg_updated_returns_r, updated_r_r)
}

# Retrieve the WSJ index from the 10th quarter to calculate correlations
```

```r
wsji <- climate_indices[climate_indices$Quarter %in% c(10:22), 'WSJ_AR1']
# Calculate correlation
wsji_cor_a <- cor(wsji_returns_a, wsji)
wsji_updated_cor_a <- cor(wsji_updated_returns_a, wsji)
wsji_cor_r <- cor(wsji_returns_r, wsji)
wsji_updated_cor_r <- cor(wsji_updated_returns_r, wsji)
print(wsji_cor_a)
print(wsji_updated_cor_a)
print(wsji_cor_r)
print(wsji_updated_cor_r)


# Retrieve the CH index from the 10th quarter to calculate correlations
chneg <- climate_indices[climate_indices$Quarter %in% c(10:25),'CHNEG_AR1']
# Calculate correlation
chneg_cor_a <- cor(chneg_returns_a, chneg)
chneg_updated_cor_a <- cor(chneg_updated_returns_a, chneg)
chneg_cor_r <- cor(chneg_returns_r, chneg)
chneg_updated_cor_r <- cor(chneg_updated_returns_r, chneg)
```