Financial Programming I

Programming Project 2019

15.

Twitter and the cross

section of stock returns

Matteo Corti, Andrea Giacobbo, Simone Giay, Filippo Testa

Università della Svizzera italiana

Master in Finance

Lugano, Ticino, Switzerland, 2019

Table of Contents

1-Introduction

The focus of this paper is a cross sectional analysis of returns based on the 80% of the SP500 stocks. In order to perform this analysis, we focused in finding cyclicality in the series of the number of tweets. Then, we created equally weighted portfolios of over and under-tweeted companies, as well as of bullish and bearish companies and we tracked their subsequent performances in order to verify how they performs with respect to the other and the market.

2-The program

To perform this type of analysis R is a good software combined with some packages, among which the crucial is "*twitteR*" that allows us to communicate with Twitter. Moreover, we needed the weights and the returns of the companies corresponding to the 80% of the SP500, easily accessible with the "*tidyquant*" package. At last, in order to analyze the polarity(sentiment) of the tweets, we used a package called "*sentimentr*".

2.1-The twitter API

An API (acronym for Application Programming Interface) is by definition a software intermediary that allows two applications to talk to each other. First of all, in order to use the Twitter API, we created a twitter account and we obtained access to the developer options at "*https://apps.twitter.com*". At this point, it is necessary to fill an application form, where the twitter team ask for what reason we are going to use the twitter API. At the end of these preliminary operations, we obtained the credentials to access the application via R (Consumer key, Consumer Secret, Access Token, Access Secret). Then, in order to create a communication between R and twitter we installed a package called "*twitteR*". Furthermore, we communicated the credential created before to R by using the function "*setup_twitter_oauth*". At this point, the R package is ready to communicate and access to twitter data.

3-Prepare the environment

3.1-Instructions to run the program

In order to run the program, the user has to set the value of some variables:

- "*Bbdate_A_U*": The user sets the date in which starts the bullish and bearish and the over and under-tweeted analysis. We remark that the Twitter API doesn't allow to access data older then about a week, so, the user must insert a value in this range.
- "*number_days_return_analysis*": Number of days in which the user wants to track the performances of the portfolios.
- "*num_tweet_A*": Number of tweets to research.

3.2- Obtaining the weights of the SP500

We used the function "*tq_index*" (from the package "*tidyquant*") in order to get all stocks in the SP500 and we stored them the variable "*all_stocks*". We limited the analysis to the current 80% (it's important to remark that the data are collected when the program is executed) of the firms in the index because using all the 500 firms can be computational quite challenging.
For further analysis it is necessary to compute the relative weights. These are obtained by dividing the absolute weight of the SP500 by the 80%.

3.3- The "Cashtag"

The function "*searchTwitter*" uses a feature of Twitter very similar to the hashtag that allows the user to know information about what the Twitter community is saying about firm's stocks. This is called

"Cashtag". It is composed by the "$" symbol ahead the company's ticker. We stored the results in the "Cashtag" vector.

3.4- The stocks returns

To track the performance of the portfolios in necessary to download the returns of the SP500.
To get the returns of each stock we used the function "*getsymbols*" (from the package "*quantmod*") that retrieve the prices of the stocks; this is a generic function in which we need to specify a provider where to collect the data; in our case, the source of the prices is *"yahoo.finance"*. Furthermore, we computed the returns using the prices.

3.5- The market portfolio

In order to evaluate the performances of the analyzed portfolios we compared them with a benchmark, formed by the returns of the "Market portfolio".

$$return\ market\ portfolio = \sum return\ stock_i * absolute\ weight_i$$

4- Analysis on the cyclicality

To have a deeper analysis of our time series, we analyzed components of trend, seasonality and cyclicality. Starting from our database, we produced a new time serie with the total daily number of tweets.

4.1-Data generating process

In order to allow the user to have an overall view to carry out an advanced analysis of the time series, it is necessary to have data of an enough wide period.
We faced different problems in the data generating process due to the limitations of the "*searchTwitter*" function:

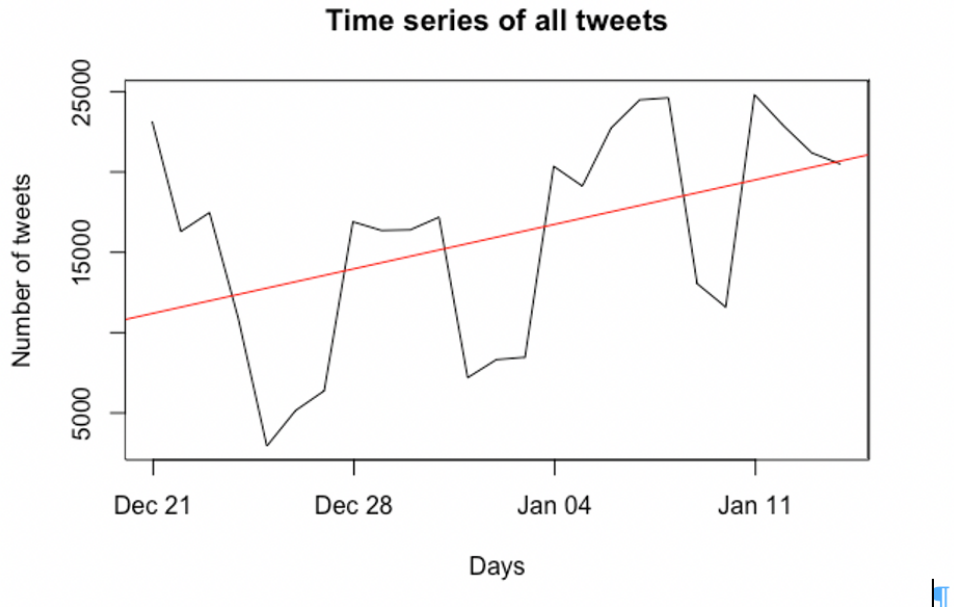
- Twitter website denies the access to the data older than about a week;
- according to the twitter API documentation, the rate limit of tweets that can be requested is 180 or 300 requests every 15 minutes; therefore, the function "*Sys.sleep*" suspend the execution of R expressions for a specified time and help to overcome this problem.

We solved this problem by creating our own database, retrieving the number of Tweets on a weekly basis from the 21st of December to 14th of January. To collect the data, we got the weights at the beginning of the period and we used them for the whole process. The main reason is to avoid the problem that companies forming the 80% of the SP500 and their own weights, could change over time. We hide the code (with #) that we used to download past data, leaving the possibility to the user to work on the database created.

4.1.1- Download the Tweets and export to excel

To fill the database, we carry out the analysis with a maximum number of tweets of 20000 to be sure to catch all the tweets of each company. Additionally, we removed all the retweets (that could create redundant and useless data) for each company. Furthermore, the "if" statement is useful to avoid any mistake coming from the usage of the function "*strip_retweets*" on an empty value. The results of the analysis are contained in the matrix "*tweets_Matrix*". Then, we exported the results of the weekly analysis in an external excel file that can be easily accessible by the user to analyze the time series trough the function "*read_excel*" (from the package "*readxl*"). Instead, we used the function "*convertToDate"* from the package "*openxlsx*" to convert the data in the right format.

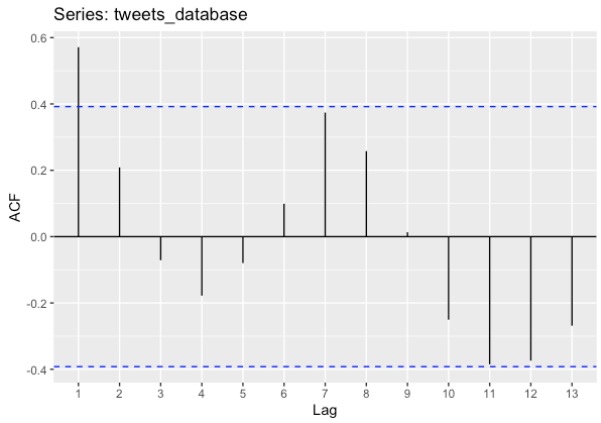
4.1.2- Perform the analysis



Our concern was to compute some statistics indicators to have useful parameters for the analysis.

Mean	Min	Max	Sd	Variance	Mean Abs Dev
15979.4	2964	24864	6705.125	44958703.8	8656.901

First of all, we plotted the time series of all tweets, then we used a linear regression model to find the trend component. The trend line (the red line) appears positively inclined; however, we know that the set of data is limited only to about 20 days, and the outputs that we get are not likely to have significant relevance. We can suppose that this trend could be part of a bigger cycle that we cannot observe.
Another analysis on the time series is the seasonality, that we decided to inspect graphically on the pattern of the total number of tweets. We cannot talk about a precise seasonality, since we are only analyzing 25 days. However, we can observe that the number of tweets consistently decrease in holiday days. In fact, we have lower peaks during Christmas and on the first days of January.



As regarding the cyclicality, the function "*ggAcf*" could be a useful indicator. The function computes the autocorrelation existing between the tweets of consecutive days. It produces a histogram and a range. If the value of the autocorrelation of days is outside the range, it means that there is a significant autocorrelation between how tweets are distributed.

From the results there is no evidence of cyclicality, even if the values are close to escape the range. Only the first period shows there is a significant autocorrelation between the tweets. We can't assure that there is no autocorrelation at all, but just that future tweets are not predictable with linear methods (there could be non-linear methods which could give better results such as exponential smoothing).
Finally, we computed the Lyung-box test to inspect the randomness of data. In case of a significant p-value, we reject the null hypothesis that the data are random (*White Noise*).
We used the function "*Box.test*" specifying the type of test (Lyung). For a p-value of 0,05 we can reject the null hypothesis. The result is better than we expected, a p-value of 0,0024 but not enough to reject the null hypothesis.

5- Portfolio analysis

We created multiple portfolios with different characteristics to study if these features are relevant over time in the returns.

5.1-Data generating process

The process to retrieve the data is very similar to the one explained in the paragraph 4.1. For this analysis is necessary to take into account not only the number of tweets but also their contents. We remark that the user can perform this analysis only for a weekly period.

5.2- Over and under tweeted portfolio analysis

One of the most reliable indicators that we should take into account in a cross-section analysis is the number of tweets. We divided the firms of our "market portfolio" into two equally weighted portfolios, and we tracked their subsequent performances.
In order to compare the "market share" of tweets to the index weight, we firstly calculated the "twitter index". This index is composed by the weights of all the companies included in the analysis. These weights were calculated as follow:

$$weight_i = \frac{n.tweets\ company\ i}{total\ sum\ of\ the\ tweets}$$

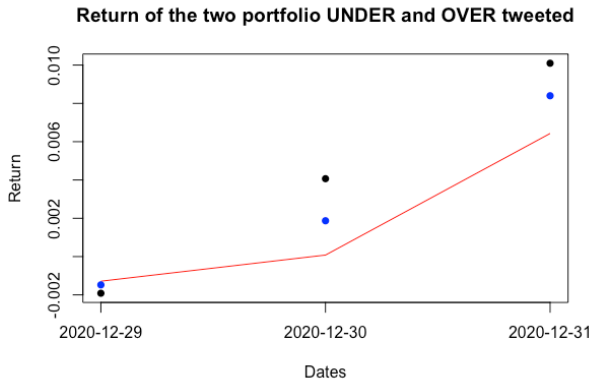
Thereafter, we proceeded with the construction of the "over-tweeted" and "under-tweeted" portfolios by comparing the values of the "twitter index" and those in our "market portfolio".
Specifically, if the value of the "twitter index" is greater than the one of the S&P500, the company will be classified as "over-tweeted" and therefore included in the corresponding portfolio and vice versa in the opposite case. After creating the two equally weighted portfolios (each company weighs 1/n in the portfolios), we calculated the returns on a daily basis.

5.2.1- Results of the analysis

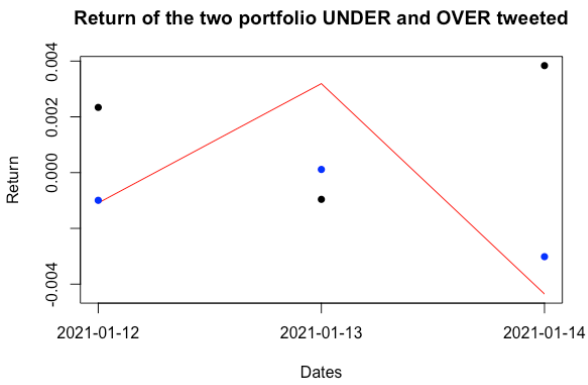
In order to compute a more precise and reliable analysis, we decided to study the trend of the returns of the different portfolios in three set of dates.
For the first analysis we have divided the firms in three different portfolios during the Christmas holidays period and evaluated the trend of returns in the subsequent days.

We gave the user the possibility to plot the graph of the returns in the different dates and visualize it to better study the data.
The results of the analysis showed that in this period the number of firms in the over-tweeted portfolio was 107 and 59 for the under-tweeted one, for the other dates, 48 of over and 118 for the under.

26 December 2020



10 January 2021



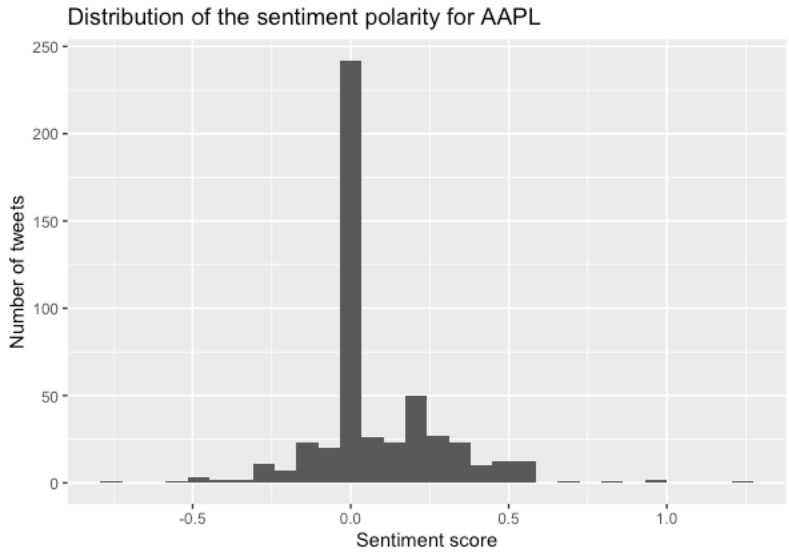
We plotted the returns of the over-tweeted portfolio (the black dot) and under-tweeted portfolio (the blue dot) and then we compared it to the returns of the market (the red line). In the first period the portfolio follows the trend of the market, we can't state the same for the second period. In fact, the pattern of the under-tweeted portfolio is opposite to the market one.
In particular, it's interesting to notice that when returns are positive, the over-tweeted portfolio performed better than the under-tweeted in both periods.
Our consideration is confirmed by the average returns of the different portfolio over the period, the over-tweeted's performance exceeds.

5.3.1- Sentiment analysis

In order to find a way to incorporate contents in the form of "bullish" and "bearish" tweets, we have decided to use the "*Sentimentr*" package. The package allows to compute a quick analysis of the contents of the tweets, correcting also the problem of some other packages, the correction of inversions (ex. I'm not good).
The function "*sentiment_by*" inspect and store the aggregate polarity (sentiment) score of the tweet.
The function gives us different measures, but we've focused on the "*ave_sentiment*", the average sentiment/polarity score of the review.
We have computed an analysis on the single tweet, labelling the single tweet as "bullish" if the average polarity score is higher than zero and bearish in the other case. If the score value of the sentiment analysis is 0, the content is "neutral" or the function can't give a score to the tweet.
In our analysis we haven't considered these neutral tweets neither bullish neither bearish.
In addiction we cleared all the tweets from all the character non relevant for our analysis such as alphanumeric characters, punctuation, and links…
Finally, we stored the *ave_sentiment* value in a matrix ("sentiment_matrix"). We used the simplistic condition that if the number of bullish tweets is bigger than number of bearish tweets then the firm is considered bullish.

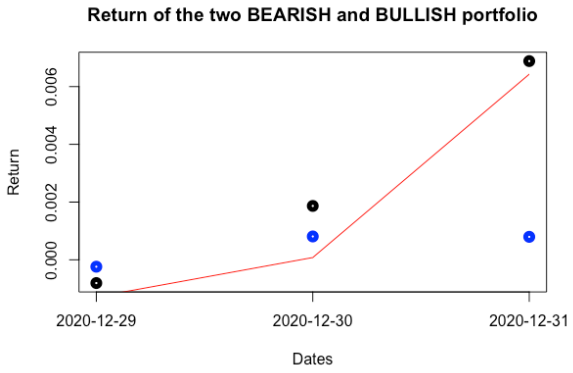
5.3.2-Results of the analysis

Showing the distribution of the polarity for all the firms could have been excessive, therefore we only plotted the sentiment tweets of Apple (the biggest in the S&P500) as example.

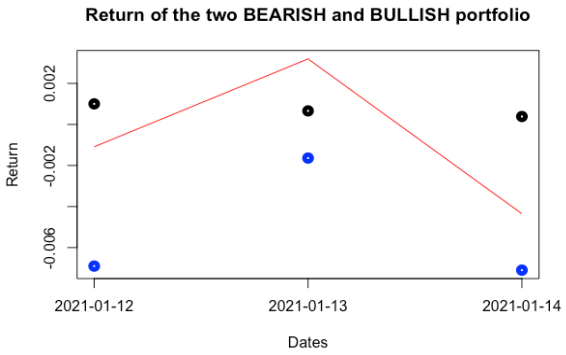


As we can see in the graph, there is a consistent number of tweets (almost 250) for which the sentiment package can't assign a score (neither positive nor negative). However, R classified Apple as a bullish firm, and as can be seen from the graph, the right tail is bigger, meaning that a larger number of tweets were assumed to be positive.

26 December 2020



10 January 2021



We have firstly analyzed, as for the under and over-tweeted firms, the period right after Christmas. The results of the analysis showed that in this period the number of firms in the bullish portfolio was 122, 19 for the bearish and 25 "neutral" (which was not considered in the analysis since they were neither bullish nor bearish) while in the second period there are 139 bullish, 11 bearish and 16 neutral.
The graph compared the return of the portfolios over time; it has in the x axis the dates while in the y axis the returns, of the bullish portfolio (the black dot) and bearish portfolio (the blue dot), compared to the returns of the market (the red line).
Even though there were limited set of data and the sentiment analysis packages was not so accurate, our analysis seems to be precise since the portfolio with positive polarity on average outperformed the market.
In both the periods, the return of bullish portfolio is over the bearish one. If we focus in the second period the bearish portfolio follows the trend of the market, but with lower returns.
Moreover, we can say the same for the bullish portfolio especially in the first period, but with positive returns giving an evidence of significance of our sentiment analysis.

5- Key results

The results obtained cannot be considered satisfying due to all the limitations of the Twitter API and of the dimension of the data sample. In fact, both the cyclicality and the portfolio performance analysis gave us encouraging results but we need a more consistent database in order to verify with confidence our hypothesis.

6- References

- https://towardsdatascience.com/setting-up-twitter-for-text-mining-in-r-bcfc5ba910f4
- https://www.lexalytics.com/technology/sentiment-analysis
- https://towardsdatascience.com/doing-your-first-sentiment-analysis-in-r-with-sentimentr-167855445132
- https://cran.r-project.org/web/packages/twitteR/README.html
- https://cran.r-project.org/web/packages/tidyquant/index.html
- https://cran.r-project.org/web/packages/tidyverse/index.html
- https://cran.r-project.org/web/packages/quantmod/index.html
- https://cran.r-project.org/web/packages/plyr/index.html
- https://cran.r-project.org/web/packages/dplyr/index.html
- https://cran.r-project.org/web/packages/sentiment/index.html
- https://cran.r-project.org/web/packages/readxl/index.html
- cran.r-project.org/web/packages/forecast/index.html
- https://cran.r-project.org/web/packages/ggfortify/index.html
- https://cran.r-project.org/web/packages/survMisc/index.html
- https://cran.r-project.org/web/packages/labstatR/index.html

Our excel database is attached, with the name "*Daily_time_series_of_all_tweets.xlsx*", and the company for which the data are referred, in the file "*Companies.xlsx*".