SCHOOL OF
HUMAN SCIENCES
& TECHNOLOGY

VaR calculation

Data Engineering Group Project

Group B:

- Alexander Verresen
- Borja Pérez
- Giuseppe Pepe
- Iurii Fedotov

Program: Master in Computer Science and Business Technology

TABLE OF CONTENTS:

2019 | Madrid | Data Engineering

1. General approach

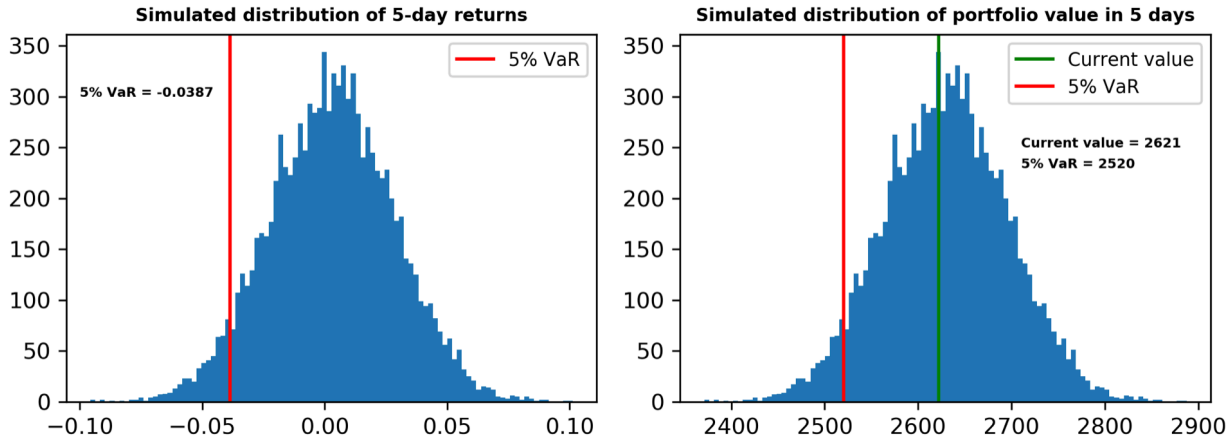
Our approach was the following:

1. Read the Symbols.txt file.
2. Write a function that takes a stock symbol as a parameter, uploads the dates and adjusted close prices for the maximum available time period from the Alpha Vantage API. We use adjusted close price, not nominal close price, to take splits into account.
3. Write functions which upload dates and values of oil prices and 10-year treasury bonds rate from Quandl API. We use the 10-Year bond, because it is the first security in the world in terms of trading volume. In addition, bonds with shorter maturity by definition to not fluctuate much according to the mood in the stock market, because their principal payment comes soon.
4. Write a function that aggregates steps 2 and 3 of this list, and prepares a dataframe of all stocks from the symbols list, together with macroeconomic factors. VERY IMPORTANT: the Symbols.txt file contained 50 symbols of stocks. However, only 34 of them were available in the API service. For those which are not there, even the Yahoo Finance pages do not exist, which means that these stocks are either traded on low-tier local exchanges, or there are typos in the Symbols.txt. **We included 34 stocks in our portfolio** (only those for which we managed to get the data from the API), and followed the rule of assuming 1 unit of each stock in the portfolio.
5. Save all the data retrieved from the API to a local .csv, because uploading from .csv is much faster in the future.
6. Import this .csv back to the program.
7. Deal with NaNs (specific approach will be explained in chapter 2 of this report).
8. Switch from absolute values to returns. All ML models used in this assignment will be aiming to predict the 5-day return of a stock, using 5-day returns/changes in macroeconomic features. We use returns instead of absolute values for obvious reason - absolute values change through time and have different orders for different factors. All models in finance use returns to predict other returns.
9. Validate that macroeconomic features' returns follow normal distribution. This validation will be demonstrated in chapter 3 of this report.
10. Simulate 10,000 values of macro factors' returns using the multivariate normal distribution.
11. Predict 10,000 values of returns for each individual stock. The details of models used in this step will be explained in chapter 3 of this report.
12. Calculate the current portfolio value and weights of each stock.

13. Predict 10,000 values of overall portfolio return and value in a 5-day horizon. The return of the whole portfolio is the weighted average return of securities included in it, where weights are proportions of values of specific securities in the whole portfolio value. Therefore, the portfolio overall return can be computed as the sum of products of individual returns and weights.
14. Plot our findings and make conclusions.

2. Conclusions

Figure below demonstrates the results of our Monte Carlo simulation.



The conclusions are:

- The 5-day 5% VaR of the overall portfolio return, given its current structure, is -3.87%.
- The current value of the portfolio is $2,621. The 5-day 5% VaR of the overall portfolio value, given its current structure, is $2,520.

3. Aligning different data sources and dealing with NaNs

3.1. Aligning different sources

Here are the data sources we used:

- Prices of all stocks and S&P index values were uploaded from Alpha Vantage API.
- Oil prices and treasury bonds rates were uploaded from Quandl API.
- Values of NASDAQ composite index were uploaded from Yahoo Finance as .csv and imported from .csv. The ticker "NDAQ" provided in the assignment is not the NASDAQ index! It is a ticker of the stock of NASDAQ stock exchange as a corporation. Unfortunately, the NASDAQ index is not available at Alpha Vantage API, we checked it in many forums and many people reported this issue.

Technically, the alignment was performed with pandas MERGE function, where all the data was merged in a form of left outer join to the NASDAQ index values, by date. This approach guarantees that no data was missed along the way, because NASDAQ index has non-NaN values for all trading days.

3.2. Dealing with NaNs

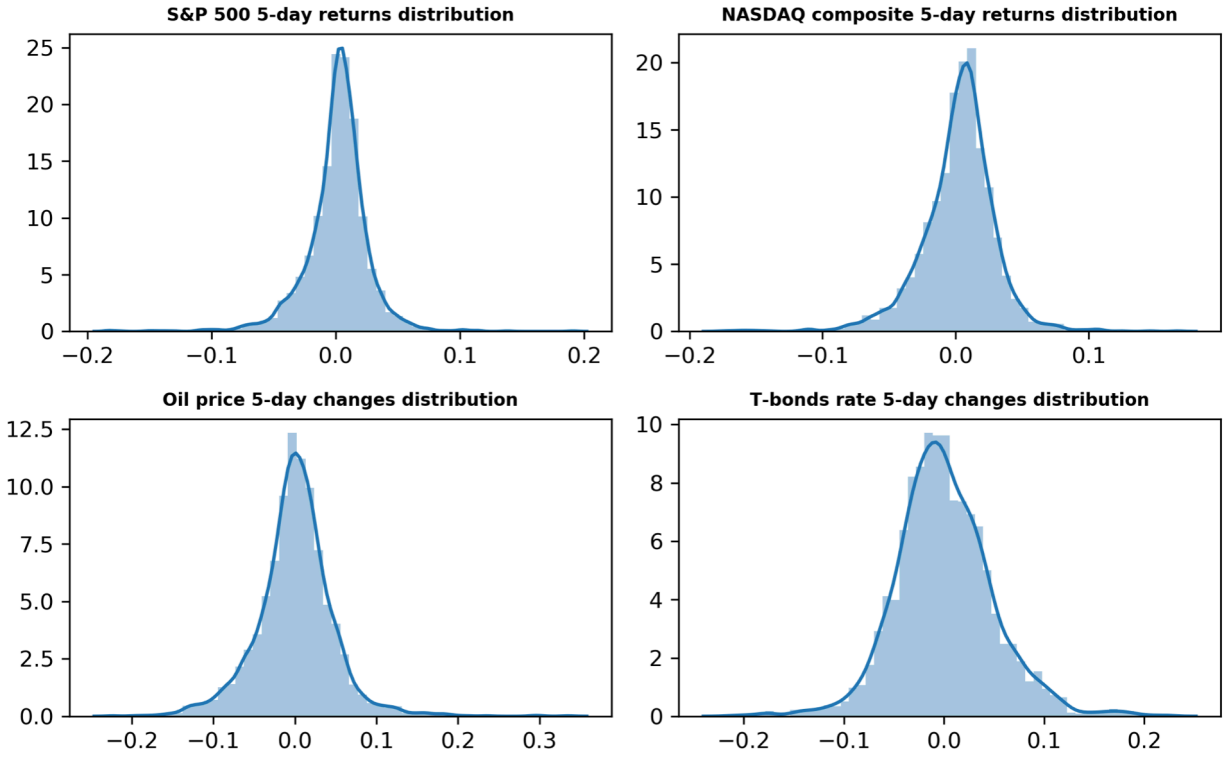
Findings: some columns contained NaNs, but even for worst of them, NaNs were ~56% of the total observations.

Decision:

- In macro factors, all NaNs were surrounded by true values, thus giving us an opportunity to **linearly interpolate the NaNs**. And that was what we did.
- In stocks, all NaNs were outside of the true values - for example, when the stock was not public yet, or when it stopped being traded for some reason. Therefore, we cannot interpolate missing values. Considering that all stocks have at least 1,000 real values of price, we decided **leave NaNs as they are and train a model for each stock only on the data which has true values for this stock.** Filling NaNs with some guessed values seems worse than reducing the dataset size from 2,800 observations to, in the worst case, 1,300. Even 1,300 observations are more than enough to build a linear regression with a few features. The advantage of this approach is that if stock A has 50% of missing values, and stock B has only 10%, stock A uses the most data available for it (50%), and stock B too – 90%.

4. Normal distribution of factors' returns: validation

The figure below validates the normal distribution of factors' returns:



5. Feature engineering: adding squared and root features

Our approach to feature engineering and feature selection:

- Since our models use returns as features, **we cannot use square roots**, since they do not exist for negative numbers. And returns can be negative.
- However, for each stock, we compared the performance of 2 models. First of them uses just 4 standard features linearly, and the second one includes the squares of the given features. Our program automatically compares the cross-validated R2 of both models for all stocks, and for each particular stock it selects the model type which has a higher R2.

Point for improvement which are out of scope of the assignment:

- Add more factors (for example, index of economic uncertainty, or the volume of trade).
- Remove obviously correlated features, like S&P500 and NASDAQ returns.
- Use other models than linear regression.
- Use models specific to time series data: ARIMA, for example.

6. Approach to parallelization with Spark

According to your email, we do not need use Spark in this assignment, and just need to explain how we would do the parallelization in theory. Our approach would be:

- Create a SparkContext object (commonly aliased as **sc**), which tells Spark how to access a cluster.
- Since we do not process, but create the data in parallel, we need to use some data structure and convert it to RDD which will then be transformed to meaningful simulations that we need. We can use a 10,000 X 4 matrix of zeros (created instantly with np.zeros()) for this purpose.
- Using **sc.parallelize**, we map the entire matrix to the random numbers generated from multivariate distribution by NumPy's function.
- In the end, we receive a 10,000 X 4 matrix with the simulated values which we need.