Libera Università Internazionale degli
Studi Sociali Guido Carli

LUISS

PREMIO TESI D'ECCELLENZA

A Study Of The Psychological Effect Of Covid-19 On Stock Prices

Marco Costantini

4 2019-2020

LUISS
Academy

LUISS Guido Carli / Premio tesi d'eccellenza
Working paper n. 4/2019-2020
Publication date: november 2021
A Study Of The Psychological Effect Of Covid-19 On Stock Prices

ISBN 9788861057432

LUISS Academy is an imprint of
LUISS University Press – Pola Srl
Viale Pola 12, 00198 Roma
Tel. 06 85225485
e-mail lup@luiss.it
www.luissuniversitypress.it

A study of the psychological effect of Covid-19 on stock prices

Marco Costantini

Abstract

Does the market behave rationally in times of crisis? Does it value stocks using the same methods used in times of normal market conditions? What has happened in the stock market during the COVID-19 period?

Under normal circumstances, the market evaluates companies by giving the most attention to the future cash flows, to prices paid in precedent transactions, to ratios of peer companies in the same industry, and so on. However, during the COVID-19 crisis, investors partially changed the way they valued companies. As a matter of fact, the risk of bankruptcy and the consequent fear to lose most or the entirety of their investments has had a predominant effect on investors, and in turn, on stock prices. Then, why did investors behave irrationally? Because of a widespread sell-off, alimented by the chaos introduced in the market by the stringent measures adopted by the various countries (such as lockdowns) to contain the spreading of the virus, inducing investors to overestimate the probability of default inherent in some companies or to just adopt a panic-sell behavior. Such a phenomenon caused valuations of stocks to decrease drastically until the 23rd of March, when, in the United States, the FED started to directly purchase securities in the market, hence increasing the liquidity and decreasing the fear of investors.

By looking at the rating of the companies as for the S&P (External Credit Assessment Institution), the stocks of the S&P500 are separated into buckets. Then, the maximum-Sharpe-Ratio portfolio is selected for each group. A comparative analysis is implemented to see how these portfolios have performed relative to each other. The objective is to assess whether those stocks which were not as risky as the market thought during the market crash in February and March were among the top performers, in terms of stock price change, in the months following the decline.

Subsequently, the same portfolios are analyzed in different years to understand whether the cross-portfolio results are just due to the structure of the market, or if, oppositely, they have been the consequence of the panic-sell.

I expect the return of the portfolio with the highest Issuer Credit Rating to be the highest (in terms of Sharpe Ratio), considering that these stocks have been hugely penalized by investors without a fundamental reason.

1. The predominance of irrationality in stock prices during the COVID-19 recession period

In this section, the stock price performance is analyzed for 373 stocks of the S&P500. The stocks are splitted into 6 different groups based on their Long-term Issuer Credit Rating (as of the end of June 2020) as for the Standard & Poor ECAI.
The choice of the credit rating as the metric used for the split is dictated by the need of creating portfolios depending on the financial soundness of the stocks considered. The aim is to build different portfolios based on the financial quality of the companies.
During the Covid-19 crisis, the market has implemented a sell-off process which has impacted almost every stock in the market. Investors started to drastically increase their perception of the probability of default of these companies, with different behavioral factors contributing to the market decline (loss aversion, herding behavior, framing, and mental accounting).
This analysis aims at analyzing the risk-adjusted returns of the different portfolios to understand whether financially sound companies performed better than others. The underlying idea is that, considering the several behavioral factors involved in the market decline, investors did not base their decisions on a rational evaluation of the fundamentals and prospects of their stocks. Contrarily, they took their decisions based on fear, ultimately selling stocks that should have not been penalized to that extent.
The credit rating is a solid measure to perform an analysis of the soundness of the company. This measure is a forward-looking opinion on the obligor's overall creditworthiness, which can be summarized by the ability and willingness to meet its liabilities when they come due. The importance of such rating is also due to the material due diligence that the ECAI conduce, which involves the judgment of not only public but also private information, which is normally not available to investors.
Consequently, the 373 stocks selected from the S&P500 were divided into groups with a number of stocks that ranges between 48 and 76, depending on the availability of stocks for certain credit ratings. The 6 groups are defined as follow:

1. AAA to A: 70 stocks;
2. A- : 54 stocks;
3. BBB+ : 76 stocks;
4. BBB: 72 stocks;
5. BBB- : 48 stocks;
6. BB+ to B- : 53 stocks.

1.1 Portfolio composition and optimization in Python

The analysis described below is integrally performed using Python programming in Anaconda. The tickers and the Long-term Issuer Credit Ratings are downloaded for the stocks of the S&P500 (data coming from Bloomberg). The data is saved on an excel file, which is then uploaded on Python to check the availability of stock price data of the single stocks. Daily stock prices are available for 373 stocks on the list.

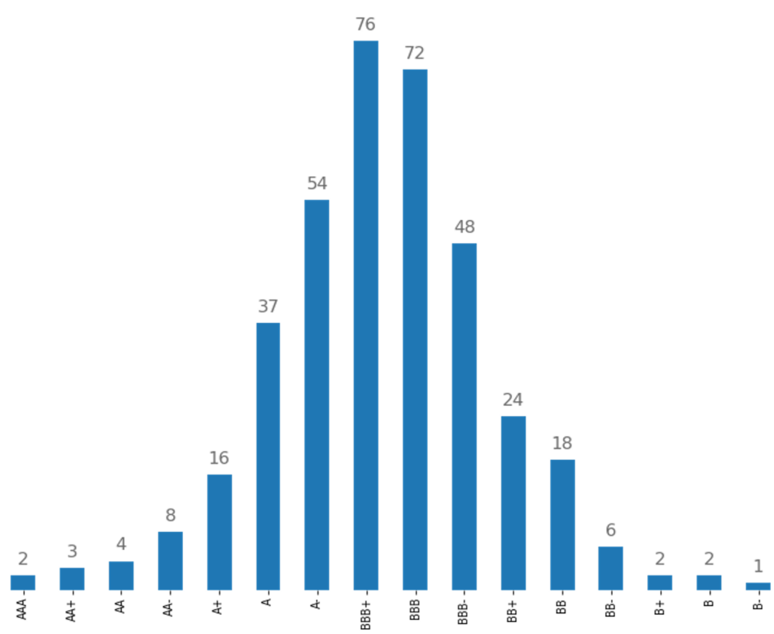
The next step is to decide how to split the different stocks to create groups of similar dimensions. To do that the following piece of code is used:

```
# Import all the necessary packages and functions for the whole code

% matplotlib inline

import numpy as np

import matplotlib.pyplot as plt

import cvxopt as opt

from cvxopt import blas, solvers

import pandas as pd

import plotly

import cufflinks

import pandas_datareader as web


# Import the file with the tickers and clean it

tickers = pd.read_excel('Tickers S&P500.xlsx')
tickers.set_index('Ticker', inplace = True)


# Convert credit ratings into categorical data and order them

available_stocks = list(returns.columns)
tickers_available =
tickers[tickers.index.str.match(('|'.join(available_stocks)))].dropna
()

tickers_available.astype(CategoricalDtype(categories = ['AAA',    'AA
+', 'AA', 'AA-', 'A +', 'A', 'A-', 'BBB +', 'BBB', 'BBB-', 'BB +',
'BB', 'BB-', 'B +', 'B', 'B-'], ordered = True ))
tickers_available.to_excel("Available Stocks.xlsx")
```

```
to_plot  =  tickers_available['S&P LT Local Currency Issuer Credit
Rating'].value_counts().reset_index().'S&P LT Local Currency Issuer
Credit Rating').astype(CategoricalDtype(categories = ['AAA', 'AA +',
'AA' , 'AA-', 'A +', 'A', 'A-', 'BBB +',  'BBB', 'BBB-', 'BB +',
'BB', 'BB-', 'B +', 'B', 'B-'], ordered = True)).sort_values('index',
axis = 0)


# Create bar graph

to_plot = to_plot.reset_index().set_index('index')

ax = to_plot.plot.bar(figsize = (12 , 10 ))

plt.box(False)


# Insert bar labels

for i in ax.patches :

  ax.text(i.get_x() + 0.25, i.get_height() + 1.5, \

   str(round((i.get_height ()))), fontsize = 15, color = 'dimgrey', ha = ‘center')
```

First, all the necessary functions are imported, both for this part of the code and the next ones. Then the tickers for the available stocks are uploaded in Python, and the ratings are converted into categorical values. Finally, a graph that shows the disposition of the stocks depending on their rating is generated. The output distribution is the following:

Figure 1: Distribution of stocks based on the issuer credit rating



From the graph, it is possible to see that the division of the stocks can be performed by grouping the tails, to obtain groups composed of about 50 to 70 stocks each. Therefore, stocks with credit rating between AAA and A have been grouped, as well as stocks with rating BB+ to B-.
Once obtained the most suitable groups for the analysis, each bucket is analyzed separately. However, the code can be easily modified from group to group by changing the excel file to upload at the beginning, since the remaining part of the code is the same for each group.
The next step is to download data for the selected stocks (in the code below I take Group 1, AAA to A) as an example.

```
# Download daily stock price data for S&P500 stocks

group_1 = pd.read_excel("Group 1_AAA to A.xlsx")

tick =  list(group_1['Ticker'])

price_data = web.get_data_yahoo(tick,

                    start = '2020-03-23',

                    end = '2020-05-31')['Adj Close’]
```

The output of this piece of code is a data frame indexed by date (daily, from 2020-03-23 to 2020-05-31), with each column representing a single stock price evolution. The period has been appositely chosen: data is collected as from March 23rd (when the Fed announced its commitment to purchase unlimited amounts of US Treasuries and agency MBS, hence halting investors’ sell-offs) for a period of about 2 months. The head of the data frame is represented below:

Table 1

Symbols	AAPL	ACN	ADBE	ADM	ADP	AMP	AMZN	ANTM	APD	ATO	...
Date											
2020-03-23	223.361542	142.522415	307.269989	28.779951	108.433922	81.628555	1902.829956	174.127914	174.597290	79.574158	...
2020-03-24	245.770386	155.575485	310.000000	31.735519	120.254578	98.549156	1940.099976	196.217896	192.073822	87.571106	...
2020-03-25	244.416489	150.586349	305.910004	32.108646	121.875069	102.035706	1885.839966	215.686157	188.843292	89.261452	...
2020-03-26	257.278442	169.947723	322.670013	33.836815	136.260681	109.195938	1955.489990	234.925171	198.910324	95.162781	...
2020-03-27	246.626511	161.556488	305.829987	32.894180	130.613815	101.769783	1900.099976	222.255341	190.838913	96.289673	...

At this point the columns with N/A values are dropped (if present in the data frame) and the daily price matrix is transformed into the return matrix:

```
# Drop na values from price matrix
price_data_no_na = price_data.dropna(axis=1)


# Transform the price matrix into a return matrix
returns = (price_data_no_na/price_data_no_na.shift(1)-1)[1:]
```

To obtain the return matrix, the following formula is applied to each column of the data frame:

$$Return_{i,\ T} = \frac{Stock\ Price_{i,\ T} - Stock\ Price_{i,T-1}}{Stock\ Price_{i,T-1}} = \frac{Stock\ Price_{i,\ T}}{Stock\ Price_{i,T-1}} - 1$$

The result of this step is the following return matrix (for reasons of space, only the head of the data frame is represented below):

Table 2

Symbols	AAPL	ACN	ADBE	ADM	ADP	AMP	AMZN	ANTM	APD	ATO	...
Date											
2020-03-24	0.100325	0.091586	0.008885	0.102695	0.109013	0.207288	0.019587	0.126861	0.100096	0.100497	...
2020-03-25	-0.005509	-0.032069	-0.013194	0.011757	0.013476	0.035379	-0.027968	0.099218	-0.016819	0.019303	...
2020-03-26	0.052623	0.128573	0.054787	0.053823	0.118036	0.070174	0.036933	0.089199	0.053309	0.066113	...
2020-03-27	-0.041402	-0.049375	-0.052190	-0.027858	-0.041442	-0.068008	-0.028325	-0.053931	-0.040578	0.011842	...
2020-03-30	0.028538	0.042608	0.041069	0.069552	0.049018	0.056324	0.033603	0.034177	0.069162	0.055538	...

Once obtained the return matrix, two functions are defined, one to produce x random weights that add up to one (where x is the number of stocks in the group), and one to calculate the standard deviations and returns of the portfolio.

```
# Produce random weights for portfolio stocks, constraining them to be additive to one
def weights(num_stocks):
    k = np.random.rand(num_stocks)
```

```
    return k / sum(k)

# Calculate mean daily return and daily stdev of portfolio i

def returns_and_stdevs(returns):

    mean_returns_vector = np.asmatrix(np.mean(returns, axis = 1))

    weights_vector = np.asmatrix(weights(returns.shape[0]))

    covariance_matrix = np.asmatrix(np.cov(returns))

    return_port = weights_vector * mean_returns_vector.T

    stdev_port = np.sqrt(weights_vector * covariance_matrix * weights_vector.T)

    return return_port, stdev_port
```

The function "returns_and_stdevs" performs the following tasks:

1. Calculate the mean return ("mean_return_vector") for each portfolio and then store the result in a matrix;
2. Create the weight matrix ("weights_vector") by recalling the precedent function ("weights"), with as many weights as the number of stocks in the portfolio;
3. Create the covariance matrix ("covariance_matrix"), by taking the return matrix and applying the covariance function;
4. Calculate the return of the portfolio ("return_port") by doing the dot product between the weight vector and the transpose of the mean daily return vector;
5. Calculate the standard deviation of the portfolio ("stdev_port") by taking the square root of the dot product between the weight vector, the covariance matrix, and the transpose of the weight vector.

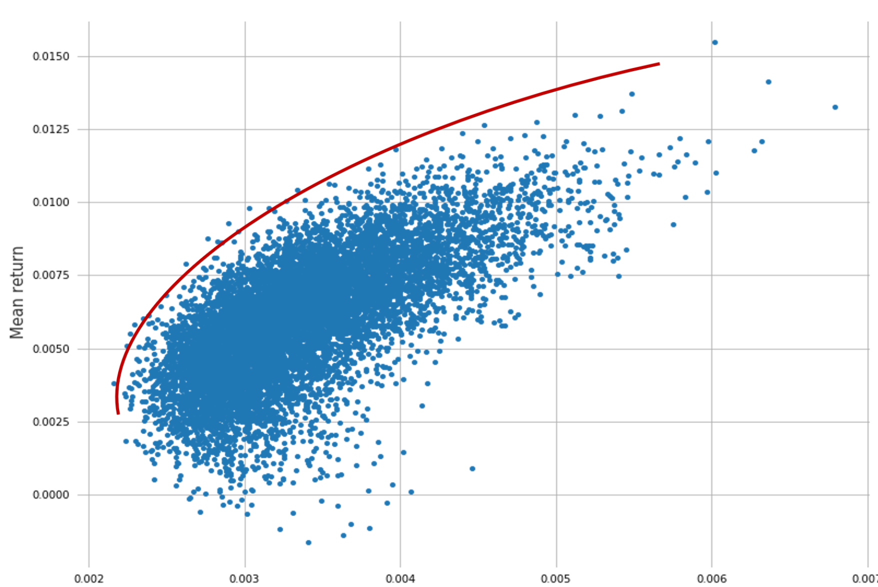
Now that the functions are defined and ready to use, the next step consists of creating 100,000 portfolios, store the results in two variables and then plot the portfolios in a graph.

```
# Create 100,000 portfolios, and record mean daily returns and daily standard deviations for each of them

n  =  100000

means, stds = np.column_stack([returns_and_stdevs(returns) for x in range(n)])
```

```
# Plot portfolios' mean daily returns and daily standard deviations previously created

import  matplotlib.ticker  as  mtick

fig, ax = plt.subplots(figsize = (13, 10))

plt.plot(stds, means, 'o', markersize = 4)

plt.xlabel('Standard Deviation', fontsize = 15, alpha = 0.8)

plt.ylabel('Mean return', fontsize = 15, alpha = 0.8)

plt.title('Return and volatility of {} portfolios'.format(str(n)), fontsize = 20, alpha = 0.8, fontweight =
"bold")

plt.box(False)

plt.grid()

plt.tick_params(top = False, bottom = False, left = False, right = False, labelleft = True, labelbottom =
'on')
```

The output of this portion of the code is a formatted graph with the mean daily return of the portfolio in the y-axis and the standard deviation of the portfolios in the x-axis. To plot a better-defined example, the graph below comprises only 10,000 portfolios (in the real simulation 100,000 portfolios are generated).
Over the graph is represented (in red) an example of an efficient frontier, i.e. the line that connects those portfolios which offer the highest expected return for a specific level of risk. The portfolios below the line are considered to be sub-optimal (i.e. there is at least one portfolio with the same level of risk but with higher mean return).

Figure 2: Portfolio returns vs standard deviations of returns (the red line represents the efficient frontier)



The scope of the analysis is to take the portfolio which maximizes the Sharpe ratio. Therefore, among the 100,000 simulated portfolios, the portfolio with the maximum Sharpe Ratio is going to be selected, and then considered in the comparative analysis, both against the other groups (to compare portfolios with different classes of issuer credit risk) and over time (to see if the effect is due to a structural component of the market, or if Covid-19 changed the physiognomy of the latter).
The Sharpe Ratio is calculated by dividing the difference between the portfolio return and the Risk Free Rate, by the portfolio volatility (the standard deviation of the portfolio calculated for the same time frame as for the portfolio returns):

$$Sharp\ Ratio_T = \frac{Portfolio\ Return_T - Risk\ Free\ Rate}{Portfolio\ Standard\ Deviation_T}$$

Where T is the period taken under consideration (23-03-2020 to 31-05-2020). The risk-free rate is assumed at zero.
Such a measure is particularly useful (even though simplistic when compared to others) to identify the risk-adjusted returns of the different portfolios.
To calculate and store the Sharpe Ratios of the 100,000 portfolios, the following piece of code is used:

```
# Function that implements the same calculation explained above, with
  the exception that it returns Sharpe Ratio and weights instead of
  return and standard deviation

def sharpe_ratios_and_weights(returns):
    mean_returns_vector = np.asmatrix(np.mean(returns, axis=1))
    weights_vector = np.asmatrix(weights(returns.shape[0]))
    covariance_matrix = np.asmatrix(np.cov(returns))

    return_port = weights_vector * mean_returns_vector.T
    stdev_port = np.sqrt(weights_vector * covariance_matrix *
                 weights_vector.T)

    return (float(return_port/stdev_port), weights_vector)

# Simulate 100,000 portfolios and store the Sharpe Ratios and
  weights for each of them

n = 100000

  sharpe_ratios, wghts =
  np.column_stack([sharpe_ratios_and_weights(returns)
    for x in range(n)])
```

```
# Find the maximum sharp ratio
max(sharp_ratios)
```

The interim output is a data frame composed of 100,000 Sharpe Ratios calculated using random weights for the stocks in the group, as well as another data frame containing all the portfolio weights for each portfolio, in the same order as for the Sharpe Ratios (therefore, for example, the Sharpe Ratio number 15 would correspond to the portfolio weight number 15, in their respective data frames).
The final output is instead a number, hence the maximum Sharp Ratios in the data frame. Considering that 100,000 simulations have been performed, the result is very stable, with a minimal and insignificant difference in terms of Sharpe Ratio when re-performing the whole analysis.

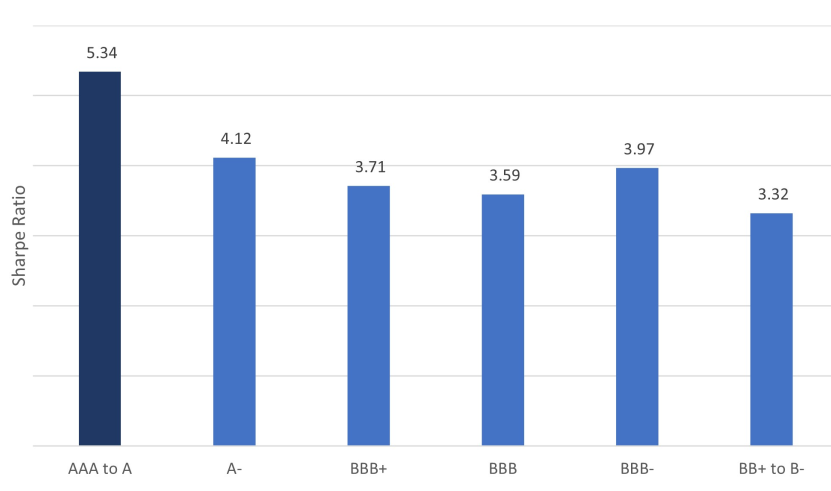
1.2 Comparative analysis by group based on Issuer Credit Rating

The precedent analysis is performed for each group, where groups are defined by classes of issuer credit ratings as follows:

1. Group 1: AAA to A;
2. Group 2: A-;
3. Group 3: BBB+;
4. Group 4: BBB;
5. Group 5: BBB-;
6. Group 6: BB+ to B-.

The analysis is based on the period that goes from March 23rd to May 31st, 2020, taking the stocks of the S&P500 and splitting them as previously stated. The results of the comparative analysis are shown below:

Figure 3: Cross-group comparison of the maximum Sharpe Ratio from the simulation



The result of the cross-group comparative analysis is clear: the maximum risk-adjusted return (Sharpe Ratio) is much higher for the group with the highest Issuer Credit Risk quality (AAA to A), even when compared with its closest group in terms of rating (A-). Moreover, the Sharpe Ratio tends to decline moving down from group 1 to group 6, with the only exception of group 5 (BBB-), which has a better risk-adjusted performance than the one of groups 3 and 4.
This result is totally in line with the behavioral analysis developed in the second section of this paper. Indeed, the idea underlying the analysis is that investors implemented a process of panic-sell, overestimating the risk of default for the majority of the quality stocks in the market. Since this sell-off was mainly dictated by irrational behaviors, after the intervention of the FED on March 23rd, the market calmed down and realized that actually, its reaction was too harsh on some quality stocks.
The main behavioral biases considered in this analysis are summarized below:

1. Loss aversion: considering the long economic expansion and the consequent strong equity returns in the past years, people were in the position of having accumulated a substantial return on their portfolios (with respect to a reference point which normally stands at the initial purchase price of the investment). Therefore, the subjective marginal utility for a unit return was lower than the marginal disutility coming from a unit loss, hence pushing people to divest quickly at the beginning of the market fall, having overestimated the probability of default of their portfolio companies;
2. Myopic Loss Aversion: similar to the standard loss aversion, with the difference that investors are deemed to evaluate their portfolios quite often, hence causing a short-term view on their investments;
3. Herding behavior: especially in times of extreme uncertainty (such as the crisis triggered by the pandemic and the stringent government measures to contain the virus), private investors tend to trade according to what other investors are doing, not following their private information;
4. Framing: agents decide differently depending on how the information is presented to them. During the Covid-19 pandemic, many relevant investors and almost every news channel was very negative on the pandemic, hence influencing people thoughts;
5. Mental accounting: people normally tend to have at least two portfolios, one for low-risk investments (downside protection) and one for high-risk investments (desire to “become rich”). In the sell-off precedent to March 23rd, investments once considered low risk started to decrease substantially in price, with an average price decrease of about 30% (when considering the major indexes). Therefore, people ended up having high-risk stocks in their low-risk portfolios, which in turn induced investors to sell these stocks.

Such behavioral biases resulted in the Sharpe Ratio of the highest Issuer Credit Rating Portfolio to be the highest, considering a strong rebound from March lows, which were dictated more by fear than anything else. Moreover, this also explains the correlation between the Sharpe Ratios and the Issuer Credit Ratings of the stock in the portfolios: the higher the credit quality of the stock, the lower the risk of default, the more probable that investors sold-off the stock just based on fear, being influenced by behavioral biases.

Finally, it is important to highlight that the portfolios have been built randomly, hence not deciding the stocks to insert in each of them, but basing the split solely on the Issuer Credit Rating. Furthermore, the stocks with an Issuer Credit Rating between AAA and A have been aggregated, to create a group with a decent number of stocks inside.

Following the logic coming from the results, it could be possible that this outperformance would be even more pronounced when considering stocks without aggregating different classes of rating.

1.3 Comparative analysis by group based on Issuer Credit Rating over time

We must consider that, even though multiple behavioral biases affected people's behavior during the Coronavirus pandemic, loss aversion and myopic loss aversion are the most measurable ones. Indeed, while other effects are more easily quantifiable through, for example, laboratory experiments and questionnaires, the presence of loss aversion can be here assessed by looking at two particular timeframes: last year returns (2019) and the subsequent stock market drop amidst Covid-19 (from 1st of January 2020 to 23rd of March 2020).

The median is used to measure the performance of the 6 groups in the two aforementioned periods. Indeed, while the mean could lead to biased results by considering outliers, the median is not influenced by these characteristics of the dataset.

The first step to find the median return for each portfolio is to download the stock price data for each group for the relevant period. The below piece of code has been used to extract this data for the period that goes from the beginning of 2020 to March 2020. The same code is then modified to retrieve 2019 stock price data.

```
# Download daily stock price data for the 6 portfolios

group_1 = pd.read_excel("Group 1_AAA to A.xlsx")

tick =  list(group_1['Ticker'])

price_data_1 = web.get_data_yahoo(tick,

                    start = ‘2020-01-01',

                    end = '2020-03-23')['Adj Close']
```

```
group_2 = pd.read_excel("Group 2_A-.xlsx")
tick =  list(group_2['Ticker'])
price_data_2 = web.get_data_yahoo(tick,
                       start = '2020-01-01',

    end = '2020-03-23')['Adj Close']

group_3 = pd.read_excel("Group 3_BBB+.xlsx")
tick =  list(group_3['Ticker'])
price_data_3 = web.get_data_yahoo(tick,
                       start = '2020-01-01',
                       end = '2020-03-23')['Adj Close']
group_4 = pd.read_excel("Group 4_BBB.xlsx")
tick =  list(group_4['Ticker'])
price_data_4 = web.get_data_yahoo(tick,
                       start = '2020-01-01',
                       end = '2020-03-23')['Adj Close']

group_5 = pd.read_excel("Group 5_BBB-.xlsx")
tick =  list(group_5['Ticker'])
price_data_5 = web.get_data_yahoo(tick,
                       start = '2020-01-01',
                       end = '2020-03-23')['Adj Close']

group_6 = pd.read_excel("Group 6_BB+ to B-.xlsx")
tick =  list(group_6['Ticker'])
price_data_6 = web.get_data_yahoo(tick,
                       start = '2020-01-01',
                       end = '2020-03-23')['Adj Close’]
```

Subsequently, the return in the period is calculated by considering only the first (1st of January 2020) and last (23rd of March 2020) price data. Therefore, in the code showed, we keep only stock price data for the 1st of January 2020 and for the 23rd of March 2020. Then, the stock price change (as well as the median of each group) is calculated for the 373 stocks.

```
# For each group, calculate the period return of every stock in the group and then store the median in a variable

first_last_1 = price_data_1.iloc[[0,55]]

returns_1 = first_last_1.pct_change()

median_loss_1 = returns_1.iloc[1].dropna().median()


first_last_2 = price_data_2.iloc[[0,55]]

returns_2 = first_last_2.pct_change()

median_loss_2 = returns_2.iloc[1].dropna().median()


first_last_3 = price_data_3.iloc[[0,55]]

returns_3 = first_last_3.pct_change()

median_loss_3 = returns_3.iloc[1].dropna().median()


first_last_4 = price_data_4.iloc[[0,55]]

returns_4 = first_last_4.pct_change()

median_loss_4 = returns_4.iloc[1].dropna().median()


first_last_5 = price_data_5.iloc[[0,55]]

returns_5 = first_last_5.pct_change()

median_loss_5 = returns_5.iloc[1].dropna().median()

first_last_6 = price_data_6.iloc[[0,55]]

returns_6 = first_last_6.pct_change()

median_loss_6 = returns_6.iloc[1].dropna().median()
```

The same process is repeated by changing the period to 2019 (the entire year). The results are as follow:

Table 3

Group	Median Return 01/01/2019 - 31/12/2019	Median return 01/01/2020 - 22/03/2020
AAA to A	27.0%	-30.8%
A-	30.3%	-35.3%
BBB+	32.4%	-39.3%
BBB	29.3%	-39.7%
BBB-	27.3%	-50.0%
BB+ to B-	25.5%	-45.7%

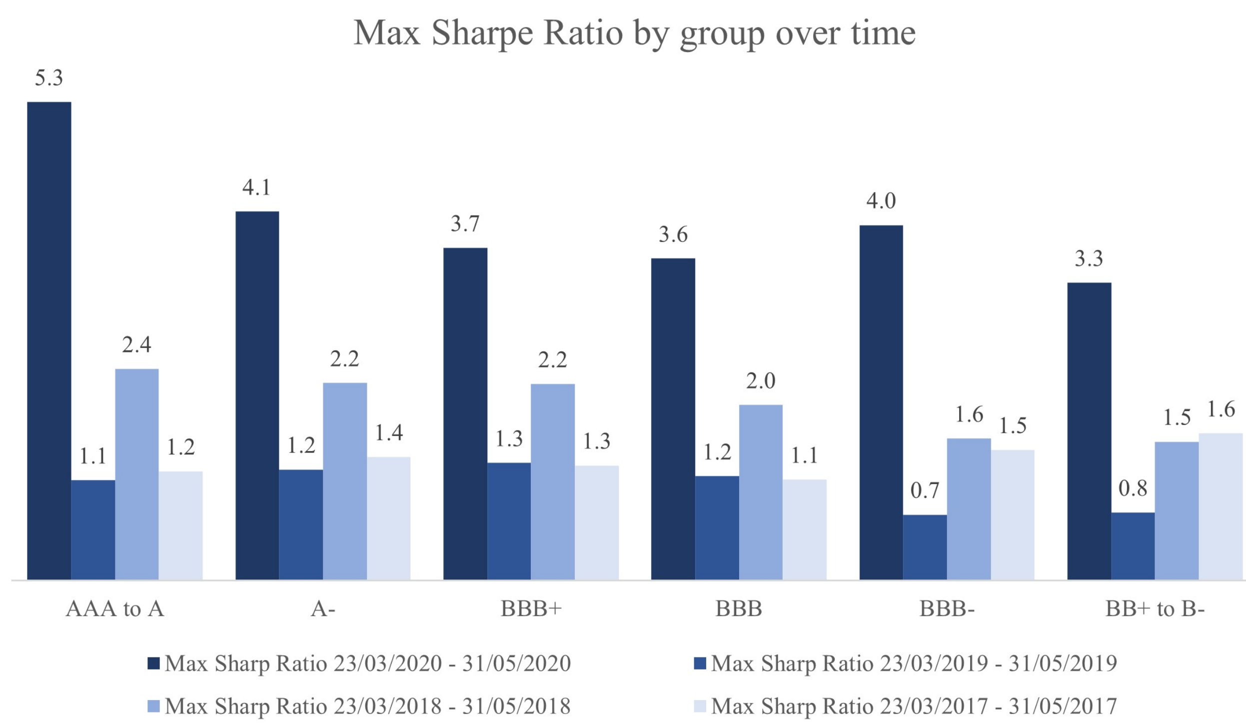
The table above highlights a strong year for stock returns of every credit-risk group in 2019, with a following disastrous first-quarter in 2020. If we consider an average investor who was invested in the market from, for example, the beginning of 2019, loss aversion would have caused this individual to divest during the market crash in the days before the 23rd of March, hence further alimenting the market sell-off. After having accumulated strong returns, the agent was in the right-tale of the utility curve, hence implying a marginal disutility coming from a unit loss to be higher than the marginal utility coming from a unit gain.
After having shown the scenario and the probable reaction of the agent amidst the market sell-off, the Sharpe Ratio results are analyzed over time. To understand whether the results are structural of the market or if, contrarily, they are a direct consequence of a panic-sell dictated by behavioral biases, the same analysis is repeated over time by keeping the same period (to adjust for seasonality) but changing the year of interest. Specifically, 4 periods are considered:

1. From 23/03/2020 to 31/05/2020;
2. From 23/03/2019 to 31/05/2019;
3. From 23/03/2018 to 31/05/2018;
4. From 23/03/2017 to 31/05/2017.

To perform the aforementioned analysis of the groups in various periods, the code previously run is used. The only modification needed is to adapt the start and ending period of the code to the new start and ending period, for each one of the remaining 3 periods (from the 23rd of March to the 31st of May 2019, 2018, and 2017).
This part of the code aims at understanding whether the maximum-sharp-ratio portfolio of the 6 groups displays the same trend over time. The aim of this exercise is to rule out the possibility that the overperformance (adjusted for the risk) of group 1 (best issuer credit rating stocks) is just a structural component of the market: if group 1 strongly outperformed the other groups

also in the same periods of the last years (2019, 2018 and 2017), it would mean that the outperformance in 2020 is not caused by the behavioral biases considered above (mostly by loss aversion), but it is just a recurring characteristic of the group.
The results of the intertemporal analysis are shown below:

Figure 4: Cross-group analysis of max-Sharpe-Ratio portfolio over time

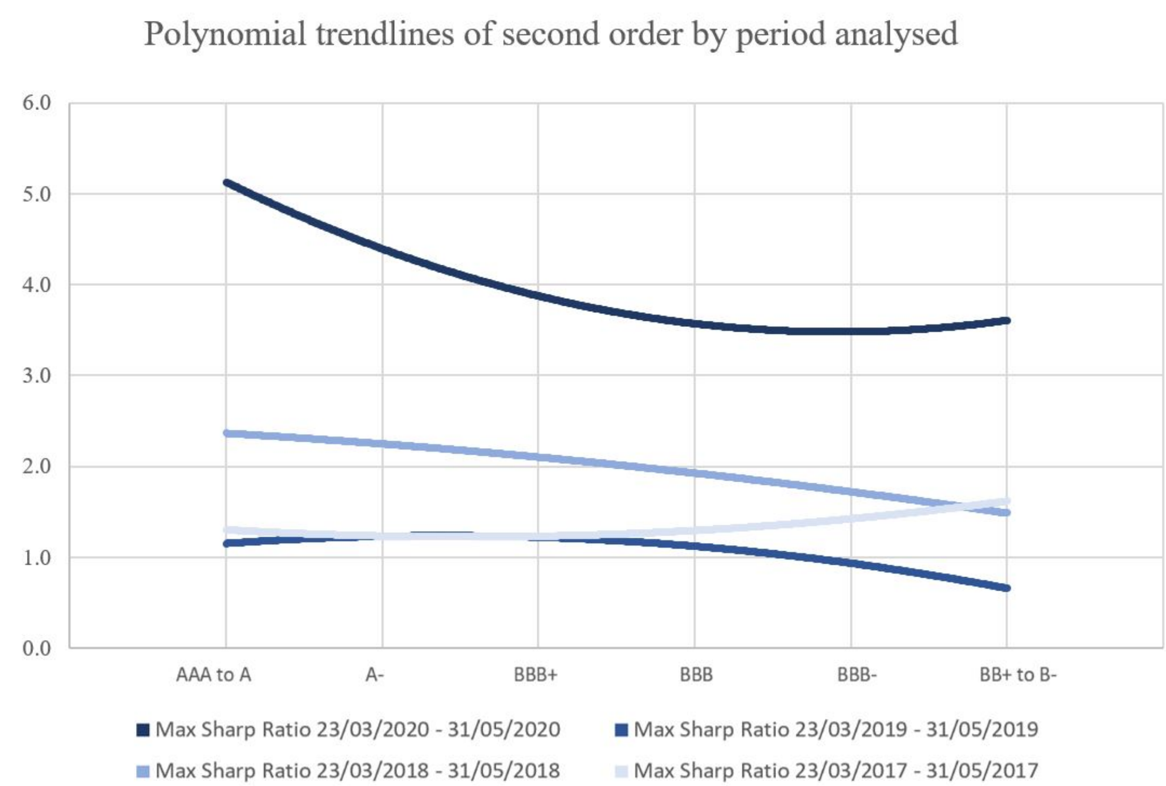


The maximum Sharpe Ratio by group has a different tendency over time:

1. In 2020, group 1 is the best one when compared to the other groups in the same year. Its Sharpe Ratio is extremely higher than the one for group 2 (closest comparable in terms of Sharpe Ratio), precisely 29.7% higher;
2. In 2019, the maximum result is scored by the third group, hence the one composed of stocks having a long-term issuer credit rating of BBB+. At the same time, group 1 ranks 4th in terms of Sharpe Ratio, after Group 2, 3, and 4;
3. In 2018, group 1 ranks 1st in terms of Sharpe Ratio, followed by the other groups in order of issuer credit rating, from the best one to the worst one. However, in this case, the maximum Sharpe Ratio of group 1 is only 7.2% higher than the Sharpe Ratio of its nearest comparable, group 2;
4. In 2017, the maximum result is scored by the last group, which has a Sharpe ratio of about 1.64. In this year, group 1 ranks 5th, almost at the end of the spectrum, with only the 4th group recording a lower Sharpe Ratio.

Therefore, it is reasonable to assess that the outperformance of group 1 is not a structural characteristic of the market, since, in recent years, group 1 has never strongly outperformed the other groups in terms of Sharpe Ratio. Actually, the group normally underperforms the others, with only 2018 being an exception (also in this case, the outperformance is far smaller than the out registered in 2020).
This numerical analysis can be further reinforced by looking at the trend of the maximum Sharpe Ratio by group over time, shown in the graph below:

Figure 5: Intertemporal Second-order Polynomial analysis by group



In 2020, the second-order polynomial trendline is much steeper around group 1. Moreover, its shape is different from the one of the other periods' second-order polynomial lines. Such shape is dictated by the strong result obtained by Group 1, which renders the line convex and decreasing at the same time.
After having considered a numerical and second-order polynomial examination of the intertemporal results of the 6 groups, it is reasonable to conclude that the results obtained from the analysis of the period coinciding with the pandemic are not dictated by a structural overperformance of high-credit-rating stocks. Instead, they are far different from any other period analyzed, suggesting that something else has influenced the results.

1.4 Sectorial analysis of the different groups to identify a possible source of disturbance to the analysis

After having ruled out the possibility of analysis to be biased by a structural component of the market (i.e. that stocks with higher long-term issuer credit rating normally outperform other groups of stocks with lower financial soundness), the sectoral composition of the various groups is taken into consideration in order to understand whether the results have been influenced by the composition of the buckets.

Also in this case, a piece of code has been developed to automatize the task. The first step is to import the packages and functions needed. Then, Financial Modelling Prep is used to download data for the industry of the selected stocks. The data coming from Financial Modelling Prep is encoded in JSON, which is a JavaScript subset used for the structured transmission of data from the server.

Considering that JSON is built-in Python, we can use the JSON library to parse JSON data and receive a well-formatted Pandas data frame as shown below:

Table 4

	AAPL
symbol	AAPL
price	499.23
beta	1.22156
volAvg	39483057
mktCap	2134522860000
lastDiv	3
range	204.22-515.14
changes	-0.81
companyName	Apple Inc
exchange	Nasdaq Global Select
exchangeShortName	NASDAQ
industry	Consumer Electronics

```python
#import packages necessary for the code

import requests
import json
from bs4 import BeautifulSoup as bs
import pandas as pd
import matplotlib.pyplot as plt
```

```
try:
    # For Python 3.0 and later
    from urllib.request import urlopen

except ImportError:
    # Fall back to Python 2's urllib2
    from urllib2 import urlopen

# Set up function to parse JSON data

def get_jsonparsed_data(url):

    response = urlopen(url)
    data = response.read().decode("utf-8")
    return json.loads(data)
```

After having set up the function to parse JSON data, an empty data frame is created, and the previous function is used to download the industry data of each stock.

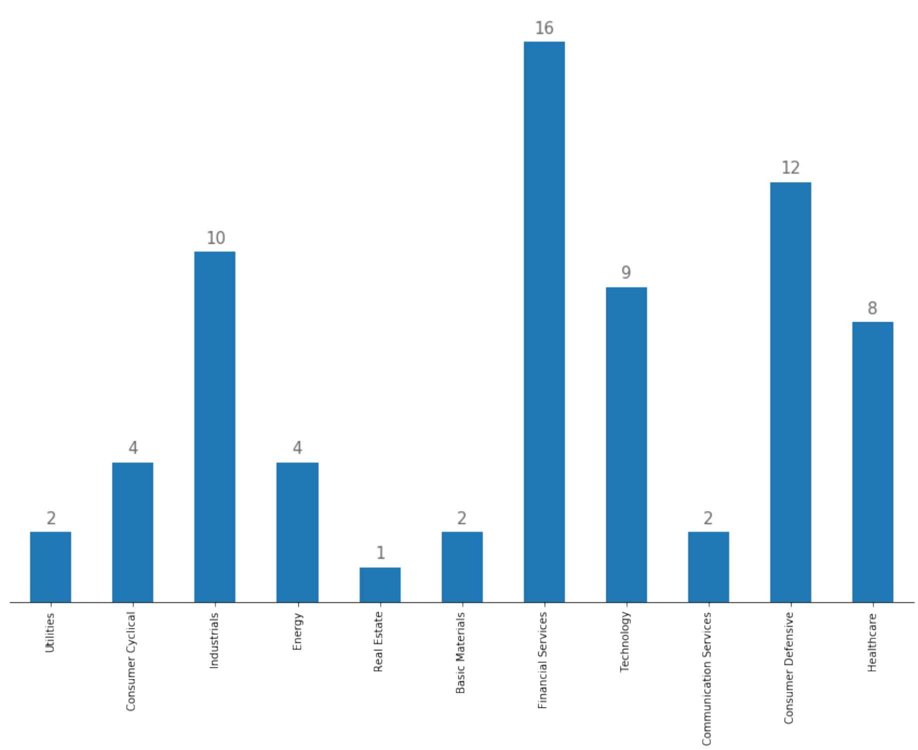
```
# Create an empty list and a list of the stocks included in the group

df1 = pd.DataFrame()
stocks = pd.read_excel('Group 1_AAA to A.xlsx')
stock_list = list(stocks['Ticker'])

# For each ticker in the previous list, create a loop to query the data
and store it in the previously created data frame

for ticker in stock_list:
    url = ("https://financialmodelingprep.com/api/v3/profile/" + ticker
+ "?apikey=7f3999ed50273c3b6f9b889a4b1e28dc")
    ratios = get_jsonparsed_data(url)
    data = pd.Series(ratios[0])
    df1[ticker] = data

df1 = df1.loc['sector']
```

Following the download of the data and the following data cleaning, the data is visualized with the use of a bar graph. The piece of code below creates the graph and formats it:

```
# Create a data frame with the number of stocks by industry for the
group

to_plot = df1.value_counts(sort=False)

# Create the graph and apply some formatting

ax = to_plot.plot.bar(figsize=(12,10))
plt.yticks([])
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.spines['left'].set_visible(False)

# Set individual bar labels using the above list

for i in ax.patches:
    # get_width pulls left or right; get_y pushes up or down

  ax.text(i.get_x()+0.25,i.get_height()+0.25, \
            str(round((i.get_height()))), fontsize=15,
color='dimgrey',ha='center')
```

The output of this part of the code is the following graph, which highlights the number of stocks by industry in the group analyzed:

Figure 6: Number of stock by industry, for group 1

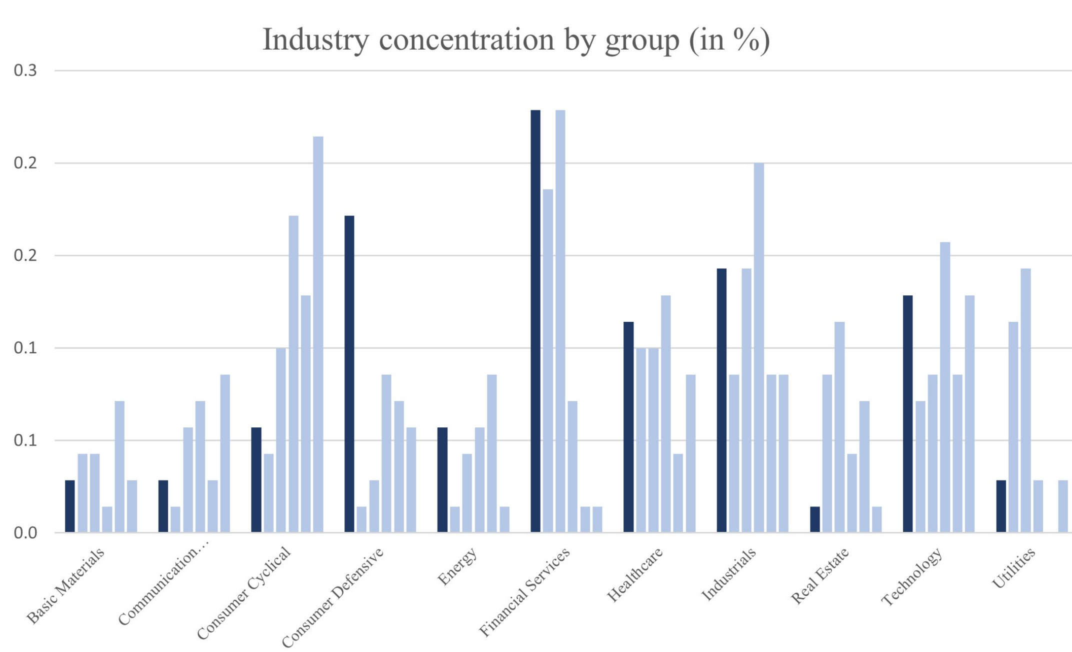


The same process is applied to the other groups. Then, for each of these databases, the number of stocks in each industry bucket is standardized by dividing it by the total number of stocks in its respective group. In fact, considering that each group has a different number of stocks inside, it is better to compare the industrial concentrations by turning the numbers into percentages. To do this, the following code snippet has been used:

```
# Create an empty data frame

dataframe = pd.DataFrame()

# Standardize industry data by turning it into percent form, while
also   storing them into a single dataset

dataframe['Group 1'] = to_plot/sum(to_plot)
dataframe['Group 2'] = data2/sum(to_plot)
dataframe['Group 3'] = data3/sum(to_plot)
dataframe['Group 4'] = data4/sum(to_plot)
dataframe['Group 5'] = data5/sum(to_plot)
dataframe['Group 6'] = data6/sum(to_plot)

# Export the dataset in excel format

dataframe.to_excel('Sectorial allocation by group.xlsx')
```

At this point, the dataset "Sectorial allocation by group" contains the percentage concentration of each industry for every group. Using excel, a graph is created to show the different sectorial concentrations of each group:

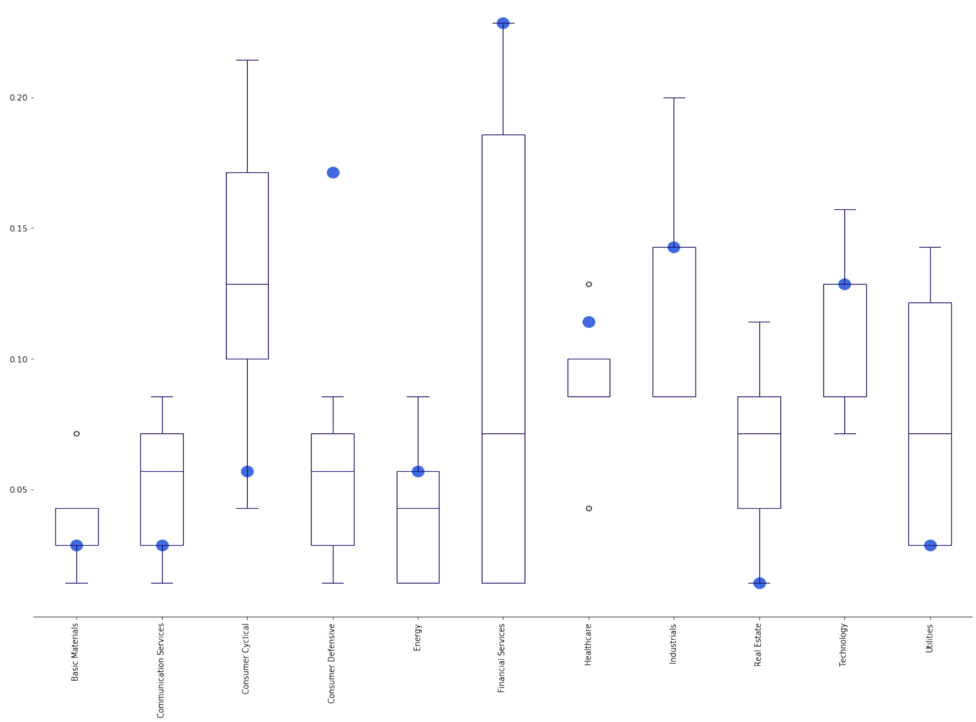
Figure 7: Industry concentration by group, sorted by group number (group 1 in dark blue)



Furthermore, another code snipped is written to create a box plot with an overlapping scatter plot for Group 1:

```
# Read excel file and store data in a data frame

df = pd.read_excel('For box plot.xlsx')
df.set_index('Unnamed: 0', inplace=True)

# Create a box plot and then insert the values for group 1

plt.figure(figsize=(28,10))
ax = df[1:].plot.box(figsize=(28,10), color = 'midnightblue')
group_1 = df.iloc[0]
count = 0
lista = list(df.columns)
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.spines['left'].set_visible(False)
for row in group_1:
    x = count +1
    count += 1
    y = row
    plt.scatter(x, y, color = 'royalblue', s=200)
plt.show()
```

Figure 8: Box plot of industry allocation of group 2 to 6, with an overlapping scatter plot for group 1



From the two graphs (the previous bar chart and the box plot) it is reasonable to assess that, even though the concentration in the different sectors varies between the groups, group 1 does not significantly differ from the others, except for few cases.
Moreover, it is useful to take into account the following table, which ranks the sectors of the S&P500 from the best performer to the worst one, in the period between the 10th of March and the 7th of August:

Table 5

S&P 500 SECTOR	PRICE CHANGE / 10th March to 7th August [1]
Information Technology	30.20%
Consumer Cyclical	30.00%
Basic Materials	22.80%
Communication Services	19.20%
Health Care	11.90%
Industrials	9.20%
Energy	4.90%
Consumer Defensive	4.70%
Financial Services	3.20%
Real Estate	-2.70%
Utilities	-3.30%

From this table and the graphs we can draw the following conclusions on the sectorial allocation of the various groups:

1. Group 1 has the highest number of stocks operating in the consumer discretionary industry and the financial services industry. However, such concentrations could not be the cause of such an outperformance, in terms of Sharpe Ratio, considering that both sectors have been poorly performing in the period;
2. Group 1 has the lowest number of stocks operating in the real estate industry (together with group 6) and a low amount of utility stocks. However, considering that the code implemented aimed to find the portfolio, inside each group, with the highest Sharpe Ratio, it is reasonable to think that, even if the concentration in real estate was higher, real estate stocks would have not been selected by the algorithm.
3. Group 1 outperformance does not depend on having a large number of stocks in the best performing industries, considering that, for information technology, consumer cyclical, and basic materials (the best-performing markets), the group has an average concentration (even lower than the average for consumer cyclical).

Therefore, the results of the sectorial analysis do not discredit the Sharpe Ratio analysis performed at the beginning. Indeed, the outperformance of group 1 in terms of risk-adjusted return does not seem to depend on the group being severely skewed towards those industries which outperformed the market in the period. This result further supports the hypothesis made above regarding the behavioral factors (primarily loss aversion) that caused the market drop and the subsequent returns.

1.5 Final remarks

Considering the results from the intertemporal analysis and the sectorial analysis, the strong outperformance of group 1 in terms of Sharpe Ratio suggests that it is reasonable to believe that the stock market crash was severely influenced by some behavioral factors (with the main one being loss aversion), which caused investors to arbitrarily and materially increase their expectation of probability of default even for financially solid companies with bright prospects. In this case, when investors' fear started to increasingly fade away (as from the 23rd of March), the market started to realize that it was too harsh on some companies which should not have been so penalized. Hence, such a dynamic would explain the strong difference in Sharpe Ratio between group 1 and the other groups.

Finally, even if the intervention of the FED had a massive influence on the returns of the stock market in the period after the 23rd of March, this intervention cannot be the cause of the strong outperformance of Group 1. In fact, this analysis does not focus on a general outperformance of the whole market, but it aims at investigating the outperformance of a specific group (best-in-class issuer credit rating stocks of the S&P500) versus the other ones (based on other levels of issuer credit rating).

2. Conclusion

The research developed in this paper aimed at finding a possible explanation of the stock market collapse in Q1 2020. The analysis starts on the theoretical side, by taking under consideration the development of the Prospect Theory, as well as its utility on the practical side. Following this introduction, a series of heuristics (or behavioral biases) are taken into consideration. Loss aversion, herding behavior, framing, and mental accounting could have been the possible behavioral fallacies that have caused the stock market crash during the coronavirus pandemic. Out of these heuristics, loss aversion is also discussed on the quantitative side, showing to the reader that the situation in Q1 2020 was the perfect scenario for the average investor to develop loss aversion, at least to some extent.

The precedent behavioral biases most probably caused the investor to overestimate the probability of default inherent in some companies or to just adopt a panic-sell behavior, hence alimenting the market decline already in place. If this was the case, the market would have

rapidly corrected this irrationality, by re-stabilizing the price of those stocks whose stock price had previously been erroneously hammered (the most financially solid ones).
To test this hypothesis, an analysis based on long-term issuer credit rating is performed by splitting 373 stocks of the S&P500 into 6 groups. Then, for each group, 100,000 portfolios are created, and the maximum Sharpe Ratio is taken. The results from the benchmarking exercise across the maximum Sharpe Ratio of the 6 groups support the previously stated hypothesis: the best risk-adjusted return of Group 1 (stocks with the highest long-term issuer credit rating) for the, approximately, two months after the market bottom outperforms by far the best risk-adjusted returns of the other groups.
Besides, to tackle some possible issues which could have undermined the results of the study, an intertemporal analysis, as well as a sectorial one, are developed. The outcome of the former is the confirmation that the results obtained are not due to an intrinsic characteristic of the market (i.e. stocks with the highest ratings outperform the other ones). Finally, the second analysis has the objective to demonstrate that sectorial concentrations have not been the cause of the outperformance of group 1. As a matter of fact, the best-credit-rating group does not present a strong concentration in the best performing industries for the relevant period.

Literature Review

D. Bernoulli, 1738, " Specimen theoriae novae de mensura sortis "

W. J. W. Botzen, H. Kunreuther & E. Michel-Kerjan, 2015, "Divergence between individual perceptions and objective indicators of tail risks: Evidence from floodplain residents in New York City"

W. Edwards, 1962, "Utility, subjective probability, their interaction, and variance preferences"

D. Kahneman & A. Tversky, 1979, "Prospect Theory: An Analysis of Decision under Risk"

H. Liu, A. Manzoor, C. Wang, L. Zhang, Z. Manzoor, 2020, "The COVID-19 Outbreak and Affected Countries Stock Markets Response"

H. Fallahgoul, 2020, "Inside the Mind of Investors During the COVID-19 Pandemic: Evidence from the StockTwist Data"

R. Kizys, P. Tzouvanas, M. Donadelli, 2020, "From COVID-19 herd immunity to investors herding in international stock markets: The role of government and regulatory restrictions"

C. N. Chiyachantana & Z. Yang, 2013, "Reference Point Adaptation and Disposition Effect"

M.S. Haigh & J. A. List, 2002, “Do Professional Traders Exhibit Myopic Loss Aversion? An Experimental Analysis”

L. Grune, W. Semmler, 2007, “Asset Pricing with Loss Aversion”

A. Berkelaar, R. Kouwenberg, 2008, “Optimal Portfolio Choices Under Loss Aversion”

V. Zakamouline, 2014, “Portfolio Performance Evaluation with Loss Aversion”

A. Szyszka, 2014, “Behavioral Anatomy of the Financial Crisis”

S. Bikhchandani, S. Sharma, 2000, “Herd Behavior in Financial Markets: A Review”

J. Lakonishok, A. Shleifer & R. W. Vishny, 1992, “The impact of institutional trading on stock prices”

S. Bikhchandani, S. Sharma, 2000, “Herd Behavior in Financial Markets: A Review”

R. Kinzys, P. Tzouvanas & M. Donadelli, 2020, “From COVID-19 Herd Immunity to Investor Herding in International Stock Markets: The role of government and regulatory restrictions”

Müller & Lyer, 1889, “Psychophysische Untersuchungen. Über die Abhängigkeit der relativen Unterschiedsempfindlichkeit von Intensität und Extension des Reizes”

A. Kumar & S. S. Lim, 2008, “How Do Decision Frames Influence the Stock Investment Choices of Individual Investors?”

N. Barberis & M. Huang, 2005, “The Loss Aversion / Narrow Framing Approach to the Equity Premium Puzzle”

H. Shefrin & M. Statman, 2000, “Behavioral Portfolio Theory”

List of figures