UNIVERSITY OF LAUSANNE
FACULTY OF BUSINESS AND ECONOMICS

Assignment 1

Group members: *Antoine Tissot, Florian Perusset, Florian Vogt, Liam Svoboda, Sebastien Gorgoni*

Course: *Empirical Methods in Finance* – Professor: *Dr. Florian Ielpo*
Due date: *March 30th, 2021*

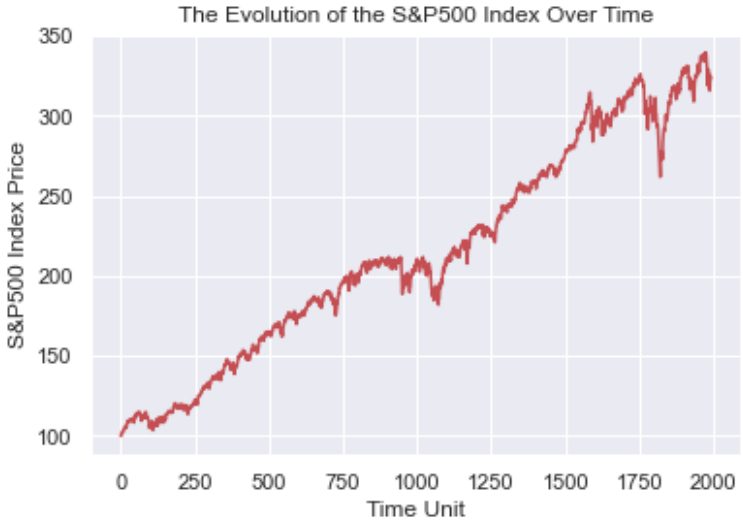
Contents

1. Introduction

This report aims to analyze the stock performance of the S&P500 index's constituents. We will first briefly explain what the S&P500 stock market index is, and how its constituents enter and leave the index. Then, we will analyze the skewness and kurtosis of the individual companies to get a better idea of their distribution characteristics. Next, we will conduct various hypothesis tests. starting with the Jarque & Bera test and the Kolmogorov & Smirnov test to check for normality of distribution followed by the Ljung Box test to check for potential auto-correlations. Finally, we will run a regression on each individual stock, with the return on the market portfolio as the independent variable. Finally, we will briefly explain the significance of these empirical results in the context of investment strategies.

2. Description of the Data Set

Our data set includes the prices of 452 out of the 500 constituents of the S&P500 index, as well as the value of the index itself, over an unspecified period of time, indexed at 100 in time 0. For the purpose of pure data description, the following graph indicates that the S&P500 has witnessed a strong upward trend over time, although it encountered various medium-to-high downward pressure on some occasions. More on the technical analysis will be presented in the forthcoming sections. The Python source code for the following graph is located in the "Initial Data Processing" part.



The data processing and analysis was conducted with Python, using various libraries. The codes will be highlighted throughout this report, as guidance for the reader, and a full version will also be included at the end of the report.

3. Initial Data Processing

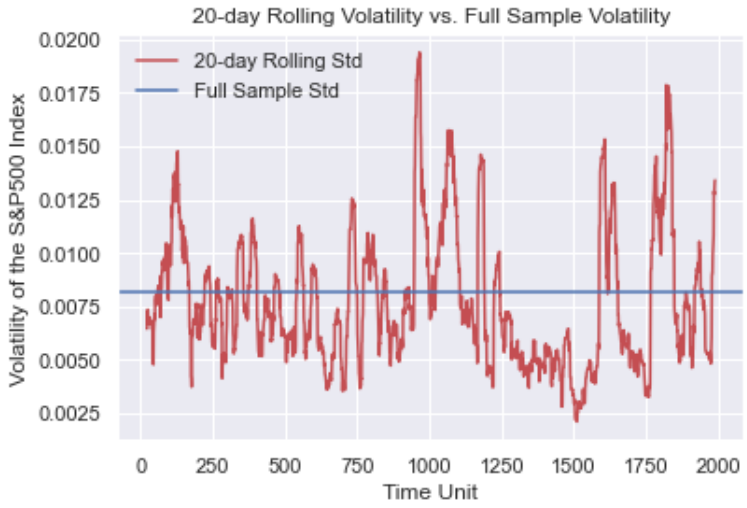
Before beginning the analysis, we will need to define some essential elements. Let P_t be the stock price at time t. Then the one-period simple return from t-1 to t is:

$$r_t = \frac{P_t - P_{t-1}}{P_{t-1}} \tag{1}$$

Let μ be the full sample average of the returns for the a given stock. Then the rolling 20-day (non annualized) variance of the returns is:

$$\hat{\sigma}_t^2 = \frac{1}{20}\sum_{i=0}^{20}(r_{t-i} - \hat{\mu})^2 \tag{2}$$

First and foremost, it is crucial to understand the rationale of using a 20-day rolling volatility rather than the full period standard deviation. Indeed, we believe that the 20-day rolling volatility would be more appropriate for technical analysis, as it allows us to account for the fact that standard deviation is not constant throughout the time period. The full period volatility would be a vast oversimplification of the behaviour of returns. As we notice on the following graph, the volatility demonstrates sharp variability over time, indicating that a constant standard deviation would not be optimal for investment purposes.



Then, we would define the return adjusted to volatility (also called curly return or adjusted return) as:

$$\tilde{r}_t = \frac{r_t}{\hat{\sigma}_t} \tag{3}$$

Such a variable is essential for any investment criterion, as it accounts for volatility. Investors using a mean-variance analysis might be sensitive to such a factor, as it would induce a trade-off between returns and volatility.
For this report, we used the following libraries on Python 3.8:

- Pandas (Version: 1.2.2)
- Scipy (Version: 1.6.1)
- Seaborn (Version: 0.11.1)
- Statsmodels (Version: 0.12.2)
- Matplotlib (Version: 3.3.4)

The Python source code for this initial data processing is as follows:

```
import pandas as pd
import scipy.stats as sc
import seaborn as sns
import statsmodels.api as sm
import matplotlib.pyplot as plt

sns.set_theme(style="darkgrid")

# Load data
sp500Price = pd.read_excel("Data_assignment.xlsx") # daily prices

#Generate Simple Return for All S&P500 Constituents and SP500 Index
 ↪  itself
sp500Ret = sp500Price / sp500Price.shift(1) - 1 # daily simple returns
 ↪  for each stock
sp500Ret = sp500Ret.dropna(how='any'); # get rid of NaN
marketRet = sp500Ret['Index'] # store the market returns
marketRet.dropna(how='any', inplace=True)
sp500Ret.drop('Index', inplace=True, axis=1) # get rid of the column
 ↪  'Index' for the first questions

m_range = range(0,21)

#20-Day Rolling Variance of All S&P500 Constituents
sp500RollingVariance = sp500Ret.copy()
sp500RollingVariance = abs(sp500RollingVariance*0)

for i in m_range:
    sp500RollingVariance += (sp500Ret.shift(i) - sp500Ret.mean())**2

sp500RollingVariance = (sp500RollingVariance/20)

#Returns Adjusted to Volatility (Curly Return)
sp500RetCurly = sp500Ret / (sp500RollingVariance**.5) # DataFrame
 ↪  containing the r_t / sigma_t

#20-Day Rolling Variance of S&P500 Index
marketRollingVariance = marketRet.copy()
marketRollingVariance = abs(marketRollingVariance*0)

for i in m_range:
    marketRollingVariance += (marketRet.shift(i) - marketRet.mean())**2

marketRollingVariance = (marketRollingVariance/20)

# clean data from NaN
sp500RetCurly.dropna(how='any',inplace=True)
```

```
sp500RollingVariance.dropna(how='any',inplace=True)
marketRollingVariance.dropna(how='any',inplace=True)

#Graphs for SP500 Index Price
plt.plot(marketRollingVariance**0.5, '-r', label='20-day Rolling Std')
plt.axhline(y=marketRet.std(), color='b', linestyle='-',label='Full
 ↪  Sample Std')
plt.xlabel('Time Unit')
plt.ylabel('Volatility of the S&P500 Index')
plt.title('20-day Rolling Volatility vs. Full Sample Volatility')
plt.legend(loc='upper left', frameon=False)

#Graph for Cumulative Returns of SP500 index (Same Interpretation as the
 ↪  Next Plot)
marketCumulRet = (marketRet + 1).cumprod() -1
plt.plot(marketCumulRet, '-b')
plt.xlabel('Time Unit')
plt.ylabel('S&P500 Index Cumulative Return')
plt.title('S&P500 Index Cumulative Return Over Time')

#Graphs for SP500 Price
plt.plot(sp500Price['Index'], '-r')
plt.xlabel('Time Unit')
plt.ylabel('S&P500 Index Price')
plt.title('The Evolution of the S&P500 Index Over Time')
```

4. Question 1: The S&P500 and its Constituents

The SP500 is a stock market index composed of the 500 largest US publicly-traded companies. It uses the capitalization weighting method[1]:

- Inclusion Criteria:
  1. Should be a US company
  2. Market Cap. $\geq$ \$ 8.2 Billion
  3. High liquidity of its shares
  4. Part of its outstanding shares available to public ≥ 50
  5. Last quarter's earnings must be positive
  6. Sum of last 4 quarter's earnings must be positive

  Hence, companies which stop conforming to these criteria become excluded from the index. There is however some leeway in order to avoid turbulence caused by high turnover within the index

[1] `https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/sp-500-index/`, accessed 18.03.21

- Components are weighted based on free-float market capitalization

$$W_i = \frac{P_i * Q_i}{\sum Q} \tag{4}$$

Where:
W_i is the weight of company i in the S&P500 index.
P_i is the stock price of company i.
Q_i is the number of free-floating shares of company i.

5. Question 2: Skewness

Before introducing the concept of skewness and kurtosis, we need to define the k^{th} central moments of a random variable X, which determine the dispersion of its distribution. Let m_1 be the sample average. Then the k^{th} central moments is defined as :

$$\mu_k = \left[(X - m_1)^k\right] = \int_{-\infty}^{\infty} (x - m_1)^k f_X(x)\, dx \text{ for } k = 1, 2, ... \tag{5}$$

When analysing the returns of an asset, especially its distribution, it is essential to determine its skewness. Since its return can be perceived as a random variable, determined by a certain distribution, the skewness allows us to ascertain the asymmetry of this distribution, in comparison to a symmetric distribution (e.g. Normal distribution). We distinguish two different outcomes[2]:

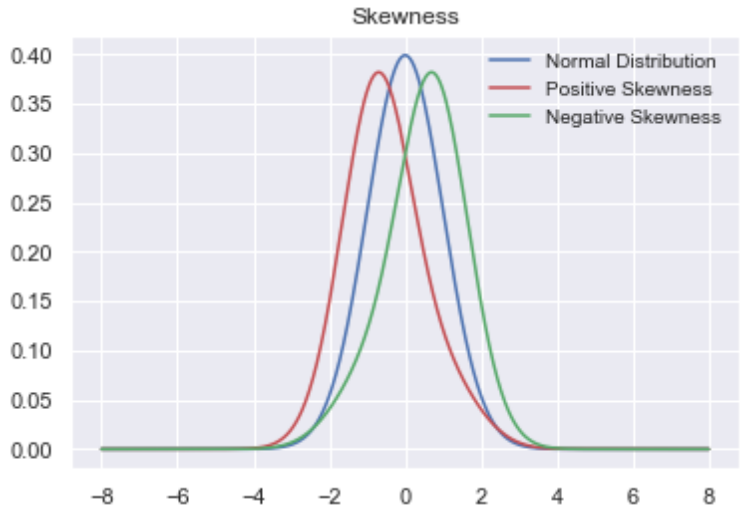
1. **Positive Skewness ($S > 0$):** When the distribution is deviated to the left, and its longest tail is located to the right (i.e. one can expect moderate negative returns frequently and occasionally some substantial positive outcomes).

2. **Negative Skewness ($S < 0$):** When the distribution is deviated to the right, and its tail is located to the left (i.e. one can expect moderate positive returns frequently and occasionally some substantial negative outcomes).

Notice then that when the skewness of a distribution is equal to zero, it implies that the random variable is symmetrically distributed (e.g Normal distribution). For better visualization, we simulated a positive and negative skewness alongside a normal distribution, by adapting the Python source code of "Statistics, Data Mining, and Machine Learning in Astronomy" (2013)[3].

[2] `https://corporatefinanceinstitute.com/resources/knowledge/other/skewness/` Accessed:21.03.21

[3] `https://www.astroml.org/book_figures/chapter3/fig_kurtosis_skew.html`, accessed 18.03.21



The code for this simulation is as follows:

```
 #Adapted from:
 ↪  https://www.astroml.org/book_figures/chapter3/fig_kurtosis_skew.html

import numpy as np
from scipy import stats
from matplotlib import pyplot as plt
import seaborn as sns


sns.set_theme(style="darkgrid")

fig = plt.figure(figsize=(6, 7.25))

# Skewness

x = np.linspace(-8, 8, 1000)
N = stats.norm(0, 1)

plt.plot(x, N.pdf(x), 'b',
              label='Normal Distribution')

plt.plot(-x, 0.5 * N.pdf(x) * (2 + x + 0.5 * (x * x - 1)),
              'r', label='Positive Skewness')

plt.plot(x, 0.5 * N.pdf(x) * (2 + x + 0.5 * (x * x - 1)),
              'g', label='Negative Skewness')
plt.legend(loc='upper right', frameon=False, fontsize=10)
plt.title('Skewness')
```

Let's denote X a random variable (e.g. stock returns), μ the sample mean and σ the sample standard-deviation. Then, the skewness will be defined by the 3^{rd} central

moment of X:

$$S[X] = E\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] \tag{6}$$

For sample of n values, where m_2 is the sample second central moment (i.e. the biased variance) and m_3 is the sample third central moment, the empirical skewness is defined as:

$$\hat{S} = \frac{m_3}{m_2^{\frac{3}{2}}} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^3}{\left[\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2\right]^{\frac{3}{2}}} \tag{7}$$

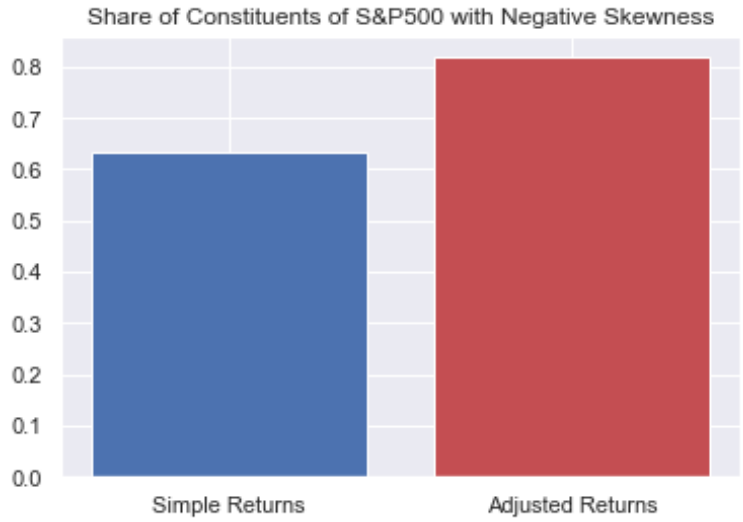
Using our database, we implemented the following code to determine the skewness of the simple returns and the returns adjusted to risk[4]:

```
#Skewness of simple returns and returns adjusted to volatility
sp500RetSkew = sp500Ret.skew().to_frame(name='skewness')
sp500RetCurlySkew = sp500RetCurly.skew().to_frame(name='skewness')

#Proportion of stocks with a negative skewness
sp500RetNegSkew = sp500RetSkew['skewness'].loc[sp500RetSkew['skewness']
 ↪  < 0].count() / sp500RetSkew.shape[0]
sp500RetTildaNegSkew =
 ↪  sp500RetCurlySkew['skewness'].loc[sp500RetCurlySkew['skewness'] <
 ↪  0].count() / sp500RetSkew.shape[0]

# share of constituents of S&P500 with negative skewness for r_t
print('share of constituents of S&P500 with negative skewness for r_t:
 ↪  ', sp500RetNegSkew,'\n')
# share of constituents of S&P500 with negative skewness for r_t_curly
print('share of constituents of S&P500 with negative skewness for
 ↪  r_t_curly: ', sp500RetTildaNegSkew,'\n')

label=['Simple Returns', 'Adjusted Returns']

#Plot the propositon obtained as a bar chart
sp500NegSkew = [sp500RetNegSkew, sp500RetTildaNegSkew]

colors = ['b', 'r']

plt.bar(label, sp500NegSkew, color=colors)
plt.title('Share of Constituents of S&P500 with Negative Skewness')

plt.figure(figsize=(20,10))

#Historgram of the number of observation per excess kurtosis
plt.subplot(121)
```

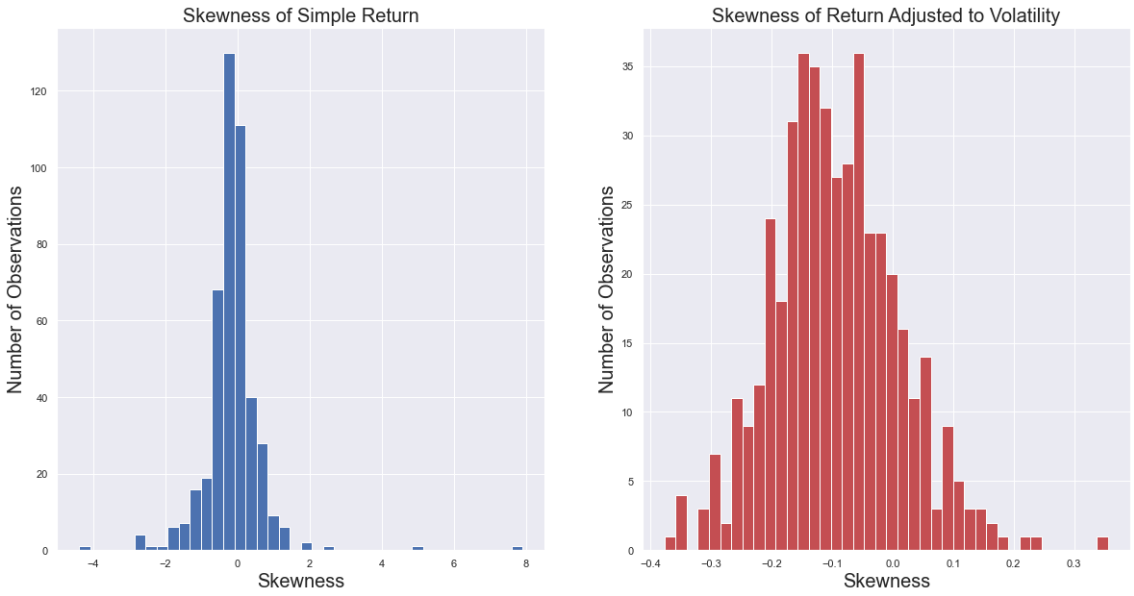
[4]Although we could have computed the skewness manually, we decided to use our libraries to compute it for the sake of preciseness.

```
plt.hist(sp500RetSkew, color='b', bins=40)
plt.title('Skewness of Simple Return', fontsize=20)
plt.ylabel('Number of Observations', fontsize=20)
plt.xlabel('Skewness', fontsize=20)

plt.subplot(122)
plt.hist(sp500RetCurlySkew, color='r', bins=40)
plt.title('Skewness of Return Adjusted to Volatility', fontsize=20)
plt.ylabel('Number of Observations', fontsize=20)
plt.xlabel('Skewness', fontsize=20)
```

Consequently, the proportion of constituents of the S&P500 with simple returns exhibiting a negative skewness is 63.5%. The same measure for adjusted returns is 81.9%, as indicated in the following histogram:



For both simple returns and returns adjusted to volatility, more than 50% of stocks are negatively skewed, meaning that these assets yield moderate positive returns frequently and occasionally some substantial negative outcomes. However, when examining the returns adjusted to the 20-day rolling volatility, the proportion of negatively skewed stocks increases. This result would suggest that the returns adjusted to volatility would have a more skewed distribution. Such difference would have an impact on the optimal portfolio allocation, relative to the risk aversion of the investor. When looking into more detail, we notice that there is more variety in the skewness of simple returns than for the return adjusted for volatility.



6. Question 3: Kurtosis

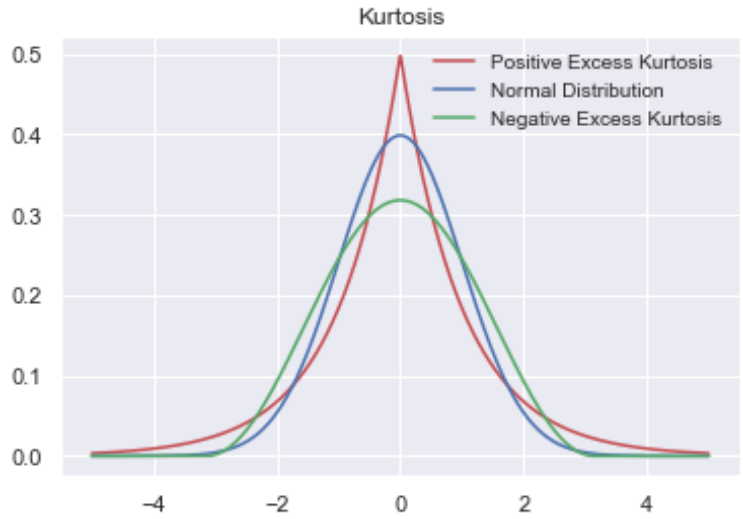
Another component of the distribution of an asset, which is sometimes overlooked is the kurtosis. More precisely, excess kurtosis allows us to determine if extreme values are more or less likely to occur than the base case of a random variable following a normal distribution. In other terms, kurtosis gives us information on the size of the tails of the distribution. We can again distinguish two major cases [5]:

1. **Positive Excess Kurtosis ($K > 3$):** When the the tails of the distribution are fatter. This means that there are larger outliers. In investment terms, one would experience extreme returns on either side of the mean.

2. **Negative Excess Kurtosis ($K < 3$):** When the the tails of the distribution are flat. This means that there are small outliers. In investment terms, extreme returns would be mitigated both on the positive and negative side.

Notice then that when the kurtosis of a distribution is equal or close to three, it implies that the distribution has similar tails that of a normal distribution. Similarly to the skewness, we also simulated distributions containing positive/negative excess kurtosis alongside the normal distribution:

[5] `https://corporatefinanceinstitute.com/resources/knowledge/other/kurtosis/` Accessed: 21.03.21



To be more specific, we simulated a platykurtic distribution (i.e. cosine distribution) and a leptokurtic distribution (i.e. Laplace distribution). This simulation has been done as follows:

```
 # Kurtosis

x = np.linspace(-5, 5, 1000)
plt.plot(x, stats.laplace(0, 1).pdf(x), 'r',
              label='Positive Excess Kurtosis')
plt.plot(x, stats.norm(0, 1).pdf(x), 'b',
              label='Normal Distribution')
plt.plot(x, stats.cosine(0, 1).pdf(x), 'g',
              label='Negative Excess Kurtosis')
plt.legend(loc='upper right', frameon=False, fontsize=10)
plt.title('Kurtosis')
```

Let's denote X a random variable (e.g. stock returns), μ the sample mean and σ the sample standard-deviation. As the kurtosis will be based on the 4^{th} central moment of X, its statistical definition will be:

$$K[X] = E\left[\left(\frac{X-\mu}{\sigma}\right)^4\right] \tag{8}$$

For sample of n values, where m_2 is the sample second central moment (i.e. the biased variance) and m_4 is the sample fourth central moment , the empirical kurtosis is defined as:

$$\hat{K} = \frac{m_4}{m_2^2} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^4}{\left[\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2\right]^2} \tag{9}$$

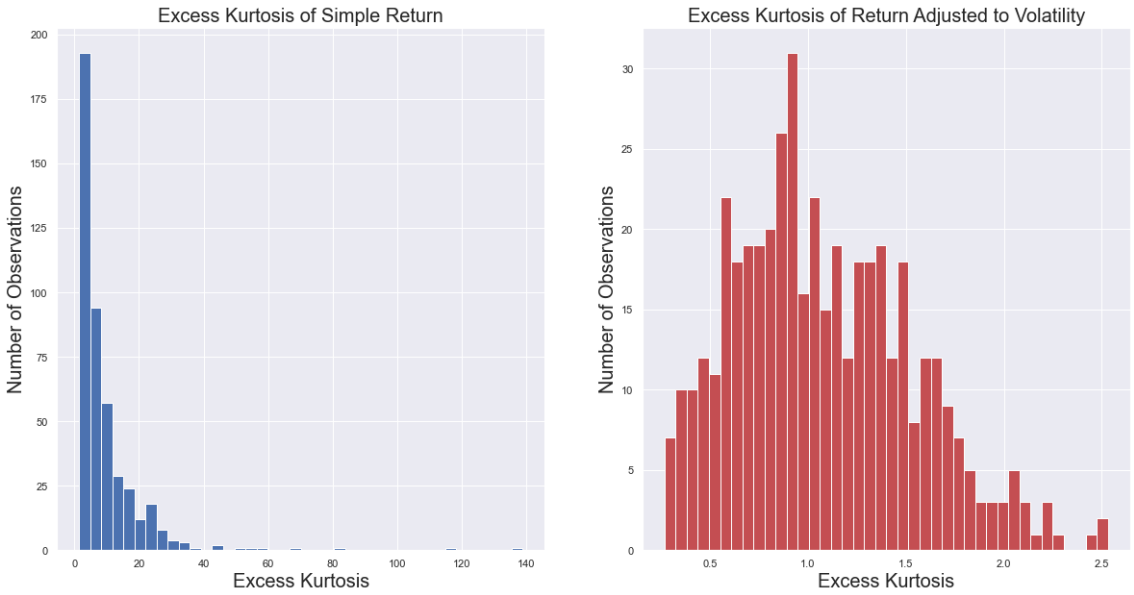
In practice, we generally deal with excess kurtosis, meaning additional kurtosis compared to a normal distribution which has a kurtosis of 3. The following code was used to determine the excess kurtosis of returns and returns adjusted for volatility[6]:

[6]Although we could have computed the excess kurtosis manually, we decided to use our libraries to compute it for the sake of preciseness.

```
 #Kurtosis of simple returns and returns adjusted to volatility
sp500RetExcessKurt = sp500Ret.kurtosis().to_frame(name='excess
 ↪  kurtosis')
sp500RetCurlyExcessKurt = sp500RetCurly.kurtosis().to_frame(name='excess
 ↪  kurtosis')

#Proportion of stocks with a kurtosis above 3 (note that the command
 ↪  .kurtosis() compute directly the excess kurtosis)
sp500RetPosExcessKurt = sp500RetExcessKurt['excess
 ↪  kurtosis'].loc[sp500RetExcessKurt['excess kurtosis'] > 0].count() /
 ↪  sp500RetExcessKurt.shape[0]
sp500RetTildaPosExcessKurt = sp500RetCurlyExcessKurt['excess
 ↪  kurtosis'].loc[sp500RetCurlyExcessKurt['excess kurtosis'] >
 ↪  0].count() / sp500RetCurlyExcessKurt.shape[0]

# proportion of constituents of S&P500 exhibiting r_t with kurtosis > 3
print('proportion of constituents of S&P500 exhibiting r_t with kurtosis
 ↪  > 3: ', sp500RetExcessKurt['excess
 ↪  kurtosis'].loc[sp500RetExcessKurt['excess kurtosis'] > 0].count() /
 ↪  sp500RetExcessKurt.shape[0],'\n')
# proportion of constituents of S&P500 exhibiting r_t_curly with
 ↪  kurtosis > 3
print('proportion of constituents of S&P500 exhibiting r_t_curly with
 ↪  kurtosis > 3: ', sp500RetCurlyExcessKurt['excess
 ↪  kurtosis'].loc[sp500RetCurlyExcessKurt['excess kurtosis'] >
 ↪  0].count() / sp500RetCurlyExcessKurt.shape[0],'\n')

plt.figure(figsize=(20,10))

#Historgram of the number of observation per excess kurtosis
plt.subplot(121)
plt.hist(sp500RetExcessKurt, color='b', bins=40)
plt.title('Excess Kurtosis of Simple Return', fontsize=20)
plt.ylabel('Number of Observations', fontsize=20)
plt.xlabel('Excess Kurtosis', fontsize=20)

plt.subplot(122)
plt.hist(sp500RetCurlyExcessKurt, color='r', bins=40)
plt.title('Excess Kurtosis of Return Adjusted to Volatility',
 ↪  fontsize=20)
plt.ylabel('Number of Observations', fontsize=20)
plt.xlabel('Excess Kurtosis', fontsize=20)
```

As a result, we arrive at the conclusion that 100% of companies in our sample exhibit a kurtosis above 3 for both returns and returns adjusted for volatility. Taking a step further, we plotted the distribution of excess kurtosis for the measures of interest which shows a wide variety of values between the stocks in our sample. Standardizing

returns by volatility gives us much smaller values for kurtosis.



7. Question 4: Jarque & Bera Test

The Jarque-Bera (JB) Test is a statistical test that aims to test whether a variable is Gaussian or not. The test statistic is computed in the following manner:

$$JB = T\left[\frac{\hat{S}^2}{6} + \frac{(\hat{K}-3)^3}{24}\right] \tag{10}$$

Where:
$\hat{S}$ is the empirical skewness
$\hat{K}$ is the empirical kurtosis.

Hence, the JB test makes use of these 2 moments of the variable that we covered in questions 2 and 3, the skewness and the kurtosis. Again, if the variable is normally distributed, its skewness is equal to 0 and its kurtosis equal to 3.
Thus, under the null hypothesis of the Jarque-Bera test, it is assumed that the variable is Gaussian. Hence, the skewness and kurtosis should take the aforementioned values. We also know that under the null hypothesis, the test statistic defined by equation (10) follows a Chi-Squared distribution with 2 degree of freedom. The 95^{th} percentile of the Chi-Squared distribution with 2 degrees of freedom is approximately equal to 6, hence we will reject the null hypothesis if JB is higher than 6.
The code that we use is the following:

```
# Jarque Bera test for normality

# for r_t
T = sp500Ret.shape[0]
jbRet = T * ((sp500Ret.skew()**2)/6 + (sp500Ret.kurtosis()**2)/24)
jbRet = jbRet.to_frame(name='JB test statistic')
jbRet['p-val'] = 1.0 - sc.chi2.cdf(jbRet['JB test statistic'], 2)
```

```
# proportion of constituents of S&P500 whose r_t are not normal at 5%
 ↪  significance level
print('proportion of constituents of S&P500 whose r_t are not normal at
 ↪  5% significance level: ', jbRet['p-val'].loc[jbRet['p-val'] <
 ↪  0.05].count() / jbRet.shape[0],'\n')


# for r_t_curly
T = sp500RetCurly.shape[0]
jbRetCurly = T * ((sp500RetCurly.skew()**2)/6 +
 ↪  (sp500RetCurly.kurtosis()**2)/24)
jbRetCurly = jbRetCurly.to_frame(name='JB test statistic')
jbRetCurly['p-val'] = 1.0 - sc.chi2.cdf(jbRetCurly['JB test statistic'],
 ↪  2)

# proportion of constituents of S&P500 whose r_t_curly are not normal at
 ↪  5% significance level
print('proportion of constituents of S&P500 whose r_t_curly are not
 ↪  normal at 5% significance level: ',
 ↪  jbRetCurly['p-val'].loc[jbRetCurly['p-val'] < 0.05].count() /
 ↪  jbRetCurly.shape[0],'\n')
```

Using this test for both the raw return r_t and return adjusted for volatility $\tilde{r}_t$ of each stock, we rejected the null hypothesis in every case, meaning that with a significance level of 5%, we rejected the hypothesis that the stock returns follow a Gaussian distribution for each stock we have in our sample.
Hence, we can say that the proportion of the stocks for which the null hypothesis can be accepted is 0%, with only 5%, of being wrong for each of the stocks in our sample. We obtained exactly the same result for the returns adjusted for volatility than for raw returns.

8. Question 5: Kolmogorov & Smirnov Test

In this question, we use another statistical test to see whether returns r_t and $\tilde{r}_t$ follow a Gaussian distribution. The test statistic of a Kolmogorov & Smirnov test is calculated as follows:

$$KS = \max_{t=1,..,T} |G_T(x) - F^*(x;\theta)| \tag{11}$$

This test statistic is based on the discrepancy between the theoretical cumulative distribution function (CDF) $F^*(x;\theta)$ that prevails under the null hypothesis (in this particular case, a Normal distribution) and the empirical cumulative distribution function $G_T(x)$, that is the sample of the sample CDF. The Python code that we have used for this part is the following:

```
# Kolmogorov and Smirnov Test
def ks_test(column):
    return pd.Series(sc.kstest(column, 'norm'), index=['KS
     ↪  statistic','p-val'])
```

```
# for r_t
ksRet = sp500Ret.apply(ks_test)
ksRet = ksRet.transpose()
# proportion of S&P500 constituents that reject Gaussian hypothesis for
 ↪  r_t at 5% significance
print('proportion of S&P500 constituents that reject Gaussian hypothesis
 ↪  for r_t at 5% significance level: ',
 ↪  ksRet['p-val'].loc[ksRet['p-val'] < 0.05].count() /
 ↪  ksRet.shape[0],'\n')

# for r_t_curly
ksRetCurly = sp500RetCurly.apply(ks_test)
ksRetCurly = ksRetCurly.transpose()
# proportion of S&P500 constituents that reject Gaussian hypothesis for
 ↪  r_t_curly at 5% significance
print('proportion of S&P500 constituents that reject Gaussian hypothesis
 ↪  for r_t_curly at 5% significance level: ',
 ↪  ksRetCurly['p-val'].loc[ksRetCurly['p-val'] < 0.05].count() /
 ↪  ksRetCurly.shape[0],'\n')
```

The results are similar than with the Jarque-Bera test.
For raw returns r_t, the null hypothesis of Gaussianity is always rejected as with the Jarque-Bera test. In the case of $\tilde{r}_t$, the null hypothesis is not rejected in only 0.66% of cases, compared to 0% with the Jarque-Bera test.
Hence, we can say that both tests provide similar results and leads to the rejection of the null hypothesis of Gaussianity in the vast majority of cases, considering both raw and volatility-adjusted returns.

9. Question 6: Ljung-Box Test

Another feature to take into account is autocorrelation. Applied to returns, autocorrelation allows one to measure the linear relationship between the return in one period and its past values up to a certain time lag. We are therefore interested in whether stock returns of the S&P500 constituents are autocorrelated or not. To conduct such a test, we run a Ljung–Box test which allows us to test whether any of the autocorrelation coefficients up to a certain lag is different from zero or not:

$$H_0 : \rho_1 = \rho_2 = ... = \rho_p = 0 \tag{12}$$

The Ljung-Box Q'-statistic is defined as:

$$Q'_p = T(T+2) \sum_{j=1}^{p} \frac{1}{T-j} \hat{\rho_j}^2 \tag{13}$$

The python code we used in order to test for the presence of autocorrelation up to lag 10 ($p = 10$) for returns and adjusted returns, is as follow:

```
# Ljung-Box test
def lb_test(column, p=10):
```

```
    T = column.shape[0]
    lbstats = 0
    for lag in range(1, p+1):
        lbstats += (column.autocorr(lag=lag) ** 2) / (T - lag)
    lbstats *= T * (T + 2)
    pvalue = 1 - sc.chi2.cdf(lbstats, p)
    return pd.Series(data=[lbstats, pvalue], index=['LB statistic',
     ↪  'p-val'])


# for r_t
lbRet = sp500Ret.apply(lb_test)
lbRet = lbRet.transpose()
# proportion of S&P500 constituents exhibiting autocorrelated returns
 ↪  r_t at 5% significance
print('proportion of S&P500 constituents exhibiting autocorrelated
 ↪  returns r_t at 5% significance: ', lbRet['p-val'].loc[lbRet['p-val']
 ↪  < 0.05].count() / lbRet.shape[0],'\n')

# for r_t_curly
lbRetCurly = sp500RetCurly.apply(lb_test)
lbRetCurly = lbRetCurly.transpose()
# proportion of S&P500 constituents exhibiting autocorrelated returns
 ↪  r_t_curly at 5% significance
print('proportion of S&P500 constituents exhibiting autocorrelated
 ↪  returns r_t_curly at 5% significance: ',
 ↪  lbRetCurly['p-val'].loc[lbRetCurly['p-val'] < 0.05].count() /
 ↪  lbRetCurly.shape[0],'\n')
```

We found that roughly 20% (19.91%) of the constituents of our sample exhibit autocorrelated returns while only 8.41% of them show autocorrelation when it comes to returns adjusted for volatility at a significance level of 5%. These results therefore suggest that the majority of S&P500 stocks returns do not exhibit any momentum effect in their dynamics meaning that we wouldn't be able to forecast any future returns using actual and passed data. When adjusting returns for risk, it comes to the fact that even fewer S&P500 constituents show returns autocorrelated in time. This is an interesting result as it is in line with the efficient market hypothesis assuming, inter alia, that stock prices are not predictable based on past prices.

It is now interesting to have a look at the dynamic of absolute returns in terms of autocorrelation. Indeed, this will allow us to test for some possible time dependency in volatility as absolute returns can be used as a time-varying measure of volatility of returns. To do so, we ran the following code:

```
# for abs_r_t
sp500AbsRet = abs(sp500Ret)
lbAbsRet = sp500AbsRet.apply(lb_test)
lbAbsRet = lbAbsRet.transpose()
# proportion of S&P500 constituents exhibiting autocorrelated absolute
 ↪  returns r_t at 5% significance
```

```
print('proportion of S&P500 constituents exhibiting autocorrelated
 ↪  absolute returns r_t at 5% significance: ',
 ↪  lbAbsRet['p-val'].loc[lbAbsRet['p-val'] < 0.05].count() /
 ↪  lbAbsRet.shape[0],'\n')

# for abs_r_t_curly
sp500AbsRetCurly = abs(sp500RetCurly)
lbAbsRetCurly = sp500AbsRetCurly.apply(lb_test)
lbAbsRetCurly = lbAbsRetCurly.transpose()
# proportion of S&P500 constituents exhibiting autocorrelated returns
 ↪  r_t_curly at 5% significance
print('proportion of S&P500 constituents exhibiting autocorrelated
 ↪  returns r_t_curly at 5% significance: ',
 ↪  lbAbsRetCurly['p-val'].loc[lbAbsRetCurly['p-val'] < 0.05].count() /
 ↪  lbAbsRetCurly.shape[0],'\n')
```

We therefore found that mostly all (98.89%) S&P500 constituents exhibit autocorrelated absolute returns and 78.98% of them show autocorrelation in their absolute risk adjusted returns processes at a significance level of 5% . The important increase in both figures suggests that most return processes exhibit some time dependency in volatility meaning that returns time series show some well-known behavior when talking about returns : volatility clustering. This result suggests that there exist some persistence in the magnitude of returns or in other word : important returns tend to be followed by important returns and vice-versa. This is a major result when it comes to financial returns prediction as it introduces a new challenge : forecasting volatility...

10. Question 7: Regression

Before interpreting the results, let us have a closer look at the model to have an insight of what we are trying to explain:

$$r_t = \alpha + \beta r_t^{MP} + \sigma\epsilon_t \text{ with } \epsilon \sim N(0,1) \; iid \tag{14}$$

As can be seen, the estimated model aims to quantify the relation between returns of a particular stock and the index. That is, it tries to identify whether stocks composing the S&P500 are rather aggressive or defensive. Recall that an aggressive stock experiences exacerbated returns relative to the market, both positive or negative. This is the case of stocks whose market beta is superior to 1 (in absolute terms). In the opposite case, a defensive stock is a stock whose returns tend to move less relative to the market, which is described by a beta lower than 1 (in absolute terms). Note that if we consider the same model as above, but with excess returns instead of simple returns only, one obtains the well-known CAPM model. This model allows us to predict excess return of a particular stock given excess return of the market.

To perform an OLS analysis on the entire S&P500, we first define a function taking as arguments a series of returns of a company and another series for the market returns (with a default value for the Index present in the database), and regressing the particular stock's returns on the market returns. This choice is made for convenience since this function can then be applied to each column of the DataFrame's returns. The function is defined as follows:

```
def ols_market_returns(column, market = marketRet):
    """Returns the results of a linear regression of @column on @market,
     ↪  by default marketRet"""
    x = sm.add_constant(market)
    y = column.values
    model = sm.OLS(y,x)
    results = model.fit()
    return pd.Series(data = [results.params[1], results.tvalues[1]],
     ↪  index=['beta estimates','t-stat'])
```

Then, the function is applied to each column and the results are stored in a new Pandas DataFrame for the betas to be retrieved more easily:

```
sp500RetOls = sp500Ret.apply(ols_market_returns)
sp500RetOls = sp500RetOls.transpose()
```

Once the entire process is achieved, the "real" analysis part can begin. To compute the proportion of stocks with a beta higher than one, an aggregate function is called:

```
print('proportion of stocks with beta_market > 1: ', sp500RetOls['beta
 ↪  estimates'].loc[sp500RetOls['beta estimates'] > 1].count() /
 ↪  sp500RetOls.shape[0],'\n')
```

The results are the following: roughly one half (51.11%) of S&P500 stocks exhibit aggressive returns over the sample period. Furthermore, the highest beta is around 2 and the lowest is around 0.29. Let us try to find some explanations to these extreme variations.

One explanation could be the hype on certain stocks, or the presence of S&P500 stocks in various ETFs that tend to be shorted when the market is declining and longed when the market is rising. Indeed, if those firms are part of many different financial products highly related to market activity, their returns would be mechanically affected by trends.

Another explanation could be the nature of the firm's output. As a matter of fact, some elements exhibit an incompressible demand such as food, beverage, or utilities. However, some sectors perform well when market is rising and future prospects are positive, and extremely poorly when the market is correcting and people expect a recession. As an example, one is more willing to buy the newest iPhone when future job/financial prospects are positive, while water supply will reasonably remain constant throughout the year. The iPhone case would thus have a beta larger than one, while the water supply company will have a beta smaller than 1.

Also note that the model has some limitations, as it contains only one explanatory variable. Such a model certainly presents many omitted components affecting both the market and the individual stock returns, leading to an omitted variable bias, and thus biased beta estimates. As a result, some betas could be higher or lower in the "true" model. Such variables could be the size, the industry, the quality of the earnings and the book-to-market ratio of the firms. Indeed, it is important to notice that S&P500 is

criticized for having a too important concentration of technology firms (around 27.6% of the index[7]). A solution to that problem would be to adapt the model to make those factors appear, such as the Fama & French three or five factor models.

11. Question 8: Interpretation

Although it was certainly a learning experience to manipulate and analyse our sample data in and of itself, the main objective of this exercise and indeed the course in general is to apply econometrics to complement our financial skill set and allow us to make better informed investment decisions. Although we explained the relevance of most of our computations with respect to investing along the way, we will now detail what each question in this assignment allowed us to conclude.

- Question 1:
  The S&P500 is a favorite among industry professionals and academics alike and is widely used as a benchmark. It is therefore crucial to understand exactly how it functions.

- Question 2:
  Although the many financial models and metrics, or at least the more simplistic ones focus purely on the first two moments of the distribution of financial products, its mean and its volatility, there is value to be derived from looking at higher moments. This question tackles the third moment, skewness which as a reminder, quantifies the asymmetry of the distribution. Through this exercise, we were able to ascertain that a majority of stocks in our sample are negatively skewed, meaning that we can expect large infrequent losses and frequent positive gains. This is a non desirable characteristic of returns, since risk averse investors fear extreme losses.

- Question 3:
  In this question, we continue to seek more information through higher moments by determining the kurtosis which measures the flatness of the tails of the distribution. All companies in our sample exhibited positive excess kurtosis meaning that there were a greater number of extreme returns than would be expected from a normal distribution. This is not good news for the investment community as most investors are inherently risk averse which means they would rather have a relatively low sure outcome than a higher expected return.

- Question 4:
  This task and the one that follows revolve around whether returns follow a Gaussian distribution. This is a very important question since many financial models and econometric models in general have a normality assumption as a critical requirement for their final result because of it's simplicity. For example, the basic Black-Scholes model uses a normality assumption for the returns of the underlying asset. Using the Jarque-Bera statistic which is derived through what we computed in questions 2 and 3, we reject the null hypothesis that returns follow a normal distribution. This was to be expected given the proportions of highly skewed and kurtosed stock returns in our sample.

[7] `https://www.thebalance.com/what-is-the-sector-weighting-of-the-s-and-p-500-4579847`, accessed: 14.03.2021

- Question 5:
  Here, we pretty much confirm our results from the previous question using another method to compare the empirical distribution of our data to the normal distribution. Debunking the normality assumption can be quite unfortunate because it brings more complexity to pricing models. In addition although a normal distribution is totally characterized by its two first moments, that is not the case for all types of distributions. Hence, this reinforces the importance of taking the time to calculate the skewness and kurtosis of returns and should make us think twice when applying overly-simplistic models.

- Question 6:
  Question 6 goes in a different direction however and aims to reveal any potential inter-temporal relations, referred to as autocorrelations in returns. The benefit of such relations is that they could potentially allow us to predict future returns which would obviously be a tremendous advantage. However, after performing the Ljung-Box test, this dream quickly comes crashing down as we find that only a slim minority of returns display such characteristics on simple returns and even fewer on volatility-adjusted returns. This supports with the Efficient Market Hypothesis which is an important point of debate. It might be interesting however to perhaps build a momentum strategy with the few companies that do have significant autocorrelation coefficients in hopes that these coefficients persist in the future.

- Question 7:
  In the final exercise, use a close relative of the CAPM but we omit the risk-free return. More specifically, we are interested in the βs. This measure allows us to see how correlated a given stock return is to the return of the index and used very often thanks to its simplicity to understand and calculate. It is especially interesting to know the β of a stock in times of economic downturn or economic growth as it can allow us to build strategies around our expectations of the economy making it very topical in these times of economic instability.

12. Conclusion

In this report, we firstly focus on the analysis of the third and fourth central moment of the S&P500 stock returns distribution. Based on our data set, we have concluded that stock returns mostly exhibit negative skewness and excess kurtosis.

Then, using the previous results, we have conducted two well-known hypothesis tests, the Jarque-Bera test and the Kolmogorov-Smirnov test, to test Gaussianity of the stock returns. The results that we have obtained with both tests show that stock returns are not Gaussian. As we have already discussed, the fact that returns are not Gaussian has a lot of implications in portfolio theory and asset pricing, as most of the classical models in finance rely on the hypothesis that stock returns are Gaussian (i.e. Black-Scholes model, Modern Portfolio Theory of Markowitz, etc.).

After that, we have also tested the auto-correlations of stock returns. We have shown that the majority of stock returns are not auto-correlated, and have therefore an unpredictable pattern. This has important implications in terms of trading strategies and it is also an evidence in favour of the Efficient Market Hypothesis.

Finally, we ran a regression to compute a sort of market beta of each stock in our sample, and we have seen that roughly 51% of the stocks have a beta larger than 1. This means that these stocks are considered to be rather aggressive, meaning that their returns exhibit larger fluctuations than the S&P500 (market portfolio).