Fredrik Aas Hellesvik

Volatility Forecasting and Portfolio Optimization

Master's thesis spring 2021
Oslo Business School
Oslo Metropolitan University
Master's program in business administration

Sammendrag

Denne oppgaven undersøker sammenhengen mellom volatilitetsprognoser og en porteføljes ytelse. Målet er å bruke stiliserte fakta om avkastning på finansielle eiendeler for å forbedre nøyaktigheten av volatilitetsprognoser og se om bedre prognoser forbedrer porteføljevalg – og ytelse. GARCH-modeller brukes til å forutsi volatilitet over en rullende periode på 1.008 handelsdager (4 år). Volatilitetsprognosene brukes til å konstruere Markowitz mean-variance optimale porteføljer ved å maksimere porteføljens Sharpe Ratio. Vi finner at evnen til å forutsi volatilitet er knyttet til porteføljens ytelse. Strategiene som forutsier volatiliteten med høyeste nøyaktighet, presterer bedre enn de andre porteføljene når det gjelder kumulativ avkastning og standardavvik for avkastningen.

Abstract

This thesis investigates the relationship between volatility forecasting and portfolio performance. The aim is to use stylized facts about financial asset returns to improve the accuracy of volatility forecasts and see if better forecasts can improve portfolio selection – and performance. GARCH type models are used in order to forecast volatility over a rolling period of 1,008 trading days (4 years). The volatility forecasts are used to construct Markowitz mean-variance optimal portfolios by maximizing the Sharpe Ratio of the portfolios. We find that the ability to forecast volatility is linked with portfolio performance. The strategies that are able to forecast the volatilities with highest accuracy outperforms the other strategies in terms of cumulative returns and standard deviation of the returns.

Preface

This study represents my last semester of the master's program in Business and Administration with specialization in Finance at the Oslo Metropolitan University. The study will be conducted as my master thesis.

I would like to thank my supervisor Johann Reindl for encouraging me to write this thesis and always being available for academic support.

The process of writing the thesis has been extremely demanding but at the same time I have acquired knowledge that I am sure will benefit me in the future. A lot of the calculations has been done in programming language, mainly Python and R. For someone that has no earlier programing experience, this has been the most demanding part of the thesis.

Oslo Metropolitan University, June 10, 2021

Fredrik Aas Hellesvik

Table of Contents

Table of Equations

Table of Figures

Table of Tables

1. Introduction

In this thesis we will look at the relationship between volatility forecasting and portfolio performance. To make better volatility forecasts we will investigate some stylized facts about financial asset returns in order to make better assumptions about the distribution and behavior of these returns. This is done in section 2.
Section 3 present the methodology of our study. We look at how we can construct optimal portfolios, to do this we use the Markowitz mean-variance criterion by maximizing the Sharpe ratio of the portfolios. We present different volatility models, mainly models in the GARCH family, and how we will carry out the analysis.
Section 4 gives an overview of the data, this consists of the price on the S&P 500 Index, the prices on ten different stocks, and the 3-Month Treasury Bill rate.
In Section 5 we look at the results of our study and evaluate the performance of the different strategies in terms of cumulative returns, standard deviation of returns, and their ability to forecast volatility. The results in section 5 are concluded in section 6.

1.1 Research Question

- "Can we use stylized facts about financial asset returns to make better volatility forecasts?"
- "Is there a relationship between volatility forecasting and portfolio performance?"

Our aim is to investigate if we can improve portfolio selection by making better volatility forecasts. It is not to find the "best" model for volatility forecasting.

2. Stylized Facts About Financial Asset Returns

2.1 Definition of Returns

If the price of an asset at time t is p_t, then the continuously compounded return is:

$$r_t = \ln\left(\frac{p_t}{p_{t-1}}\right) = \ln(p_t) - \ln(p_{t-1}) \tag{1}$$

2.2 Stylized Fact 1

Non-Normality:

One of the most frequently made assumption in finance is that returns of financial assets are normally distributed. Almost every financial textbook makes this assumption because the normal distribution can be described by its first two moments, the mean and the standard deviation. Thus, the normal distribution is a lot easier to work with compared to other more complex distributions.

Nevertheless, the returns of financial assets usually do not follow a normal distribution. Figure 1 shows the distribution of the daily returns on the SP 500 Index between 02/01/1990 – 31/12/2019 and the corresponding normal distribution. As we can see from the figure the returns on the SP 500 are more centered around the mean and have fatter tails (leptokurtic) compared to a normal distribution with the same mean and the same standard deviation. The distribution also skews a little to the left. Returns of financial assets usually follow this type of distribution, which resembles more of a student t than a normal distribution (Franke, Härdle and Hafner 2008, 227).

Figure 2 shows the Normal Q-Q Plot of the SP 500 Index returns. The figure compares the theoretical quantile of the normal distribution to the sample quantile of the returns. If the returns where normally distributed the circles would be aligned with the blue line, as we can see this is not the case.

Another way to check for normality in financial asset returns is to calculate the statistical moments of the returns. In order to check for normality, we need the third and fourth moments of the distribution, namely the skewness and the kurtosis. The skew tells us whether the distribution is leaning towards higher or lower values, and the kurtosis tells us something about the shape of the distribution.

Figure 1

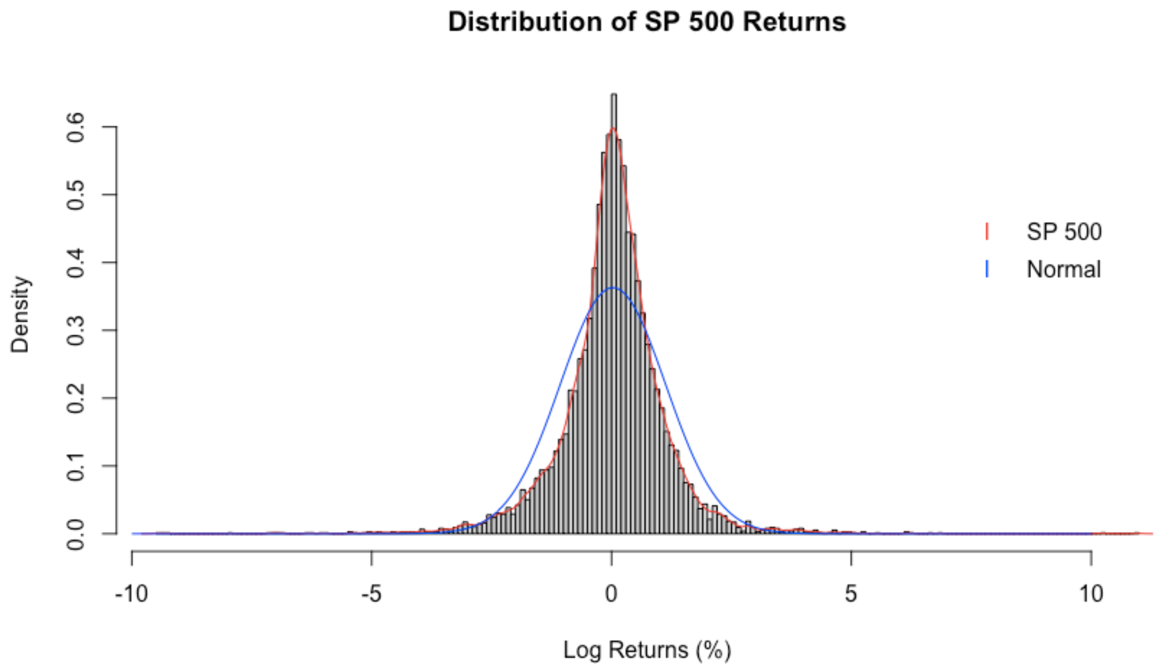
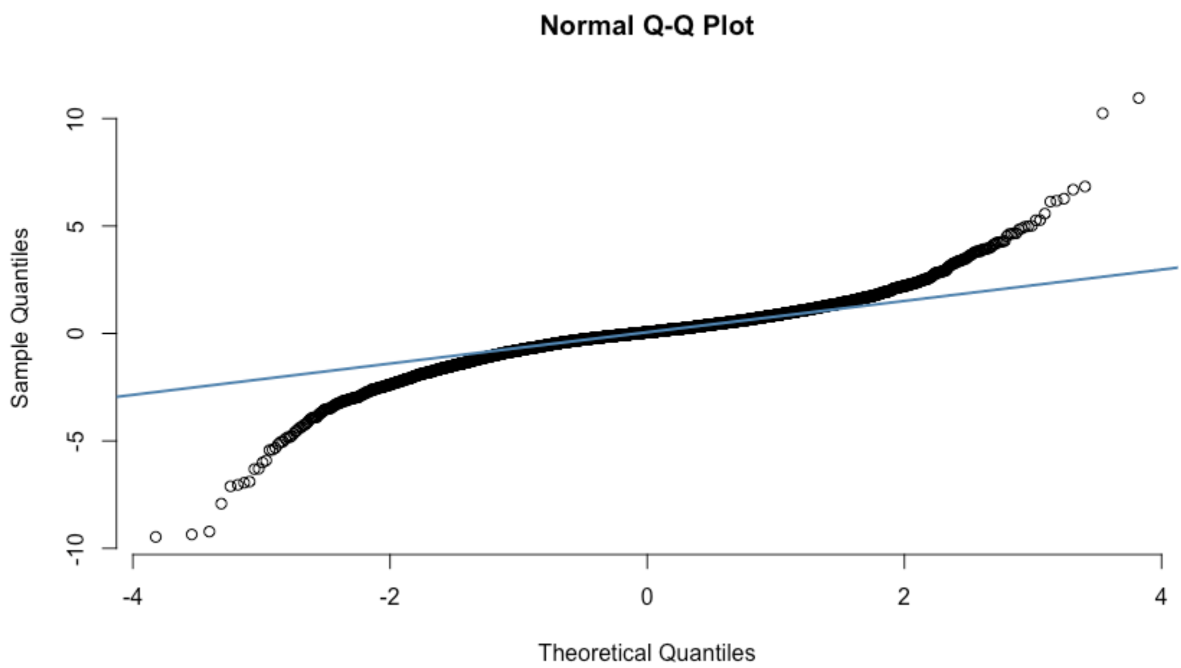


Figure 2



To test for normality, we use the Jarque-Bera Test:

$$JB = T\left(\frac{skew^2}{6} + \frac{(kurt-3)^2}{24}\right) \tag{2}$$

The test statistics, JB, is χ^2 distributed with two degrees of freedom.
The null hypothesis is that the sample is derived from a normal distribution (Brooks 2014, 209-210).

Table 1 shows that the null hypothesis is rejected for all the stocks we will use to construct our portfolios.

Table 1: Test of Normality

Ticker	Mean	Std.	Skew.	Kurt.	JB Stat.	p - value
MMM	0.0581%	1.1416%	-0.5010	4.4137	1288.0	0.0000
VZ	0.0455%	1.0257%	-0.1117	1.4384	133.2	0.0000
NEE	0.0688%	1.0384%	-0.2250	2.8246	514.4	0.0000
PM	0.0462%	1.0822%	0.0095	3.8290	921.8	0.0000
GE	0.0499%	1.3044%	0.2634	5.1431	1680.6	0.0000
AMT	0.0537%	1.3159%	-0.4368	3.7954	953.7	0.0000
TGT	0.0231%	1.2767%	-0.3378	5.0089	1606.2	0.0000
AMD	0.0231%	3.6087%	0.8035	15.2854	14852.7	0.0000
MO	0.0861%	0.9920%	-0.4577	2.3126	389.0	0.0000
CCI	0.0524%	1.2566%	-0.0809	3.0631	591.6	0.0000

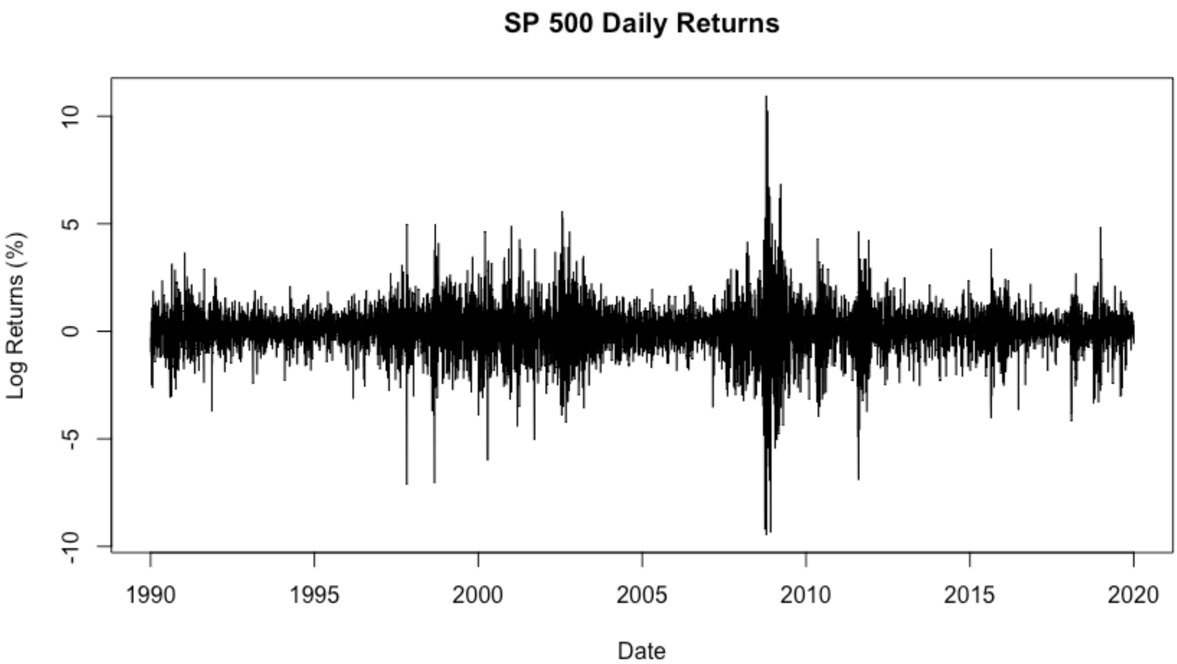
2.3 Stylized Fact 2

Volatility Clustering:

In order to know how risky a financial asset is, we need to quantify the risk in one way or the other. The most used method to quantify risk in financial assets is to estimate the standard deviation of a sample of the financial asset returns. This is a very useful way to estimate risk in financial assets, however, the sample standard deviation is a sample average and assumes that the volatility of the returns is constant over time.

Figure 3 shows the daily returns on the S&P 500 Index from 02/01/1990 – 31/12/2019. We clearly see that there are periods with higher volatility and other periods with lower volatility. This is called volatility clustering and is common for most financial assets (Franke, Härdle and Hafner 2008, 228).

Figure 3



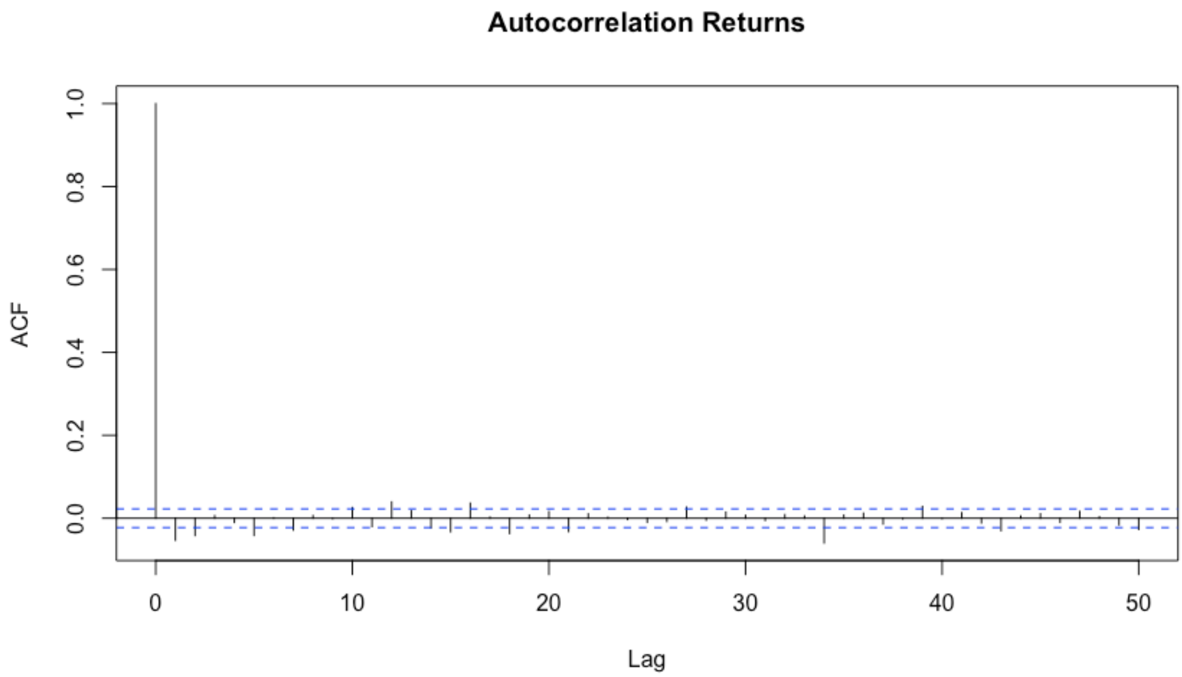
2.4 Stylized Fact 3

Absence of Autocorrelation in Returns:

The autocorrelation is a measure of how a time series is correlated with its own lags. If there exists high autocorrelation between a series of returns and its previous lags, then it would be possible to predict the returns of the series in the next time period. This would be very useful if we want to make money in financial markets, however, this is unfortunately not the case. For most financial assets there exist no significant autocorrelation between returns and its previous lags. Thus, there is no predictive information in previous returns. Even though there exists little or no predictive information in previous lags it does not mean that we cannot model the mean in order to make better portfolio decisions (Franke, Härdle and Hafner 2008, 228).

Figure 4 shows the autocorrelation between the returns on the S&P 500 Index and its previous 50 lags. As we clearly see there is no significant correlation between the returns and previous lags. The first autocorrelation is the autocorrelation with zero lags, which is the series correlated with itself and is obviously equal to one.

Figure 4



2.5 Stylized Fact 4

Small and Decreasing Autocorrelation in Squared and Absolute Returns:

Unlike with returns of financial assets, we see autocorrelation in squared and absolute returns of financial assets. By squaring or taking the absolute value of returns we convert the negative returns into positive values. The variance of the returns is defined as the squared deviation from the mean and the standard deviation of returns are defined as the square root of the variance. The squared and absolute return for a specific day (or week, month etc.) are good proxies for the variance and the standard deviation. This means that autocorrelation in squared and absolute returns is an indicator that there is autocorrelation in volatility of returns (Franke, Härdle and Hafner 2008, 228).

As we see in Figure 5 and Figure 6, the squared – and absolute returns shows small and decreasing autocorrelation. This means that volatilities are correlated with previous lags, days with high volatility is followed by days with high volatility, and days with low volatility is followed by days with low volatility. This is the same as we saw in Stylized Fact 2 (volatility clustering).

Figure 5

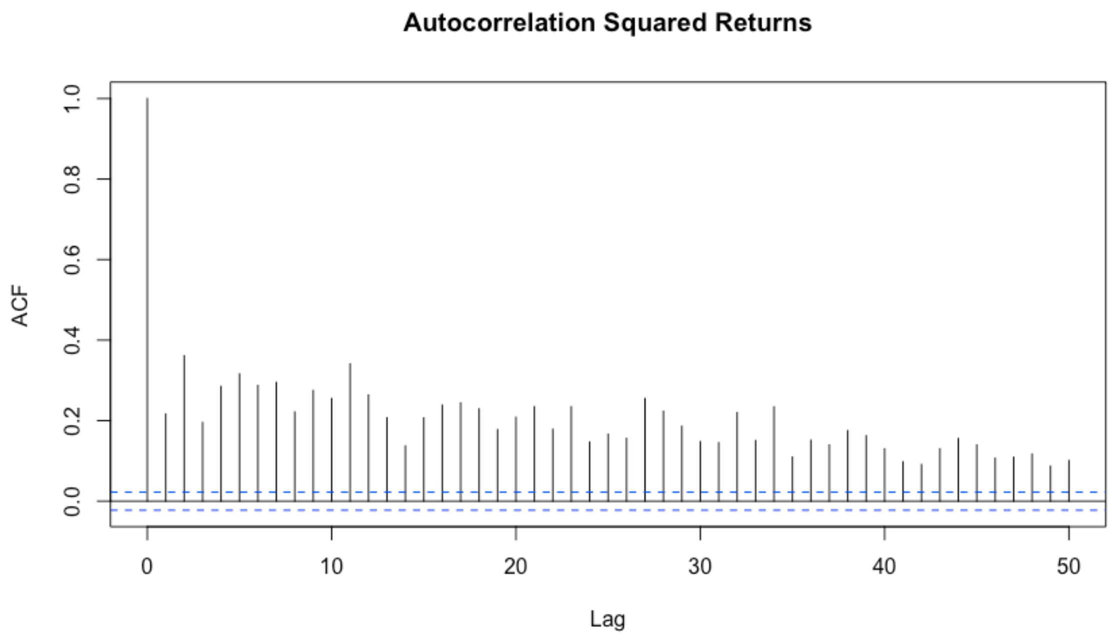
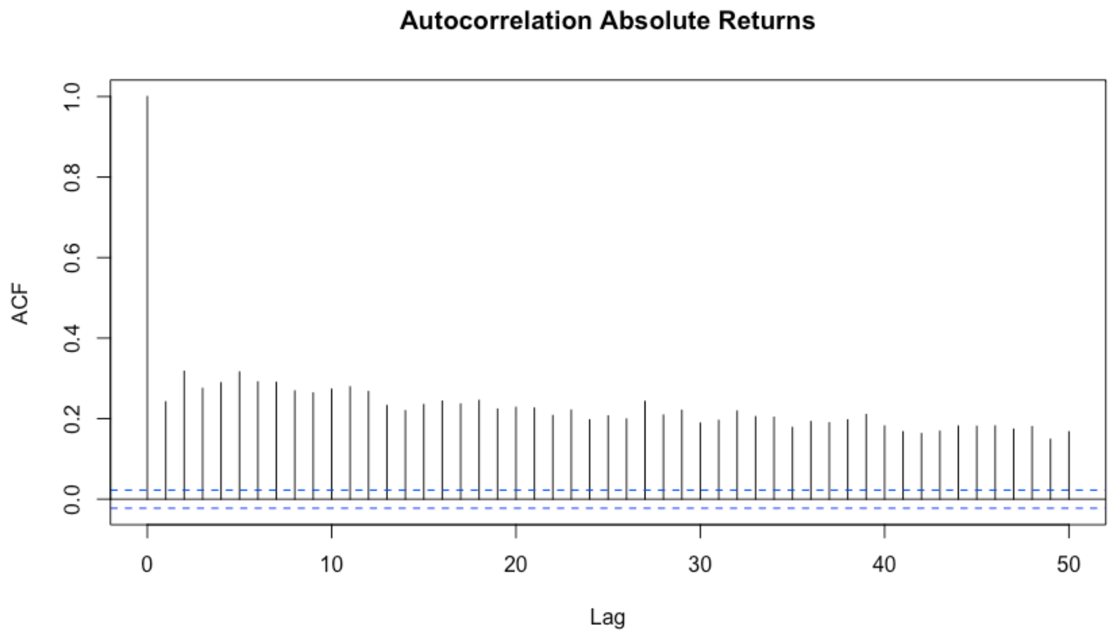


Figure 6



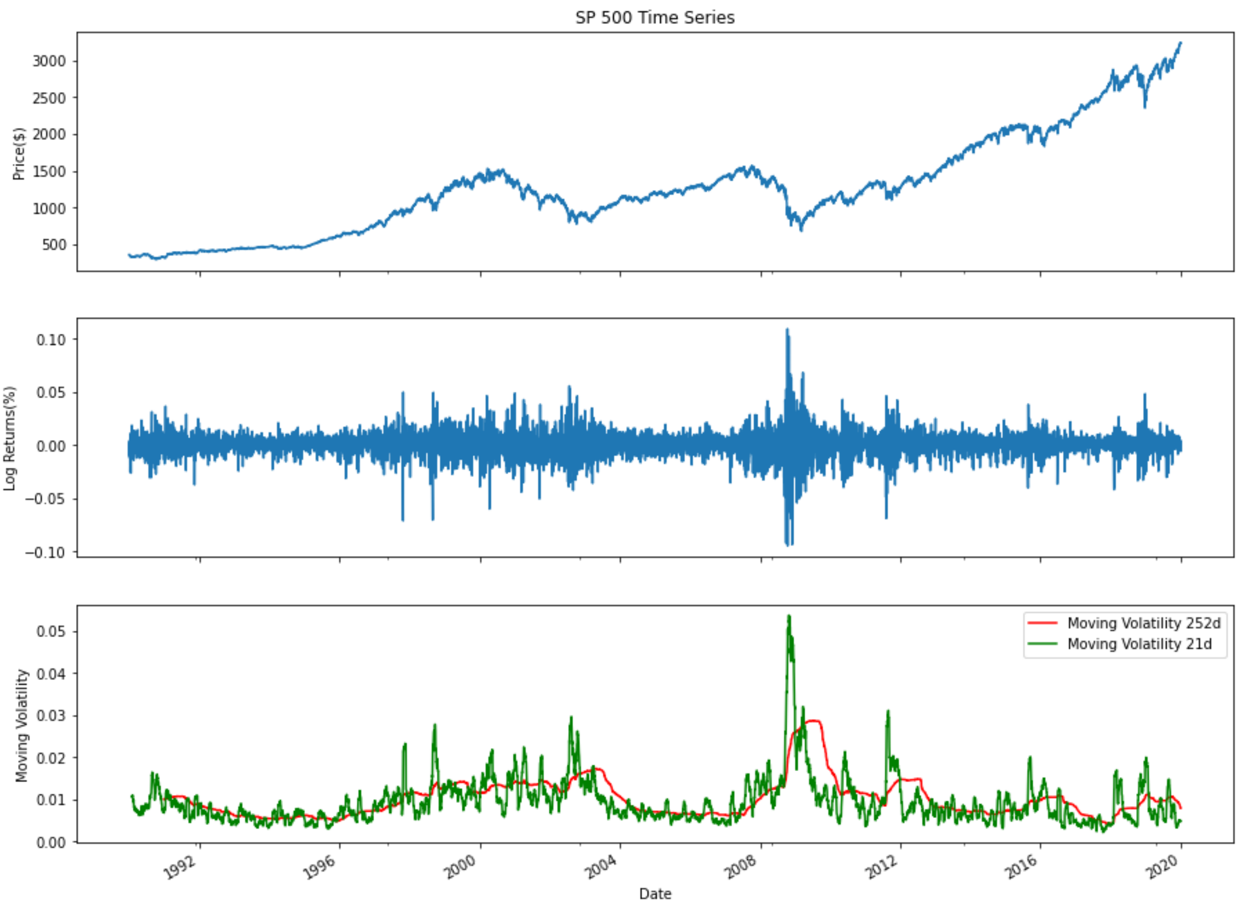
2.6 Stylized Fact 5

Leverage Effect:

We have already seen that the distribution of financial asset returns is usually not normally distributed. But is it still symmetrical around the mean? Most of the time the distribution of returns is skewed to the left which means that negative returns are more likely than positive returns. This is called the leverage effect and refers to the tendency of returns to be negatively correlated with volatility. The negative correlation comes from the fact that when stock prices fall the debt/equity ratio increases, which is associated with higher risk (Franke, Härdle and Hafner 2008, 252).

Figure 7 displays the price, returns and rolling volatility of the S&P 500 Index between 02/01/1990 – 31/12/2019. We see that the volatility is high when prices rapidly increase or decrease, but the volatility is much higher in periods when prices go down, compared to when prices go up. This is evidence of the leverage effect.

Figure 7



3. Methodology

We will use the stylized facts about financial asset returns to try to improve our portfolio selection process. When we know the statistical attributes of financial asset returns, we can use this to model the expected return and the volatility of the individual assets. Our main focus will be on modelling the volatility, and we will try to forecast the volatility in the future in order to construct optimal portfolios.

3.1 Portfolio Optimization

The motivation for investing in a portfolio of risky assets, rather than investing in a single risky asset, is that we can remove some of the risk while keeping the same expected return. This is possible because different risky assets have different sources of risk, and as long as the correlation between the risky assets returns is less than one, we have diversification effects when combining the risky assets in a portfolio.

3.1.1 Mean – Variance Efficient Frontier

We will use the mean-variance criterion in order to construct the optimal portfolio for a set of risky assets. The selection of portfolios is based on the means and variances of the risky assets returns. The optimal portfolio is the portfolio that gives the highest expected return for a given level of variance or standard deviation (risk), or the portfolio that gives the lowest variance or standard deviation, for a given level of expected return. If we assume that all investors are able to lend and borrow at a risk-free rate of return, then, there exist one single optimal portfolio for all investors.

The analytical technique for deriving the feasible set of risky assets was developed by Harry Markowitz (1952). By assigning different weights between zero and one (sum of alle weights must be equal to one) to the risky assets in a portfolio we can construct the feasible set of the risky assets. The feasible set is all the portfolios that we can construct given the assets available. In order to derive the feasible set, we need the expected returns, variances and/or standard deviations of the different portfolios.

The expected return of a portfolio consisting of M risky assets is:

$$E(r_P) = \sum_{i=1}^{M} w_i \, E(r_i) \tag{3}$$

$$E(r_P) = \boldsymbol{w^T E(r)} \tag{4}$$

The variance of a portfolio consisting of M risky assets is:

$$\sigma_P^2 = \sum_{i=1}^{M} (w_i)^2 \underbrace{Var(r_i)}_{\sigma_{ii}} + 2 \sum_{i=1}^{M} \sum_{j=i+1}^{M} w_i w_j \underbrace{Cov(r_i, r_j)}_{\sigma_{ij}} \tag{5}$$

$$\sigma_P^2 = \boldsymbol{w^T S w} \tag{6}$$

The standard deviation of a portfolio consisting of M risky assets is:

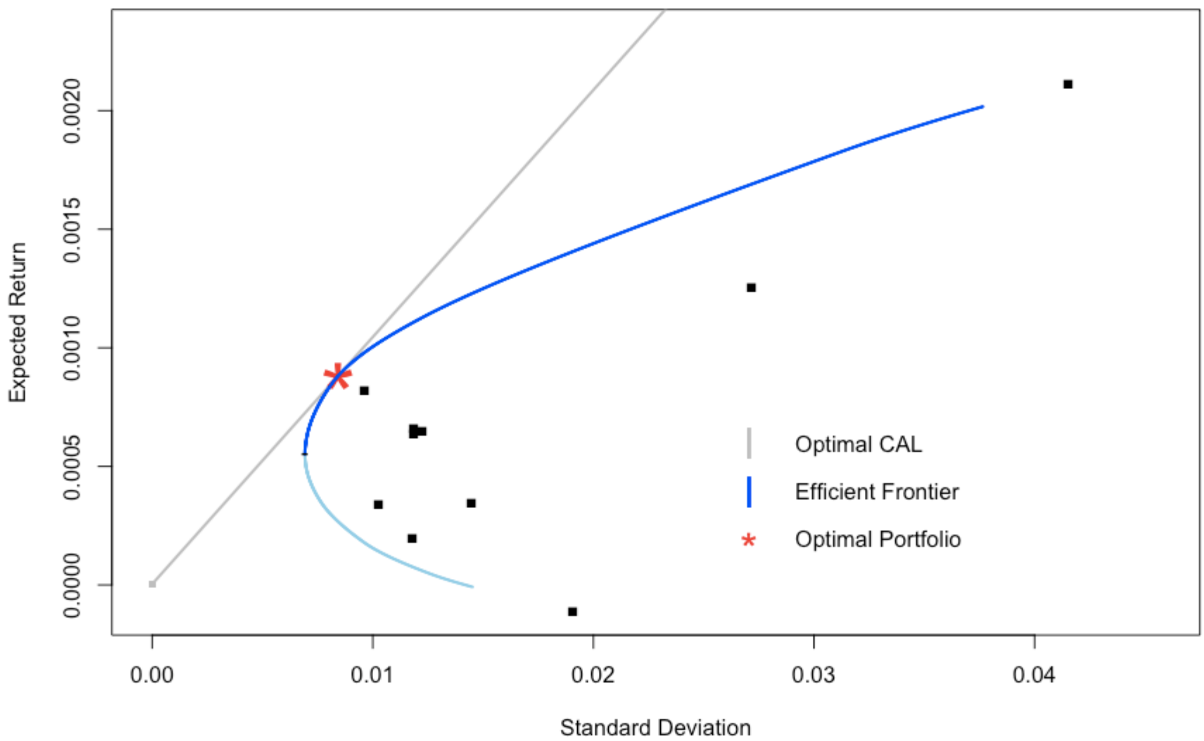
$$\sigma_P = \sqrt{\sigma_P^2} \tag{7}$$

$$\sigma_P = \sqrt{\boldsymbol{w^T S w}} \tag{8}$$

- $\boldsymbol{w}$ is an $M \times 1$ vector of weights
- $\boldsymbol{E(r)}$ is an $M \times 1$ vector of expected returns
- $\boldsymbol{S}$ is an $M \times M$ variance-covariance matrix

We can plot the standard deviations against the expected returns of the different portfolios to form the risk-return feasible set (see Figure 8).

Figure 8



3.1.2 Optimal Portfolio

The part of the graph in Figure 8 that connects all the northwesternmost portfolios is called the efficient frontier. The efficient frontier is the set of portfolios that offers the highest expected return for each level of portfolio standard deviation.
The optimal portfolio is the portfolio on the efficient frontier with the highest Sharpe ratio. The Sharpe ratio, Θ, was first proposed by William Sharpe (1966) and is the ratio of excess expected return per unit of standard deviation (risk).

The portfolio optimization problem can be written:

$$\max_{w} \Theta = \frac{E(r_P) - r_f}{\sigma_P} \tag{9}$$

$$\max_{w} \Theta = \frac{\boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{E}(\boldsymbol{r}) - r_f}{\sqrt{\boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{S} \boldsymbol{w}}} \tag{10}$$

where

$$\sum_{i=1}^{M} w_i = 1, \qquad \text{no short:} \qquad w_i \geq 0, \qquad i = 1, \dots, M$$

In order to find the portfolio weights that maximizes the Sharpe ratio we use a method called "Monte Carlo Simulation" (Raychaudhuri 2008) to generate 100,000 random portfolios that satisfies the weight conditions of the portfolio optimization problem. The risk-return feasible set in Figure 8 is generated this way. The portfolio with the weights that generate the highest Sharpe ratio is our solution for the optimization problem.

If investors are able to lend or borrow at the risk-free rate of return, the optimal portfolio in Figure 8 is the optimal portfolio for all investors. Risk averse investors will invest some of their funds in the optimal risky portfolio and the rest at the risk-free rate. Risk seeking investors will borrow extra funds at the risk-free rate and invest all their funds in the optimal risky portfolio. In this way, all investors can achieve the same Sharpe ratio for any level of risk. This is the optimal capital allocation line (CAL). In the optimization problem in (9) and (10) we have three inputs: the expected return – and the standard deviation of the portfolio, and the risk-free rate. The risk-free rate is given, while the expected return and the standard deviation of the portfolio are estimates. In order to find the optimal risky portfolio, we need accurate estimates of the expected returns and the standard deviations of returns of the risky assets in the portfolio.

3.2 Equally Weighted Portfolio

To evaluate the performance of the different portfolios we would like to compare them to a benchmark portfolio. For the benchmark portfolio we will use an equally weighted portfolio, or a so called "naïve" portfolio strategy where we assign equal weights to all the assets in the portfolio. It is called naïve because we make no assumptions about the risky assets prior to assigning the weights.

The portfolio specifications can be written:

$$E(r_{EWP}) = \boldsymbol{w^T E(r)} \tag{11}$$

$$\sigma_{EWP} = \sqrt{\boldsymbol{w^T S w}} \tag{12}$$

$$w_i = \frac{1}{M}, \qquad \sum_{i=1}^{M} w_i = 1$$

3.3 Sample Optimal Portfolio

The most common and easiest way to estimate expected returns, variance of returns and covariance between returns is the sample estimates. These are average numbers and are assumed to be constant over time. Since these estimates are not conditional on time, they are called unconditional estimates. The estimate for the expected returns is the sample mean, the estimate for the variance of returns is the sample variance, and the estimate for the covariance between returns is the sample covariance. These estimates are defined below.

Sample mean:

$$E(r_i) = \frac{1}{N}\sum_{t=1}^{N} r_{i,t} \tag{13}$$

Sample variance:

$$Var(r_i) = \frac{1}{N-1}\sum_{t=1}^{N}\left(r_{i,t} - E(r_i)\right)^2 \tag{14}$$

Sample covariance:

$$Cov(r_i, r_j) = \frac{1}{N-1}\sum_{t=1}^{N}\left(r_{i,t} - E(r_i)\right)\left(r_{j,t} - E(r_j)\right) \tag{15}$$

After calculating the sample estimates we have all we need in order to solve the portfolio optimization problem. The problem can be written:

$$\max_{w} \Theta = \frac{E(r_{SOP}) - r_f}{\sigma_{SOP}} \tag{16}$$

$$E(r_{SOP}) = \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{E}(\boldsymbol{r}) \tag{17}$$

$$\sigma_{SOP} = \sqrt{\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{S}\boldsymbol{w}} \tag{18}$$

- $\boldsymbol{w}$ is an $M \times 1$ vector of weights
- $\boldsymbol{E}(\boldsymbol{r})$ is an $M \times 1$ vector of sample means
- $\boldsymbol{S}$ is an $M \times M$ sample variance-covariance matrix

3.4 Conditional Estimates

In the sample portfolio optimization problem, we use the unconditional estimates for the expected return, variance, and covariance, which are constant over time. As we have showed in stylized facts about financial asset returns (2.2 – 2.6) this is not necessarily the case, especially for the variance. The variance tends to cluster with periods of high volatility and periods with low volatility. What we want to do is to model the asset returns in order to get more precise estimates. These models are time dependent, which means that the estimates are conditional on time, hence, they are called conditional estimates (Brooks 2014, 288-289, 431).

The returns can be modeled as:

$$\begin{cases} r_t = \mu + ARMA_{t-1} + \varepsilon_t \\ \varepsilon_t = z_t\sqrt{h_t} \\ z_t \approx (i.i.d.) \\ h_t = GARCH_{t-1} \end{cases} \tag{19}$$

- r_t is the conditional return at time t
- μ is the unconditional mean
- z_t are the standardized residuals
- h_t is the conditional variance at time t
- $ARMA_{t-1}$ is some sort of ARMA process
- $GARCH_{t-1}$ is some sort of GARCH process

In (19) the returns are assumed to be following some sort of autoregressive moving average (ARMA) process. If we model the returns without an ARMA process, then $ARMA_{t-1}$ is equal to zero and the model is reduced to:

$$
\begin{cases} r_t = \mu + \varepsilon_t \\ \varepsilon_t = z_t\sqrt{h_t} \\ z_t \approx (i.i.d.) \\ h_t = GARCH_{t-1} \end{cases} \tag{20}
$$

3.5 ARMA Process in Financial Returns

As we have seen in stylized fact 3 (2.4), there tends to be low autocorrelation in the lags of financial asset returns, so there is probably little (if any) predictive information in previous lags of the returns. Even if there is little predictive information in previous lags, it is not unusual to model the returns with some sort of autoregressive moving average (ARMA) process. The ARMA process models the returns on previous lags, p, of the returns (r_{t-i}) and the previous lags, q, of the error term (ε_{t-i}). The order of lags must be specified before estimating the model.

The ARMA (p, q) process can be written:

$$
r_t = \sum_{i=1}^{p} \phi_i \, r_{t-i} + \sum_{i=1}^{q} \theta_i \, \varepsilon_{t-i} + \varepsilon_t \tag{21}
$$

The estimation of the parameters ϕ and θ of the specified model can be done using OLS (ordinary least squares) or maximum likelihood estimation (Brooks 2014, 273).

To find the order of (p, q) that gives us the “best” ARMA model for explaining our sample of data, we use something called information criterion.
Two of the most famous information criteria are the Akaike Information Criteria (AIC) and Bayes Information Criteria (BIC):

$$
AIC = -2\ln L + 2k \tag{22}
$$

$$
BIC = -2\ln L + k\ln T \tag{23}
$$

- L is the maximized value of the log-likelihood function
- k is the number of free parameters in the model
- T is the number of observations

The best model is the one with the lowest AIC or the model with lowest BIC.
The AIC has a tendency to choose models with too many lags. BIC on the other hand punishes complexity and will usually deliver a model with lower lags than the AIC. When we check the AIC and BIC for the ten different stocks we have chosen for our analysis, we see that there is some evidence of ARMA processes in the returns.

Table 2

```
AIC: MMM                                BIC: MMM
 ARIMA(1,0,0)(0,0,0)[0] intercept        ARIMA(1,0,0)(0,0,0)[0]
---------------------------------       ---------------------------------
AIC: VZ                                 BIC: VZ
 ARIMA(0,0,0)(0,0,0)[0] intercept        ARIMA(0,0,0)(0,0,0)[0]
---------------------------------       ---------------------------------
AIC: NEE                                BIC: NEE
 ARIMA(0,0,0)(0,0,0)[0] intercept        ARIMA(0,0,0)(0,0,0)[0]
---------------------------------       ---------------------------------
AIC: PM                                 BIC: PM
 ARIMA(0,0,0)(0,0,0)[0] intercept        ARIMA(0,0,0)(0,0,0)[0]
---------------------------------       ---------------------------------
AIC: GE                                 BIC: GE
 ARIMA(0,0,0)(0,0,0)[0] intercept        ARIMA(0,0,0)(0,0,0)[0]
---------------------------------       ---------------------------------
AIC: AMT                                BIC: AMT
 ARIMA(2,0,2)(0,0,0)[0] intercept        ARIMA(1,0,0)(0,0,0)[0]
---------------------------------       ---------------------------------
AIC: TGT                                BIC: TGT
 ARIMA(0,0,0)(0,0,0)[0]                  ARIMA(0,0,0)(0,0,0)[0]
---------------------------------       ---------------------------------
AIC: AMD                                BIC: AMD
 ARIMA(1,0,1)(0,0,0)[0]                  ARIMA(0,0,0)(0,0,0)[0]
---------------------------------       ---------------------------------
AIC: MO                                 BIC: MO
 ARIMA(1,0,0)(0,0,0)[0] intercept        ARIMA(0,0,0)(0,0,0)[0] intercept
---------------------------------       ---------------------------------
AIC: CCI                                BIC: CCI
 ARIMA(3,0,2)(0,0,0)[0] intercept        ARIMA(0,0,0)(0,0,0)[0]
---------------------------------       ---------------------------------
```

The AIC suggests some lags in some of the stocks, while the BIC suggests one p lag in two of the stocks. Since the AIC has a tendency to choose models with too many lags, the BIC suggests very few lags, and our main focus is to model the volatility, we will assume that all the stocks follow an ARMA(0,0) process.

3.6 Univariate Volatility Models

The most popular volatility models used in finance are the volatility models in the autoregressive conditionally heteroscedastic (ARCH) family. The ARCH model was first developed by Robert Engle (1982). Under the ARCH model the autocorrelation in volatility is modeled by allowing the conditional variance of the error term to be dependent on the previous value of the squared error.

The ARCH model can be written:

$$h_t = \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 \tag{24}$$

where

$$q \geq 0, \qquad \alpha_0 > 0, \qquad \alpha_i \geq 0, \qquad i = 1, \dots, q$$

3.6.1 The GARCH Model

Later Tim Bollerslev (1986) developed a generalized version of Engle's ARCH model, namely the generalized autoregressive conditionally heteroscedastic (GARCH) model. The ARCH model works well in order to model the volatility in returns of financial assets, but a drawback of the ARCH model is that in order to capture all of the dependence in the conditional variance the order of lags might become very large. Bollerslev solved this by allowing the conditional variance to be dependent upon previous own lags. The GARCH model has this nice property that it usually needs few lags of p and q in order to model the conditional volatility quite well. Hansen and Lunde (2005) found no evidence that the GARCH model in its simplest form, the GARCH(1, 1), was inferior in forecasting out of sample volatility in DM-$ exchange rate, when compared to 229 other ARCH-type models.

The GARCH model can be written:

$$h_t = \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{i=1}^{p} \beta_j h_{t-i} \tag{25}$$

where

$p \geq 0, \qquad q \geq 0$

$\alpha_0 > 0, \qquad \alpha_i \geq 0, \qquad \beta_j \geq 0$

The parameters α_0, α_i, and β_i in the GARCH model are estimated using maximum likelihood estimation (see Appendix A).

3.6.2 Distribution of Errors

As already mentioned, a common assumption in finance is that financial asset returns are normally distributed. Returns are usually leptokurtic and resembles more of a student t distribution. We can take this into account in the maximum likelihood estimation of the GARCH model parameters (see Appendix A).

3.6.3 The GJR-GARCH Model

In section 2.6 we saw that financial asset returns tends to be negatively correlated with volatility, if this is the case then we would like to incorporate this in our model. The GJR-GARCH was introduced by Lawrence Glosten, Ravi Jagannathan and David Runkle (1993) and is a simple extension of the linear GARCH model. The model allows us to model any asymmetries (leverage effect) that may be present in financial asset returns.

The GJR-GARCH model can be written:

$$h_t = \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{i=1}^{p} \beta_i h_{t-i} + \sum_{i=1}^{o} \gamma_i \varepsilon_{t-i}^2 I_{t-i} \tag{26}$$

where

$I_{t-i} = 1 \quad \text{if} \quad \varepsilon_{t-i} < 0, \qquad I_{t-i} = 0 \quad \text{if} \quad \varepsilon_{t-i} \geq 0$

In the GJR-GARCH model I_{t-i} is a dummy variable that is equal to 1 if ε_{t-i} is positive and 0 if ε_{t-i} is negative. For a leverage effect we will have $\gamma_i > 0$.

3.6.4 Order of GARCH Model

To find the order of (p, o, q) that gives us the "best" GARCH model for our data, we could use the information criterion which we used to find the (p, q) order of the "best" ARMA model in section 3.5. The problem with using information criterion like AIC and BIC is that they return the order of the "best" model based on the sample we have provided. They are in-sample performance estimates and there is no guarantee that the "best" model in-sample is the "best" model out-of-sample.
Another problem is that the order of (p, o, q) can vary given the sample we calculate the AIC and BIC from. Since we want to forecast the volatility for 1,008 trading days, we would have to calculate the AIC and BIC for 1,008 samples in order to update our model with new information.

Table 3 displays the order of the "best" model based on the first sample (03/01/2011 – 29/1272016) in our rolling forecast origin (section 3.8) for the 10 stocks we have chosen. We have calculated the AIC and BIC for the GARCH with normally distributed errors, the GARCH model with t distributed errors, and the GJR-GARCH with t distributed errors.

Instead of making any assumptions of what the "best" model will be in the future we will model the volatility with several different orders of (p, o, q):

- GARCH(1,1)
- GARCH(2,2)
- GJR-GARCH(1,1,1)
- GJR-GARCH(2,1,2)
- GJR-GARCH(2,2,2)

Table 3

AIC: GARCH(N)

	"Best" Model
MMM	GARCH(2,4)
VZ	GARCH(1,1)
NEE	GARCH(2,4)
PM	GARCH(1,1)
GE	GARCH(1,2)
AMT	GARCH(1,2)
TGT	GARCH(2,1)
AMD	GARCH(3,1)
MO	GARCH(2,1)
CCI	GARCH(1,1)

BIC: GARCH(N)

	"Best" Model
MMM	GARCH(1,1)
VZ	GARCH(1,1)
NEE	GARCH(1,1)
PM	GARCH(1,1)
GE	GARCH(1,1)
AMT	GARCH(1,1)
TGT	GARCH(1,1)
AMD	GARCH(3,1)
MO	GARCH(1,1)
CCI	GARCH(1,1)

AIC: GARCH(t)

	"Best" Model
MMM	GARCH(1,1)
VZ	GARCH(1,1)
NEE	GARCH(2,4)
PM	GARCH(1,4)
GE	GARCH(1,1)
AMT	GARCH(1,2)
TGT	GARCH(1,1)
AMD	GARCH(1,1)
MO	GARCH(1,1)
CCI	GARCH(1,1)

BIC: GARCH(t)

	"Best" Model
MMM	GARCH(1,1)
VZ	GARCH(1,1)
NEE	GARCH(1,1)
PM	GARCH(1,1)
GE	GARCH(1,1)
AMT	GARCH(1,1)
TGT	GARCH(1,1)
AMD	GARCH(1,1)
MO	GARCH(1,1)
CCI	GARCH(1,1)

AIC: GJR-GARCH(t)

	"Best" Model
MMM	GARCH(1,2,2)
VZ	GARCH(1,1,1)
NEE	GARCH(1,1,4)
PM	GARCH(1,1,1)
GE	GARCH(1,2,2)
AMT	GARCH(1,2,3)
TGT	GARCH(1,1,1)
AMD	GARCH(1,1,1)
MO	GARCH(1,2,4)
CCI	GARCH(1,3,3)

BIC: GJR-GARCH(t)

	"Best" Model
MMM	GARCH(1,1,1)
VZ	GARCH(1,1,1)
NEE	GARCH(1,1,1)
PM	GARCH(1,1,1)
GE	GARCH(1,1,1)
AMT	GARCH(1,1,1)
TGT	GARCH(1,1,1)
AMD	GARCH(1,1,1)
MO	GARCH(1,1,1)
CCI	GARCH(1,1,1)

Table 4 gives an overview of the strategies will use to construct our portfolios.

Table 4: Strategy Overview

Strategy	Univariate GARCH	Multivariate GARCH	Error Distribution
1. Passive (S&P 500)	-	-	-
2. Equally Weighted	-	-	-
3. Sample Weighted	-	-	-
4. GARCH (1,1) (N)	GARCH (1,1)	CCC – GARCH	Normal
5. GARCH (1,1) (t)	GARCH (1,1)	CCC – GARCH	Student t
6. GJR-GARCH (1,1,1) (t)	GJR-GARCH (1,1,1)	CCC – GARCH	Student t
7. GARCH (2,2) (N)	GARCH (2,2)	CCC – GARCH	Normal
8. GARCH (2,2) (t)	GARCH (2,2)	CCC – GARCH	Student t
9. GJR-GARCH (2,1,2) (t)	GJR-GARCH (2,1,2)	CCC – GARCH	Student t
10. GJR-GARCH (2,2,2) (t)	GJR-GARCH (2,2,2)	CCC – GARCH	Student t

3.7 Multivariate GARCH Models

Multivariate GARCH models use the conditional estimates of univariate GARCH models to form a multivariate model. The first multivariate GARCH model, the constant conditional correlation (CCC) GARCH model, was introduced by Tim Bollerslev (1990). Later Robert Engle (2002) developed a more complex version of Bollerslev's CCC GARCH model, namely the dynamic conditional correlation (DCC) GARCH model. The CCC GARCH model assumes that the correlation matrix is constant over time, while the DCC GARCH assumes that the correlation matrix is not constant over time. For our analysis we will focus on the CCC GARCH model.

3.7.1 The CCC-GARCH Model

As stated above the CCC GARCH model assumes that the correlations matrix, **R**, is constant over time when estimating the model.

The CCC-GARCH model can be written:

$$\mathbf{r}_t = \mathbf{C}\mathbf{r}_{t-1} + \boldsymbol{\epsilon}_t \tag{27}$$

$$\boldsymbol{\epsilon}_t = \mathbf{H}_t^{1/2}\boldsymbol{v}_t$$

$$\mathbf{H}_t = \mathbf{D}_t^{1/2}\mathbf{R}\mathbf{D}_t^{1/2} \tag{28}$$

$$\mathbf{D}_t = \begin{pmatrix} h_{11,t} & 0 & \cdots & 0 \\ 0 & h_{22,t} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_{ii,t} \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 1 & \rho_{12} & \cdots & \rho_{1i} \\ \rho_{21} & 1 & \cdots & \rho_{2i} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{i1} & \rho_{i2} & \cdots & 1 \end{pmatrix}$$

$$\mathbf{H}_t = \begin{pmatrix} h_{11,t} & h_{12,t} & \cdots & h_{1j,t} \\ h_{21,t} & h_{22,t} & \cdots & h_{2j,t} \\ \vdots & \vdots & \ddots & \vdots \\ h_{i1,t} & h_{i2,t} & \cdots & h_{ii,t} \end{pmatrix}$$

$$h_{ij,t} = \rho_{ij}\left(h_{ii,t}h_{jj,t}\right)^{1/2} \tag{29}$$

- $\mathbf{r}_t$ is an $M \times 1$ vector of returns
- $\mathbf{C}$ is an $M \times k$ matrix of model parameters
- $\mathbf{r}_{t-1}$ is an $M \times 1$ vector of lagged returns
- $\boldsymbol{v}_t$ is an $M \times 1$ vector of normal (or student t), independent, and identically distributed innovations

- $\mathbf{D}_t$ is a diagonal matrix of conditional univariate variances
- $\mathbf{R}$ is a matrix of time-invariant unconditional correlations of the standardized residuals
- $\mathbf{H}_t$ is the forecast of the variance-covariance matrix
- $h_{ij,t}$ is the conditional covariance between the returns of asset i and j
- $h_{ii,t}$ is the conditional variance of the returns of asset i
- ρ_{ij} is the correlation between the returns of asset i and j

The estimates of the parameters in the CCC-GARCH model are estimated with maximum likelihood estimation (see Appendix A).

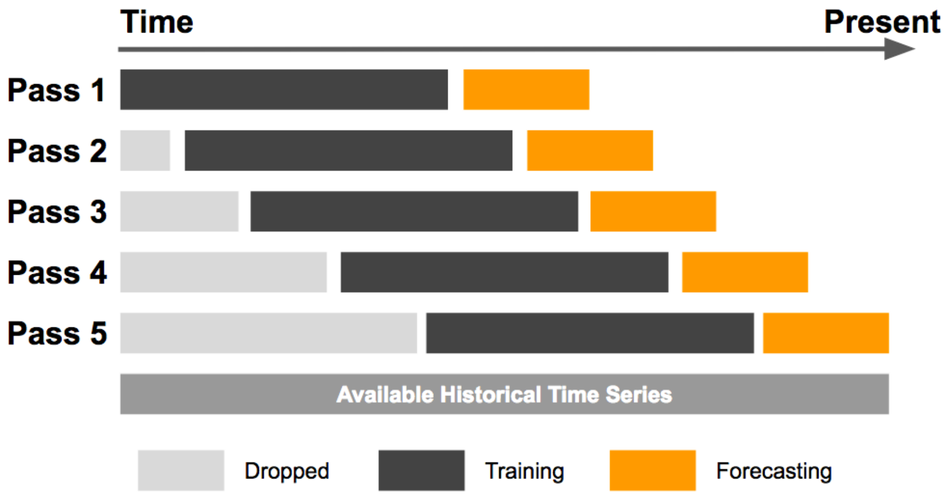
3.8 Rolling Forecast Origin

We want to forecast the volatility and variance-covariance matrix of ten different stocks for 1,008 trading days in order in order to construct optimal portfolios. Forecasting something that is going to happen tomorrow sounds like a difficult task, forecasting something that is going to happen in four years' time sounds impossible, and it probably is.

We will use a rolling forecast origin; we forecast the volatility and variance-covariance matrix one day ahead. When we know what really happened that day, we use this information and update our model to forecast the volatility and the variance-covariance matrix for the next day after that. We repeat this every day in order to update our model to make, hopefully, better forecasts.

Figure 9 shows how the rolling forecast origin works. We forecast one day ahead, the next day we include the new information from that day and drop the information from the first day of our sample. Next day we repeat the same procedure (Tashman 2000).

Figure 9



3.9 Evaluation of Volatility Forecasts

One way to evaluate the different portfolio strategies we have chosen is to look at the cumulative portfolio returns over the investment period. This gives us a picture of how the portfolio have performed in terms of actual returns but can be misleading if we want to generalize our findings to other financial assets and other time periods. This is because we treat the returns as random variables and the best strategy in terms of returns for some financial assets in one period, does not necessarily mean that this is the best strategy for other financial assets in other time periods. In other words, maybe we were just lucky when we constructed the portfolios.

Since the mean – variance portfolio optimization relies heavily on the estimates or forecasts of the volatility of the financial assets we would like to see if there is any relationship between how accurate we can forecast the volatility and the performance of the portfolio. If the strategy that produces the highest portfolio returns also forecasted the volatility with most accuracy, we would have more confidence that this strategy is a good strategy for other financial assets in other time periods as well.

To evaluate the volatility forecast we use the root mean squared error (RMSE) of the predicted values compared to the actual values.

The RMSE can be written:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Predicted_i - Actual_i)^2}{N}} \tag{30}$$

The predicted values are our forecasts of the volatility. Since volatility is not directly observable, we do not know what the actual volatility is. Thankfully, there exists proxies which we can use to represent the actual volatility for a specific time period. The most used proxy for the variance of returns is the squared returns and for the standard deviation the absolute returns. Even though these measures can be quite noisy, argued by Hansen and Lunde (2005), they are accessible and easy to calculate when we use returns over a longer time period (days, weeks, months etc.).

To evaluate the volatility forecasts we can write (30) as:

$$RMSE(var) = \sqrt{\frac{\sum_{i=1}^{N}(h_i - r_i^2)^2}{N}} \tag{31}$$

$$RMSE(std) = \sqrt{\frac{\sum_{i=1}^{N}\left(\sqrt{h_i} - |r_i|\right)^2}{N}} \tag{32}$$

The strategy with the lowest $RMSE$ is the strategy that was able to forecast the volatility with highest accuracy over the investment period.

3.10 Recap

Construct Markowitz mean-variance optimal portfolios:

- Consisting of 10 stocks
- Maximize Sharpe Ratio
- No short selling
- Unconditional mean

- Forecasts of univariate volatility
- Forecasts of multivariate variance-covariance matrix
- Rolling forecast for 1,008 trading days
- The optimization problem can be written:

$$\max_{w} \Theta = \frac{\mu - r_f}{\sqrt{h_{t+1}}} \tag{33}$$

$$\max_{w} \Theta = \frac{\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{\mu} - r_f}{\sqrt{\boldsymbol{w}^{\boldsymbol{T}}\mathbf{H}_{t+1}\boldsymbol{w}}} \tag{34}$$

where

$$\sum_{i=1}^{M} w_i = 1, \qquad w_i \geq 0, \qquad i = 1, \dots, M$$

- Evaluate volatility forecasts

4. Data

The data set consists of the price of the S&P 500 Index, the 3-Month Treasury Bill: Secondary Market Rate, and the prices of ten stocks, which all are included in the S&P 500. The data for the S&P 500 consists of 7,812 observations (02/01/1990 – 31/12/2020), where 7,558 observations (02/01/1990 – 31/12/2019) are used to demonstrate the stylized facts of financial asset returns, and 1,009 observations (29/12/2016 – 31/12/2020) are used as proxy for the passive investment strategy. The data for the 3-Month Treasury Bill consists of 1,008 observations (28/12/2016 – 30/12/2020) and is used as the risk-free rate in the mean-variance portfolio optimization problem. For the ten stocks the estimation sample spans from 02/01/2011 – 28/12/2016 (1,509 observations) and the out-of-sample rolling forecasting period spans from 29/12/2016 – 31/12/2020 (1,009 observations).

The data for the S&P 500 and the ten stocks are downloaded from Yahoo Finance. The data for the 3-Month Treasury Bill is downloaded from the Federal Reserve Economic Data (FRED). Table 5 gives an overview of the ten stocks.

Table 5: Data Overview

Company Name	Ticker	Sector	Industry	Estimation Sample	Forecasting Sample
3M	MMM	Industrials	Specialty Industrial Machinery	02/01/2011 - 28/12/2016	29/12/2016 - 31/12/2020
Verizon Communications	VZ	Communication Services	Telecom Services	02/01/2011 - 28/12/2016	29/12/2016 - 31/12/2020
NextEra Energy	NEE	Utilities	Industry Utilities – Regulated Electric	02/01/2011 - 28/12/2016	29/12/2016 - 31/12/2020
Phillip Morris International	PM	Consumer Defensive	Tobacco	02/01/2011 - 28/12/2016	29/12/2016 - 31/12/2020
General Electric	GE	Industrials	Specialty Industrial Machinery	02/01/2011 - 28/12/2016	29/12/2016 - 31/12/2020
American Tower	AMT	Real Estate	REIT – Specialty	02/01/2011 - 28/12/2016	29/12/2016 - 31/12/2020
Target	TGT	Consumer Defensive	Discount Stores	02/01/2011 - 28/12/2016	29/12/2016 - 31/12/2020
Advanced Micro Devices	AMD	Technology	Semiconductors	02/01/2011 - 28/12/2016	29/12/2016 - 31/12/2020
Altria Group	MO	Consumer Defensive	Tobacco	02/01/2011 - 28/12/2016	29/12/2016 - 31/12/2020
Crown Castle International	CCI	Real Estate	REIT – Specialty	02/01/2011 - 28/12/2016	29/12/2016 - 31/12/2020

5. Results

Table 6 displays the performance in terms of returns for the ten different strategies over the 1,008 trading days. It contains the mean, standard deviation, minimum, maximum, cumulative, minimum cumulative, and maximum cumulative.

In terms of cumulative returns, we see that the passive and the equally weighted strategy performed quite similarly. The sample weighted strategy outperformed the passive and the equally weighted strategies, while all the GARCH strategies outperformed the passive, equally weighted, and the sample weighted strategies. The GARCH(1,1) and GARCH(2,2) with student t distributed errors performed almost identically and outperformed all the other strategies.

In terms of standard deviation (risk) the passive and equally weighted strategies performed best, the sample weighted strategy performed the worst, and all the GARCH strategies performed quite similarly.

Table 6: Strategy Performance

Strategy	Mean	Std.	Min.	Max.	Cum.	Min. Cum.	Max. Cum.
1. Passive (S&P 500)	0.0509%	1.2976%	-12.77%	8.97%	**53.26%**	-6.24%	53.26%
2. Equally Weighted	0.0490%	1.2502%	-10.77%	8.30%	**51.41%**	-6.49%	52.15%
3. Sample Weighted	0.0668%	1.4164%	-12.69%	10.07%	**77.04%**	-2.22%	77.04%
4. GARCH (1,1) (N)	0.0815%	1.3821%	-13.34%	9.43%	**106.42%**	-0.72%	106.42%
5. GARCH (1,1) (t)	0.0850%	1.3726%	-13.34%	9.30%	**113.89%**	-0.65%	113.89%
6. GJR-GARCH (1,1,1) (t)	0.0803%	1.3817%	-13.42%	9.31%	**103.86%**	-0.65%	103.86%
7. GARCH (2,2) (N)	0.0776%	1.3814%	-13.37%	9.30%	**98.45%**	-0.73%	98.45%
8. GARCH (2,2) (t)	0.0849%	1.3717%	-13.24%	9.39%	**113.75%**	-0.65%	113.75%
9. GJR-GARCH (2,1,2) (t)	0.0821%	1.3826%	-13.46%	9.53%	**107.44%**	-0.65%	107.44%
10. GJR-GARCH (2,2,2) (t)	0.0812%	1.3841%	-13.30%	9.56%	**105.53%**	-0.69%	105.53%

Figure 10 displays how the cumulative returns evolved over the 1,008 trading days for the passive, equally weighted, and sample weighted strategies.

Figure 10

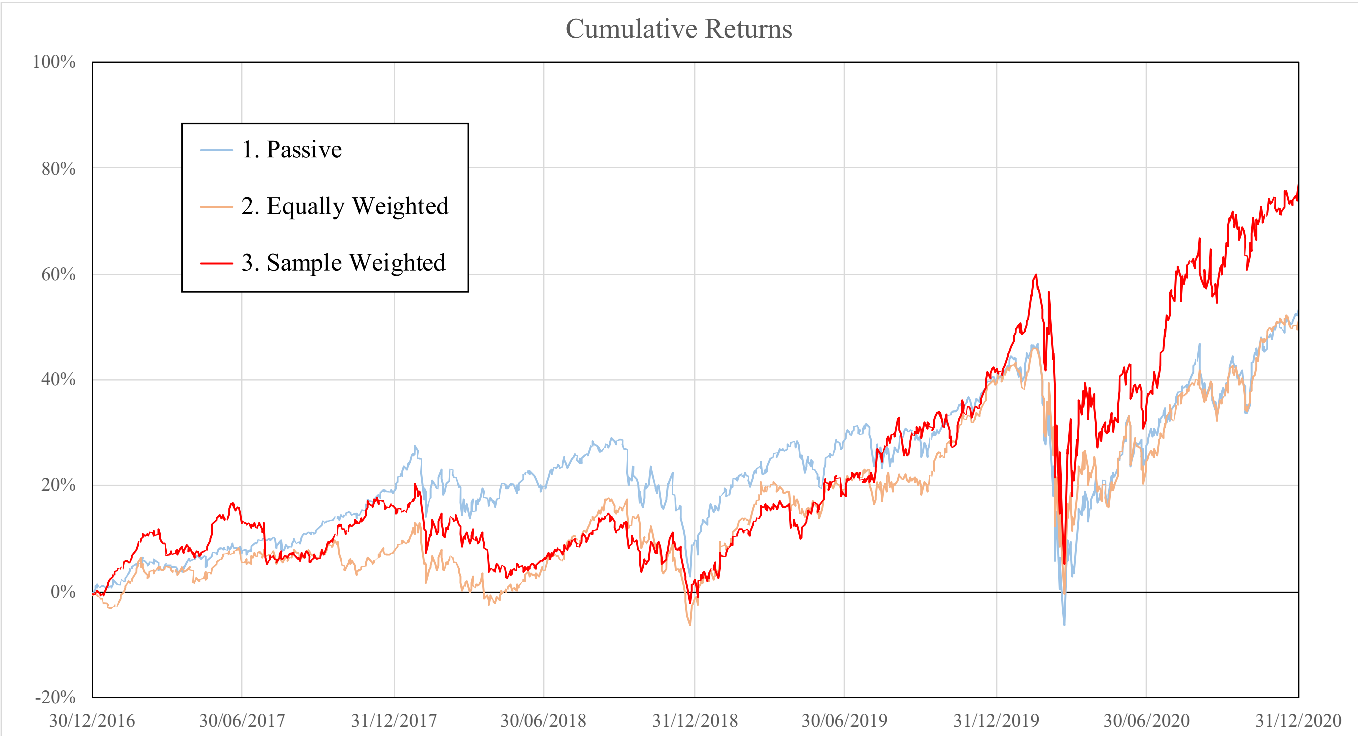


Figure 11 displays how the cumulative returns evolved over the 1,008 trading days for the sample weighted strategy, and the GARCH strategies with order (1,1) and (1,1,1). It also shows a subplot of the daily returns on the S&P 500 (market) for the same time period. We see that the portfolios perform quite similarly in the first six months of 2017, but as time passes and periods of higher volatility emerge all the GARCH strategies outperform the sample weighted strategy.

Figure 12 displays how the cumulative returns evolved over the 1,008 trading days for the sample weighted strategy, and the GARCH strategies with order (2,2), (2,1,2) and (2,2,2). It also shows a subplot of the daily returns of the S&P 500. Figure 12 shows the same pattern as Figure 11, the sample weighted strategy is outperformed by all the GARCH strategies from mid 2017 and onwards.

Figure 11


Cumulative Returns
120%
100%
80%
60%
40%
20%
0%
-20%
3. Sample Weighted
4. GARCH(1,1)(N)
5. GARCH(1,1)(t)
6. GJR-GARCH(1,1,1)(t)
30/12/2016
30/06/2017
31/12/2017
30/06/2018
31/12/2018
30/06/2019
31/12/2019
30/06/2020
31/12/2020

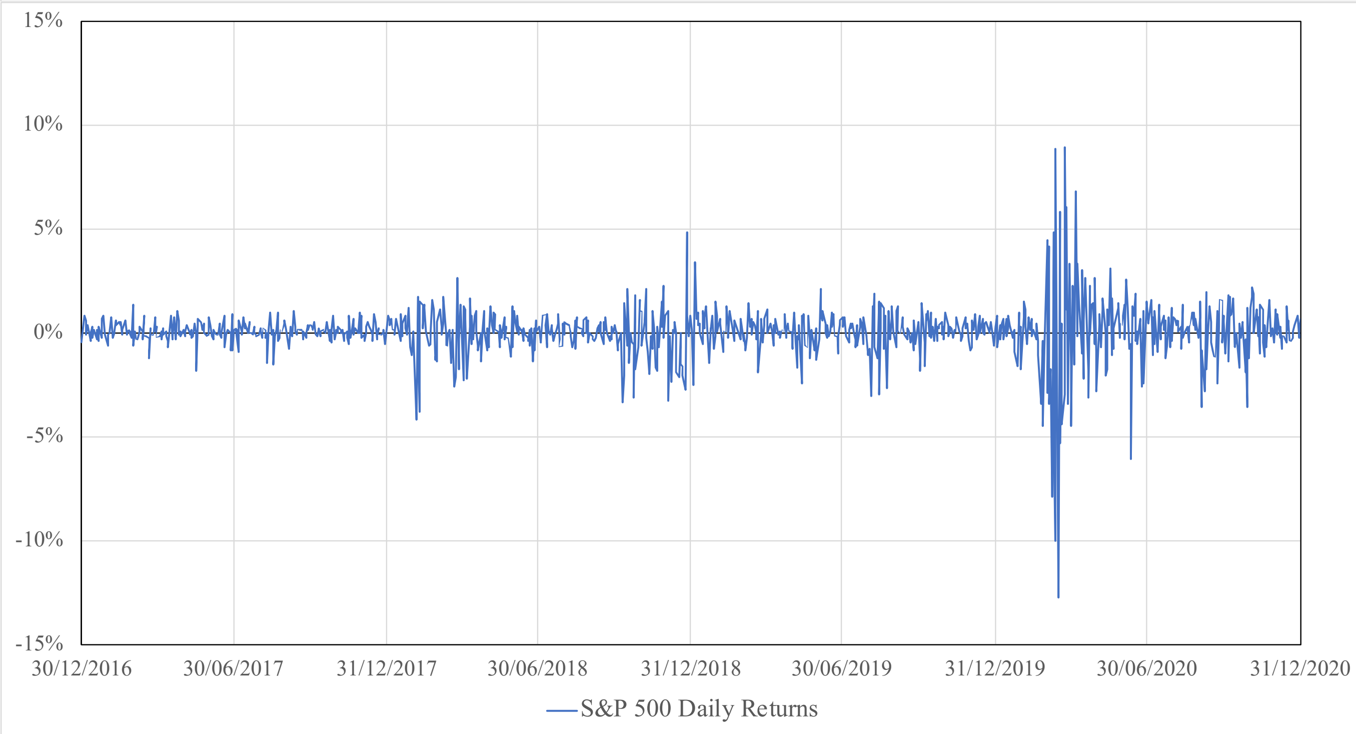
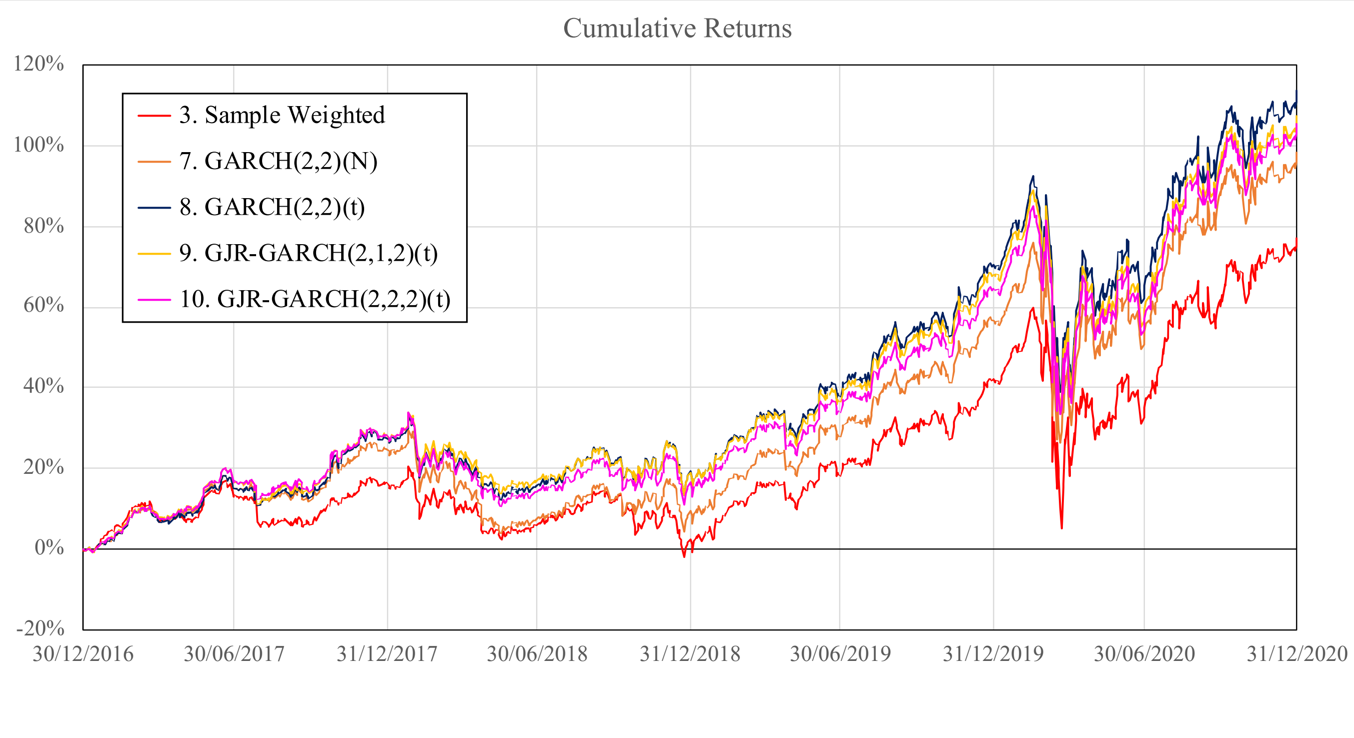
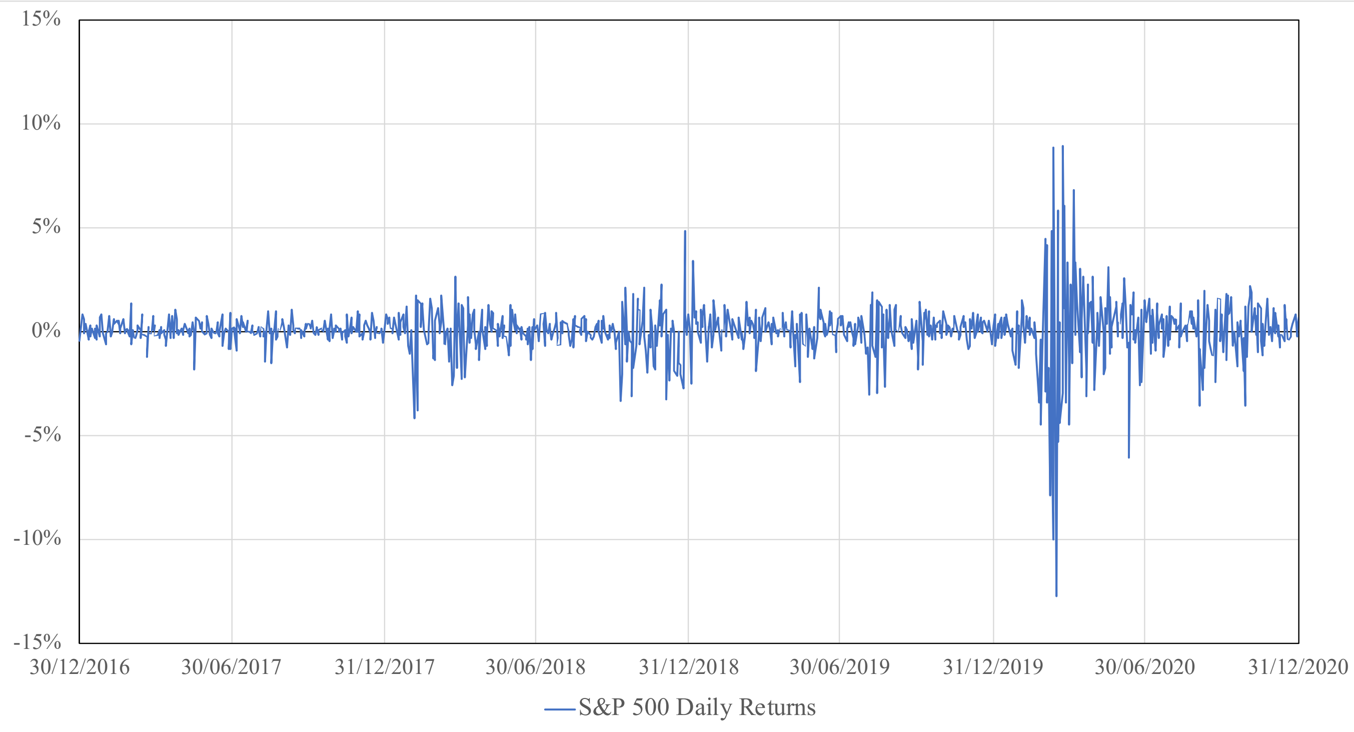

15%
10%
5%
0%
-5%
-10%
-15%
30/12/2016
30/06/2017
31/12/2017
30/06/2018
31/12/2018
30/06/2019
31/12/2019
30/06/2020
31/12/2020
S&P 500 Daily Returns

Figure 12



Cumulative Returns
120%
100%
80%
60%
40%
20%
0%
-20%
30/12/2016
30/06/2017
31/12/2017
30/06/2018
31/12/2018
30/06/2019
31/12/2019
30/06/2020
31/12/2020
3. Sample Weighted
7. GARCH(2,2)(N)
8. GARCH(2,2)(t)
9. GJR-GARCH(2,1,2)(t)
10. GJR-GARCH(2,2,2)(t)



15%
10%
5%
0%
-5%
-10%
-15%
30/12/2016
30/06/2017
31/12/2017
30/06/2018
31/12/2018
30/06/2019
31/12/2019
30/06/2020
31/12/2020
S&P 500 Daily Returns

Figure 13

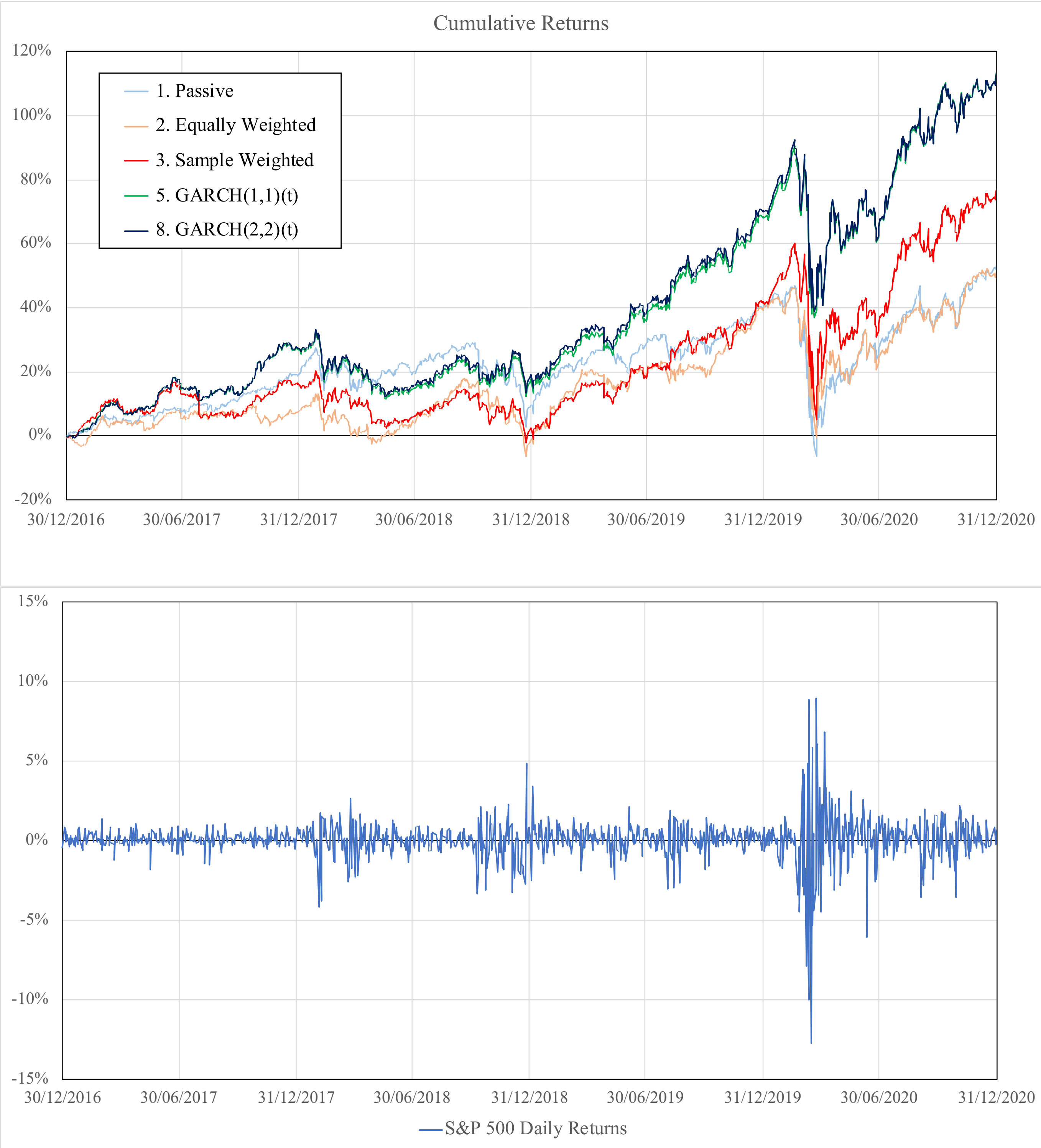

Cumulative Returns
120%
100%
80%
60%
40%
20%
0%
-20%
1. Passive
2. Equally Weighted
3. Sample Weighted
5. GARCH(1,1)(t)
8. GARCH(2,2)(t)
30/12/2016
30/06/2017
31/12/2017
30/06/2018
31/12/2018
30/06/2019
31/12/2019
30/06/2020
31/12/2020
15%
10%
5%
0%
-5%
-10%
-15%
S&P 500 Daily Returns

Figure 13 displays how the cumulative returns evolved over the 1,008 trading days for the passive, equally weighted, sample weighted, and the two GARCH strategies that performed the best, GARCH(1,1)(t) and GARCH(2,2)(t). It also shows a subplot of the daily returns of the S&P 500. The two GARCH strategies performed almost identically with the GARCH(2,2)(t) a little bit ahead before the COVID-19 pandemic broke out in February of 2020, but the GARCH(1,1)(t) catches up during the pandemic and end up a couple of basis points ahead at the end of the year.

As we pointed out in section 3.9, we would like to see if there in any relationship between how accurately a strategy can forecast volatility and the cumulative returns of the strategy. Table 7 and Table 8 displays the RMSE of the variance and the RMSE of the standard deviation for the univariate GARCH models for the individual stocks. We see that different models performed best for different stocks, but the GARCH(2,2)(t) did not perform best for any of the stocks.

What is more interesting to investigate is how the different strategies were able to forecast the volatility. Since we chose the same order for all the stocks in each strategy, the univariate RMSE is not very informative. Instead, we want to look at the RMSE for each strategy and compare this to the cumulative returns and the standard deviation of the given strategy.

Table 9 displays the RMSE of the variance and the RMSE of the standard deviation of the different strategies. As we clearly see the two strategies that performed the best in terms of cumulative returns and standard deviation also has the lowest RMSE for variance and RMSE for standard deviation. The sample weighted strategy performed the worst in terms of cumulative returns and standard deviation and also has the highest RMSE for variance and RMSE for standard deviation.

Figure 14 shows how the estimated volatility for the sample weighted strategy fits with the absolute daily returns of the sample weighted strategy.

Figure 15 shows how the forecasted volatility for the GARCH(2,2)(t) strategy fits with the absolute daily returns of the GARCH(2,2)(t) strategy.

Table 7: RMSE (Variance)

Univariate Model	MMM	VZ	NEE	PM	GE	AMT	TGT	AMD	MO	CCI
GARCH (1,1) (N)	0.000996	0.000398	0.000948	0.001330	0.001856	0.001023	0.001746	**0.003997**	**0.000714**	**0.000837**
GARCH (1,1) (t)	0.000981	0.000397	0.000949	**0.001301**	**0.001852**	0.001029	0.001726	0.004032	0.000732	0.000847
GJR-GARCH (1,1,1) (t)	**0.000977**	0.000394	0.000939	0.001326	0.001861	0.001019	**0.001718**	0.004076	0.000732	0.000840
GARCH (2,2) (N)	0.001003	0.000398	0.000952	0.001360	0.001860	0.001016	0.001819	0.004010	0.000716	0.000840
GARCH (2,2) (t)	0.000984	0.000397	0.000954	0.001314	0.001859	0.001025	0.001728	0.004038	0.000730	0.000850
GJR-GARCH (2,1,2) (t)	0.000979	0.000393	0.000940	0.001350	0.001871	**0.001013**	0.001719	0.004044	0.000730	0.000840
GJR-GARCH (2,2,2) (t)	0.000981	**0.000392**	**0.000938**	0.001351	0.001874	0.001021	0.001726	0.004096	0.000735	0.000844

Table 8: RMSE (Standard Deviation)

Univariate Model	MMM	VZ	NEE	PM	GE	AMT	TGT	AMD	MO	CCI
GARCH (1,1) (N)	**0.01291**	**0.00871**	0.01002	0.01297	0.01955	0.01069	0.01668	**0.03038**	**0.01119**	0.01086
GARCH (1,1) (t)	0.01296	0.00875	0.00990	**0.01285**	0.01956	0.01077	0.01600	0.03065	0.01158	0.01081
GJR-GARCH (1,1,1) (t)	0.01315	0.00873	**0.00979**	0.01308	0.01954	0.01065	**0.01579**	0.03076	0.01164	**0.01073**
GARCH (2,2) (N)	0.01305	0.00872	0.01007	0.01357	0.01952	0.01069	0.01734	**0.03038**	0.01124	0.01086
GARCH (2,2) (t)	0.01298	0.00875	0.00996	0.01295	0.01955	0.01077	0.01604	0.03066	0.01159	0.01084
GJR-GARCH (2,1,2) (t)	0.01314	0.00872	0.00981	0.01319	**0.01949**	**0.01065**	0.01582	0.03076	0.01162	0.01074
GJR-GARCH (2,2,2) (t)	0.01321	**0.00871**	0.00980	0.01320	0.01952	0.01070	0.01588	0.03097	0.01170	0.01075

Table 9: Volatility Forecasts				
Strategy	RMSE (Variance)	RMSE (Standard Deviation)	Standard Deviation	Cumulative Returns
3. Sample Weighted	0.0009586	0.01116	1.4164%	77.04%
4. GARCH (1,1) (N)	0.0008841	0.009161	1.3821%	106.42%
5. GARCH (1,1) (t)	**0.0008776**	**0.009094**	**1.3726%**	**113.89%**
6. GJR-GARCH (1,1,1) (t)	0.0009059	0.009125	1.3817%	103.86%
7. GARCH (2,2) (N)	0.0008807	0.009179	1.3814%	98.45%
8. GARCH (2,2) (t)	**0.0008654**	**0.009070**	**1.3717%**	**113.75%**
9. GJR-GARCH (2,1,2) (t)	0.0009103	0.009154	1.3826%	107.44%
10. GJR-GARCH (2,2,2) (t)	0.0009065	0.009151	1.3841%	105.53%

Figure 14

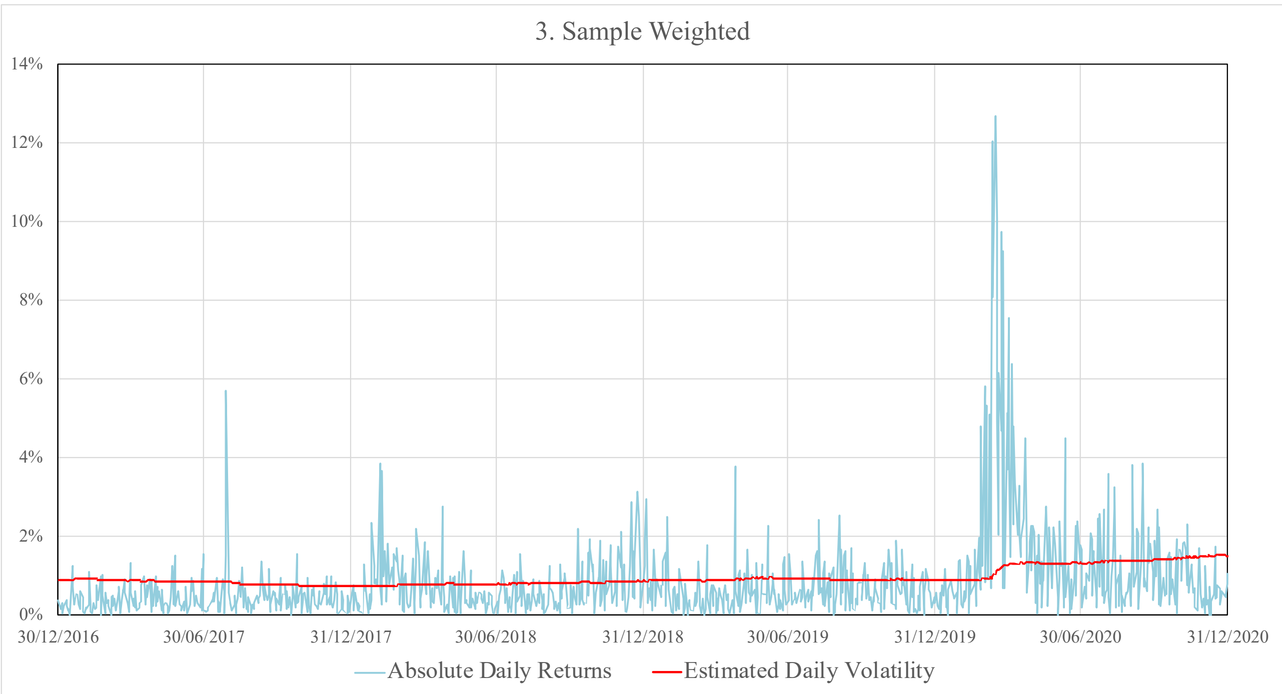
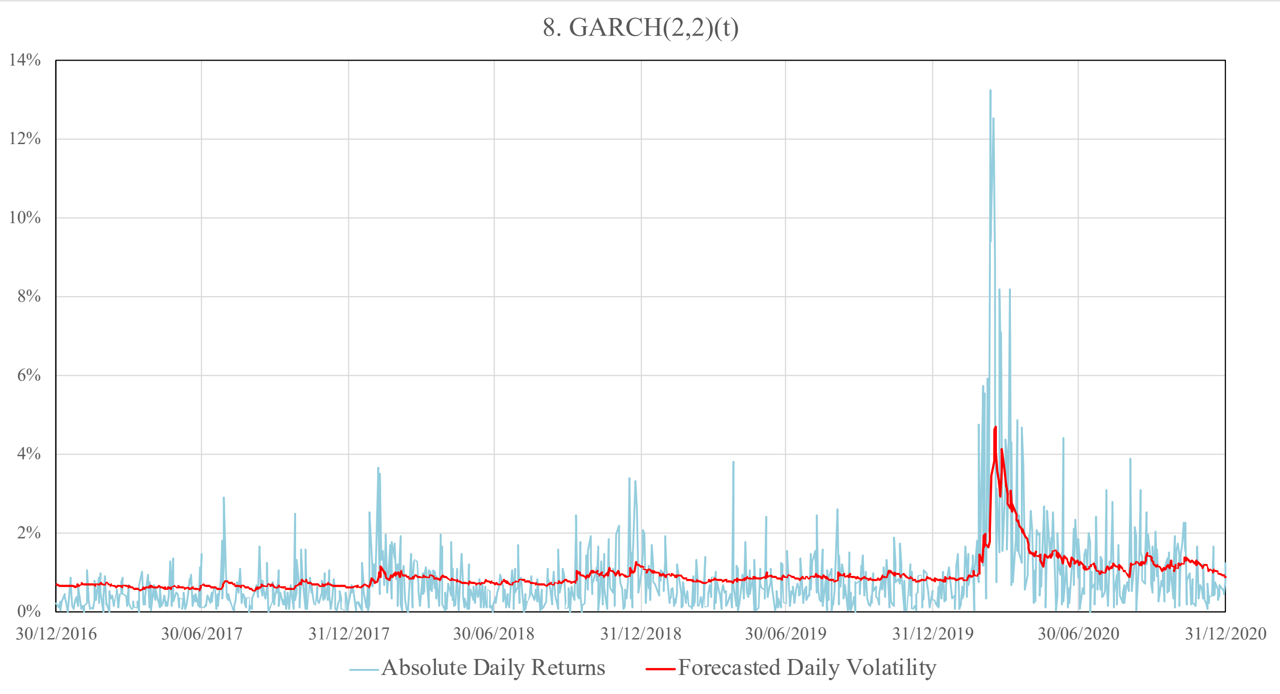


Figure 15



6. Conclusions

The aim with our analysis was to see if we could use stylized facts about financial asset returns to improve volatility forecasts, and if there were any relationship between volatility forecasting and portfolio performance. We compared a passive, equally weighted, and sample weighted strategy with different GARCH model strategies. We saw that all the GARCH model strategies outperformed the passive and equally weighted in terms of cumulative returns, but the passive and equally weighted was less risky than the GARCH strategies. Compared with the sample weighted strategy the GARCH strategies outperformed this strategy both in terms of cumulative returns and risk. Of the GARCH strategies the GARCH(1,1) and GARCH(2,2) with t distributed errors performed the best. When we look at the relationship between volatility forecasting and portfolio performance, we see that the strategies that provided the most accurate volatility forecast also performed best in terms of cumulative returns and risk. We saw that the GARCH strategies with t distributed errors was able to forecast the volatility with more accuracy than the other GARCH strategies. Based on our analysis we have seen that we can improve volatility forecasts using stylized facts about financial asset returns and that there is a relationship between volatility forecasting and portfolio performance.

Appendix A

Parameter Estimation Using Maximum Likelihood

With maximum likelihood estimation the parameter values in the model are chosen by finding the most likely values of the parameters given the actual data. This is done by forming a likelihood function. The likelihood function is a multiplicative function of the actual data and is difficult to maximize, this can be solved by taking its logarithm in order to turn the likelihood function into a log likelihood function, which is an additive function of the data (Stock, Watson 2020, 405-406).

The log-likelihood function is:

$$\sum_{t=1}^{T} l_t$$

Univariate GARCH Normal Distributed Errors

$$\epsilon_t \sim N(0, h_t)$$

$$l_t = -\frac{1}{2}\left\{\ln(2\pi h_t) + \frac{\epsilon_t^2}{h_t}\right\} \tag{35}$$

Univariate GARCH t Distributed Errors

$$\epsilon_t \sim t(\mathrm{df})$$

$$l_t = \ln\Gamma\left(\frac{\mathrm{df}+1}{2}\right) - \ln\Gamma\left(\frac{\mathrm{df}}{2}\right) - \frac{1}{2}\left[\{(\mathrm{df}-2)\pi h_t\} + (\mathrm{df}+1)\ln\left\{1 + \frac{\epsilon_t^2}{(\mathrm{df}-2)h_t}\right\}\right] \tag{36}$$

$$\mathrm{df} > 2$$

CCC GARCH Multivariate Normal Distribution

$$l_t = -0.5m\log(2\pi) - 0.5\log\{\det(\mathbf{R})\} - \log\{\det(\mathbf{D}_t^{1/2})\} - 0.5\tilde{\boldsymbol{\epsilon}}_t\mathbf{R}^{-1}\tilde{\boldsymbol{\epsilon}}_t' \tag{37}$$

$$\tilde{\boldsymbol{\epsilon}}_t' = \mathbf{D}_t^{-1/2}\boldsymbol{\epsilon}_t$$

$$\boldsymbol{\epsilon}_t = \mathbf{r}_t - \mathbf{C}\mathbf{r}_{t-1}$$

CCC GARCH Multivariate t Distribution

$$\begin{aligned} l_t = {} & \log\Gamma\left(\frac{\mathrm{df}+m}{2}\right) - \log\Gamma\left(\frac{\mathrm{df}}{2}\right) - \frac{m}{2}\log\{(\mathrm{df}-2)\pi\} \\ & -0.5\log\{\det(\mathbf{R})\} - \log\{\det(\mathbf{D}_t^{1/2})\} - \frac{\mathrm{df}+m}{2}\log\left(1 + \frac{\tilde{\boldsymbol{\epsilon}}_t\mathbf{R}^{-1}\tilde{\boldsymbol{\epsilon}}_t'}{\mathrm{df}-2}\right) \end{aligned} \tag{38}$$

$$\tilde{\boldsymbol{\epsilon}}_t' = \mathbf{D}_t^{-1/2}\boldsymbol{\epsilon}_t$$

$$\boldsymbol{\epsilon}_t = \mathbf{r}_t - \mathbf{C}\mathbf{r}_{t-1}$$

$$\mathrm{df} > 2$$

Appendix B

```
Python Script:
------------------------------------------------------------------------------------------------------------------------------------

# Import libraries

import pandas as pd
from pandas_datareader import data as web
import numpy as np
from arch import arch_model
import pmdarima as pm
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as scs
import statsmodels.api as sm
import statsmodels.tsa.api as smt
import warnings
warnings.filterwarnings("ignore")

------------------------------------------------------------------------------------------------------------------------------------

# Specify assets and time horizon

market = '^GSPC'
start_m = pd.Timestamp('1990-01-01')
end_m = pd.Timestamp('2021-01-01')

assets = ['MMM', 'VZ', 'NEE', 'PM', 'GE', 'AMT', 'TGT', 'AMD', 'MO', 'CCI']
start = pd.Timestamp('2011-01-01')
end = pd.Timestamp('2021-01-01')

------------------------------------------------------------------------------------------------------------------------------------

# Download data from Yahoo Finance

sp500 = web.DataReader(market, 'yahoo', start_m, end_m)[['Adj Close']]

df = web.DataReader(assets, 'yahoo', start, end)[['Adj Close']]

------------------------------------------------------------------------------------------------------------------------------------

# Calculate returns

sp500_returns = 100 * (np.log(sp500['Adj Close']) - np.log(sp500['Adj Close'].shift(1))).dropna()

returns = 100 * (np.log(df['Adj Close']) - np.log(df['Adj Close'].shift(1))).dropna()

------------------------------------------------------------------------------------------------------------------------------------

# Export returns to Excel

sp500_returns.to_excel(r'/Users/fredrikhellesvik/Desktop/Skole/Oslo Met/4. Vår 2021/Masteroppgave/Volatility Master Thesis/R
                       project/SP 500 Returns.xlsx')

returns.to_excel(r'/Users/fredrikhellesvik/Desktop/Skole/Oslo Met/4. Vår 2021/Masteroppgave/Volatility Master Thesis/R
                 project/Asset Returns.xlsx')

------------------------------------------------------------------------------------------------------------------------------------

# Stylized fact 5: Leverage effect

df = web.DataReader(market, 'yahoo', start_m, end_m)[['Adj Close']]
df.rename(columns={'Adj Close':'adj_close'}, inplace = True)
log_ret = (np.log(sp500['Adj Close']) - np.log(sp500['Adj Close'].shift(1))).dropna()
df['log_ret'] = log_ret

df['moving_std_252'] = df[['log_ret']].rolling(window = 252).std()
df['moving_std_21'] = df[['log_ret']].rolling(window = 21).std()

fig, ax = plt.subplots(3, 1, figsize = (15, 12), sharex = True)

df.adj_close.plot(ax=ax[0])
ax[0].set(title='SP 500 Time Series', ylabel='Price($)')

df.log_ret.plot(ax=ax[1])
ax[1].set(ylabel='Log Returns(%)')

df.moving_std_252.plot(ax=ax[2], color='r', label='Moving Volatility 252d')

df.moving_std_21.plot(ax=ax[2], color='g', label='Moving Volatility 21d')
```

```
ax[2].set(ylabel='Moving Volatility', xlabel='Date')

ax[2].legend()
```

```
# Check for ARMA process, AIC, BIC

train = returns[:-(1008)]
for asset in train.columns:
        model = pm.auto_arima(train[asset], d=0, start_p=1, start_q=1, max_p=4, max_q=4, seasonal=False,
        information_criterion='aic', stepwise=True)
        print('AIC:', asset)
        print(model)
        print('--------------------------------')

train = returns[:-(1008)]
for asset in train.columns:
        model = pm.auto_arima(train[asset], d=0, start_p=1, start_q=1, max_p=4, max_q=4, seasonal=False,
        information_criterion='bic', stepwise=True)
        print('BIC:', asset)
        print(model)
        print('--------------------------------')
```

```
# Find GARCH order from AIC and BIC

train = returns[:-(1008)]
mean = 'Constant'
vol = 'GARCH'
dist = 'Normal'
o = 0
p_order = [1, 2, 3, 4]
q_order = [1, 2, 3, 4]

AIC = [[] for i in range(10)]
BIC = [[] for i in range(10)]

for asset in train.columns:
    for p in p_order:
        for q in q_order:
            model = arch_model(train[asset], mean = mean, vol = vol, p = p, q = q, o = o, dist = dist)
            model = model.fit(update_freq = 0, disp = 'off')
            AIC[train.columns.get_loc(asset)].append(round(model.aic, 2))
            BIC[train.columns.get_loc(asset)].append(round(model.bic, 2))

columns = assets
index = ['GARCH(1,1)', 'GARCH(1,2)', 'GARCH(1,3)', 'GARCH(1,4)',
         'GARCH(2,1)', 'GARCH(2,2)', 'GARCH(2,3)', 'GARCH(2,4)',
         'GARCH(3,1)', 'GARCH(3,2)', 'GARCH(3,3)', 'GARCH(3,4)',
         'GARCH(4,1)', 'GARCH(4,2)', 'GARCH(4,3)', 'GARCH(4,4)']

df_AIC = pd.DataFrame(index = index, columns = columns)
df_BIC = pd.DataFrame(index = index, columns = columns)

for i in range(10):
    df_AIC[columns[i]] = AIC[i]
    df_BIC[columns[i]] = BIC[i]

index = assets

min_AIC = []
for i in index:
    MIN = df_AIC[i][df_AIC[i] == min(df_AIC[i])].index.tolist()
    min_AIC.append(MIN)

Best_AIC = pd.DataFrame(min_AIC, columns = ['    "Best" Model'], index = index)

min_BIC = []
for i in index:
    MIN = df_BIC[i][df_BIC[i] == min(df_BIC[i])].index.tolist()
    min_BIC.append(MIN)

Best_BIC = pd.DataFrame(min_BIC, columns = ['    "Best" Model'], index = index)

print('AIC: GARCH(N)')
print('')
print(Best_BIC)

print('BIC: GARCH(N)')
print('')
print(Best_BIC)
```

```
# Find GJR-GARCH order from AIC and BIC

train = returns[:-(1008)]
mean = 'Constant'
vol = 'GARCH'
dist = 't'

p_order = [1, 2, 3, 4]
o_order = [1, 2, 3, 4]
q_order = [1, 2, 3, 4]

AIC = [[] for i in range(10)]
BIC = [[] for i in range(10)]

for asset in train.columns:
    for p in p_order:
        for o in o_order:
            for q in q_order:
                model = arch_model(train[asset], mean = mean, vol = vol, p = p, q = q, o = o, dist = dist)
                model = model.fit(update_freq = 0, disp = 'off')
                AIC[train.columns.get_loc(asset)].append(round(model.aic, 2))
                BIC[train.columns.get_loc(asset)].append(round(model.bic, 2))

columns = assets
index = ['GARCH(1,1,1)', 'GARCH(1,1,2)', 'GARCH(1,1,3)', 'GARCH(1,1,4)',
         'GARCH(1,2,1)', 'GARCH(1,2,2)', 'GARCH(1,2,3)', 'GARCH(1,2,4)',
         'GARCH(1,3,1)', 'GARCH(1,3,2)', 'GARCH(1,3,3)', 'GARCH(1,3,4)',
         'GARCH(1,4,1)', 'GARCH(1,4,2)', 'GARCH(1,4,3)', 'GARCH(1,4,4)',

         'GARCH(2,1,1)', 'GARCH(2,1,2)', 'GARCH(2,1,3)', 'GARCH(2,1,4)',
         'GARCH(2,2,1)', 'GARCH(2,2,2)', 'GARCH(2,2,3)', 'GARCH(2,2,4)',
         'GARCH(2,3,1)', 'GARCH(2,3,2)', 'GARCH(2,3,3)', 'GARCH(2,3,4)',
         'GARCH(2,4,1)', 'GARCH(2,4,2)', 'GARCH(2,4,3)', 'GARCH(2,4,4)',

         'GARCH(3,1,1)', 'GARCH(3,1,2)', 'GARCH(3,1,3)', 'GARCH(3,1,4)',
         'GARCH(3,2,1)', 'GARCH(3,2,2)', 'GARCH(3,2,3)', 'GARCH(3,2,4)',
         'GARCH(3,3,1)', 'GARCH(3,3,2)', 'GARCH(3,3,3)', 'GARCH(3,3,4)',
         'GARCH(3,4,1)', 'GARCH(3,4,2)', 'GARCH(3,4,3)', 'GARCH(3,4,4)',

         'GARCH(4,1,1)', 'GARCH(4,1,2)', 'GARCH(4,1,3)', 'GARCH(4,1,4)',
         'GARCH(4,2,1)', 'GARCH(4,2,2)', 'GARCH(4,2,3)', 'GARCH(4,2,4)',
         'GARCH(4,3,1)', 'GARCH(4,3,2)', 'GARCH(4,3,3)', 'GARCH(4,3,4)',
         'GARCH(4,4,1)', 'GARCH(4,4,2)', 'GARCH(4,4,3)', 'GARCH(4,4,4)']

df_AIC = pd.DataFrame(index = index, columns = columns)
df_BIC = pd.DataFrame(index = index, columns = columns)

for i in range(10):
    df_AIC[columns[i]] = AIC[i]
    df_BIC[columns[i]] = BIC[i]

index = assets

min_AIC = []
for i in index:
    MIN = df_AIC[i][df_AIC[i] == min(df_AIC[i])].index.tolist()
    min_AIC.append(MIN)

Best_AIC = pd.DataFrame(min_AIC, columns = ['    "Best" Model'], index = index)

min_BIC = []
for i in index:
    MIN = df_BIC[i][df_BIC[i] == min(df_BIC[i])].index.tolist()
    min_BIC.append(MIN)

Best_BIC = pd.DataFrame(min_BIC, columns = ['    "Best" Model'], index = index)

print('AIC: GARCH(N)')
print('')
print(Best_BIC)

print('BIC: GARCH(N)')
print('')
print(Best_BIC)

--------------------------------------------------------------------------------------------------------------------------
```

```
# Calculate CCC Var-CoVar Matrices for 1008 days, rolling forecast

N = len(assets)
test_size = 1008

mean = 'Constant'
vol = 'GARCH'
dist = 'Normal'

p = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
p = pd.DataFrame(p).transpose()
p.columns = assets

o = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
o = pd.DataFrame(o).transpose()
o.columns = assets

q = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
q = pd.DataFrame(q).transpose()
q.columns = assets


Var_forc = pd.DataFrame([[0,0,0,0,0,0,0,0,0,0]], columns = assets)
CCC_mat = []
for i in range(test_size):
    train = returns[i:-(test_size - i)]
    cond_vol = []
    std_resids = []
    models = []
    for asset in train.columns:
        model = arch_model(train[asset], mean = mean, vol = vol, p = int(p[asset]), o = int(o[asset]), q = int(q[asset]), dist =
        dist).fit(update_freq=0, disp='off')
        cond_vol.append(model.conditional_volatility)
        std_resids.append(model.resid / model.conditional_volatility)
        models.append(model)
    cond_vol_df = pd.DataFrame(cond_vol).transpose().set_axis(returns.columns, axis='columns', inplace=False)
    std_resids_df = pd.DataFrame(std_resids).transpose().set_axis(returns.columns, axis='columns', inplace=False)
    R = std_resids_df.transpose().dot(std_resids_df).div(len(std_resids_df))
    diag = []
    D = np.zeros((N, N))
    for model in models:
        diag.append(model.forecast(horizon=1).variance.values[-1][0])
    diag = np.sqrt(np.array(diag))
    np.fill_diagonal(D, diag)
    H = np.matmul(np.matmul(D, R.values), D)
    H = pd.DataFrame(H)
    CCC_mat.append(H)
    var_forc = np.diagonal(H)
    Var_forc_length = len(Var_forc)
    Var_forc.loc[Var_forc_length] = var_forc

------------------------------------------------------------------------------------------------------------------------------------

# Export Variance Forecast to Excel

Var_forc.to_excel(r'/Users/fredrikhellesvik/Desktop/Skole/Oslo Met/4. Vår 2021/Masteroppgave/Volatility Master Thesis/R
                  project/4. Variance Forecast.xlsx', index = True)

------------------------------------------------------------------------------------------------------------------------------------

# Export Var-Covar Matrices Forecast to Excel

names = np.arange(1, 1009).tolist()
names = [str(int) for int in names]

writer = pd.ExcelWriter(r'/Users/fredrikhellesvik/Desktop/Skole/Oslo Met/4. Vår 2021/Masteroppgave/Volatility Master Thesis/R
         project/4. Var-CoVar Matrices.xlsx')
for i, A in enumerate(CCC_mat):
    A.to_excel(writer,sheet_name = "{0}".format(names[i]))

writer.save()

------------------------------------------------------------------------------------------------------------------------------------
```

R Script:

```
---------------------------------------------------------------------------------------------------------------------------

# Load libraries ----

library(xlsx)
library(tseries)
library(FinTS)
library(zoo)
library(rugarch)
library(e1071)
library(rmgarch)
library(quadprog)
library (Rsymphony)
library(Rglpk)
library("writexl")

---------------------------------------------------------------------------------------------------------------------------

# Read in the data ----

sp500 <- read.xlsx("SP 500 Returns.xlsx", sheetIndex = 1)
sp500_price <- read.xlsx("SP 500 Price.xlsx", sheetIndex = 1)

sp500_ret <- sp500$log.ret * 100
sp500_pr <- sp500_price$Adj.Close

returns <- read.xlsx("Asset Returns.xlsx", sheetIndex = 1)

---------------------------------------------------------------------------------------------------------------------------

# Stylized facts about financial asset returns ----

# Fact 1: Non-Normal distribution of returns ----

hist(sp500_ret, freq = F, breaks = 200, main = 'Distribution of SP 500 Returns', xlab = 'Log Returns (%)')
lines(density(sp500_ret), col = "red", cex = 1.5)
lines(seq(-10, 10, by = 0.1), dnorm(seq(-10, 10, by = 0.1), mean(sp500_ret), sd(sp500_ret)), col = "blue", cex = 1.5)
legend(7, 0.5, legend = c("SP 500", "Normal"), col = c("red", "blue"), pch = c("l", "l"), cex = 1, bty = "n")

qqnorm(sp500_ret, pch = 1, frame = FALSE)
qqline(sp500_ret, col = "steelblue", lwd = 2)


# Fact 2: Volatility Clustering ----

plot(sp500$Date, sp500_ret, type = 'l', col = 'black', main = 'SP 500 Daily Returns', xlab = 'Date', ylab = 'Log Returns (%)')


# Fact 3: Absence of autocorrelation in returns ----

plot(acf(sp500_ret, lag.max = 50), main = 'Autocorrelation Returns')


# Fact 4: Small and decreasing autocorrelation in squared and absolute returns ----

sq_ret <- sp500_ret**2
plot(acf(sq_ret, lag.max = 50), main = 'Autocorrelation Squared Returns')

abs_ret <- abs(sp500_ret)
plot(acf(abs_ret, lag.max = 50), main = 'Autocorrelation Absolute Returns')

---------------------------------------------------------------------------------------------------------------------------

# Equally Weighted Portfolio Returns

test <- 1008

pret_EW <- c(1)
port_var_est <- c(1)
port_sd_est <- c(1)
w <- c(0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1)

for(i in 1:test){
  t <- length(returns$MMM) - test + i - 1
  mu <- returns[length(returns$MMM) - test + i, 2:11]
  mu <- as.numeric(as.character(unlist(mu)))
  cov.R <- cov(returns[i:t, 2:11])
  pr_EW <- (t(w) %*% mu)/100
  pvar_est <- (t(w) %*% cov.R %*% w)/10000
  psd_est <- sqrt(pvar_est)
  pret_EW <- c(pret_EW, pr_EW)
  port_var_est <- c(port_var_est, pvar_est)
  port_sd_est <- c(port_sd_est, psd_est)
  print(i)
}
```

```
l <- length(returns$Date)
date <- returns$Date[(l-(test-1)):l]
pret_EW <- pret_EW[2:length(pret_EW)]
port_var_est <- port_var_est[2:length(port_var_est)]
port_sd_est <- port_sd_est[2:length(port_sd_est)]
final <- data.frame(date, pret_EW, port_var_est, port_sd_est)

write_xlsx(final,"/Users/fredrikhellesvik/Desktop/Skole/Oslo Met/4. Vår 2021/Masteroppgave/Volatility Master Thesis/R project//2.
           Equally Weighted.xlsx")

---------------------------------------------------------------------------------------------------------------------------------

# Sample max Sharpe ratio portfolio optimization ----

test <- 1008

pret_SR <- c(1)
port_var_est <- c(1)
port_sd_est <- c(1)
OW_1 <- c(1)
OW_2 <- c(1)
OW_3 <- c(1)
OW_4 <- c(1)
OW_5 <- c(1)
OW_6 <- c(1)
OW_7 <- c(1)
OW_8 <- c(1)
OW_9 <- c(1)
OW_10 <- c(1)

for(i in 1:test){
  t <- length(returns$MMM) - test + i - 1
  mean.R <- apply(returns[i:t, 2:11], 2, mean)
  cov.R <- cov(returns[i:t, 2:11])
  sd.R <- sqrt(diag(cov.R))

  # Max Sharpe ratio

  Amat <- cbind(rep(1, 10), mean.R, diag(1, nrow = 10))
  mu.P <- seq(min(mean.R) + 1e-04, max(mean.R) - 1e-04, length = 100000)
  sigma.P <- mu.P
  weights <- matrix(0, nrow = 100000, ncol = 10)
  for (j in 1:length(mu.P)) {
    bvec <- c(1, mu.P[j], rep(0, 10))
    result <- solve.QP(Dmat = 2 * cov.R, dvec = rep(0, 10), Amat = Amat, bvec = bvec, meq = 2)
    sigma.P[j] <- sqrt(result$value)
    weights[j, ] <- result$solution
  }

  mu.free <- returns$RF[length(returns$RF) - test + i]
  sharpe = (mu.P - mu.free)/sigma.P
  ind = (sharpe == max(sharpe))
  SRw <- round(weights[ind,], digits = 4)
  SRw <- c(SRw)
  mu <- returns[length(returns$MMM) - test + i, 2:11]
  mu <- as.numeric(as.character(unlist(mu)))
  pr_SR <- (t(SRw) %*% mu)/100
  pvar_est <- (t(SRw) %*% cov.R %*% SRw)/10000
  psd_est <- sqrt(pvar_est)
  pret_SR <- c(pret_SR, pr_SR)
  port_var_est <- c(port_var_est, pvar_est)
  port_sd_est <- c(port_sd_est, psd_est)
  OW_1 <- c(OW_1, SRw[1])
  OW_2 <- c(OW_2, SRw[2])
  OW_3 <- c(OW_3, SRw[3])
  OW_4 <- c(OW_4, SRw[4])
  OW_5 <- c(OW_5, SRw[5])
  OW_6 <- c(OW_6, SRw[6])
  OW_7 <- c(OW_7, SRw[7])
  OW_8 <- c(OW_8, SRw[8])
  OW_9 <- c(OW_9, SRw[9])
  OW_10 <- c(OW_10, SRw[10])
  print(i)
}

l <- length(returns$Date)
date <- returns$Date[(l-(test-1)):l]
pret_SR <- pret_SR[2:length(pret_SR)]
port_var_est <- port_var_est[2:length(port_var_est)]
port_sd_est <- port_sd_est[2:length(port_sd_est)]
OW_1 <- OW_1[2:length(OW_1)]
OW_2 <- OW_2[2:length(OW_2)]
OW_3 <- OW_3[2:length(OW_3)]
OW_4 <- OW_4[2:length(OW_4)]
```

```
OW_5 <- OW_5[2:length(OW_5)]
OW_6 <- OW_6[2:length(OW_6)]
OW_7 <- OW_7[2:length(OW_7)]
OW_8 <- OW_8[2:length(OW_8)]
OW_9 <- OW_9[2:length(OW_9)]
OW_10 <- OW_10[2:length(OW_10)]
final <- data.frame(date, pret_SR, port_var_est, port_sd_est, OW_1, OW_2, OW_3, OW_4, OW_5, OW_6, OW_7, OW_8, OW_9, OW_10)

write_xlsx(final,"/Users/fredrikhellesvik/Desktop/Skole/Oslo Met/4. Vår 2021/Masteroppgave/Volatility Master Thesis/R project//3.
          Sample Weighted.xlsx")

------------------------------------------------------------------------------------------------------------------------------------

# CCC GARCH max Sharpe ratio portfolio optimization ----

test <- 1008

pret_SR <- c(1)
port_var_est <- c(1)
port_sd_est <- c(1)
OW_1 <- c(1)
OW_2 <- c(1)
OW_3 <- c(1)
OW_4 <- c(1)
OW_5 <- c(1)
OW_6 <- c(1)
OW_7 <- c(1)
OW_8 <- c(1)
OW_9 <- c(1)
OW_10 <- c(1)


for(i in 1:test){
  H <- read.xlsx("4. Var-CoVar Matrices.xlsx", sheetIndex = i)
  H <- H[1:10,2:11]
  t <- length(returns$MMM) - test + i - 1
  mean.R <- apply(returns[i:t, 2:11], 2, mean)
  cov.R <- matrix(unlist(H), ncol = 10, byrow = T)
  sd.R <- sqrt(diag(cov.R))

  # Max Sharpe ratio

  Amat <- cbind(rep(1, 10), mean.R, diag(1, nrow = 10))
  mu.P <- seq(min(mean.R) + 1e-04, max(mean.R) - 1e-04, length = 100000)
  sigma.P <- mu.P
  weights <- matrix(0, nrow = 100000, ncol = 10)
  for (j in 1:length(mu.P)) {
    bvec <- c(1, mu.P[j], rep(0, 10))
    result <- solve.QP(Dmat = 2 * cov.R, dvec = rep(0, 10), Amat = Amat, bvec = bvec, meq = 2)
    sigma.P[j] <- sqrt(result$value)
    weights[j, ] <- result$solution
  }

  mu.free <- returns$RF[length(returns$RF) - test + i]
  sharpe = (mu.P - mu.free)/sigma.P
  ind = (sharpe == max(sharpe))
  SRw <- round(weights[ind,], digits = 4)
  SRw <- c(SRw)
  mu <- returns[length(returns$MMM) - test + i, 2:11]
  mu <- as.numeric(as.character(unlist(mu)))
  pr_SR <- (t(SRw) %*% mu)/100
  pvar_est <- (t(SRw) %*% cov.R %*% SRw)/10000
  psd_est <- sqrt(pvar_est)
  pret_SR <- c(pret_SR, pr_SR)
  port_var_est <- c(port_var_est, pvar_est)
  port_sd_est <- c(port_sd_est, psd_est)
  OW_1 <- c(OW_1, SRw[1])
  OW_2 <- c(OW_2, SRw[2])
  OW_3 <- c(OW_3, SRw[3])
  OW_4 <- c(OW_4, SRw[4])
  OW_5 <- c(OW_5, SRw[5])
  OW_6 <- c(OW_6, SRw[6])
  OW_7 <- c(OW_7, SRw[7])
  OW_8 <- c(OW_8, SRw[8])
  OW_9 <- c(OW_9, SRw[9])
  OW_10 <- c(OW_10, SRw[10])
  print(i)
}

l <- length(returns$Date)
date <- returns$Date[(l-(test-1)):l]
pret_SR <- pret_SR[2:length(pret_SR)]
port_var_est <- port_var_est[2:length(port_var_est)]
port_sd_est <- port_sd_est[2:length(port_sd_est)]
OW_1 <- OW_1[2:length(OW_1)]
OW_2 <- OW_2[2:length(OW_2)]
OW_3 <- OW_3[2:length(OW_3)]
OW_4 <- OW_4[2:length(OW_4)]
```

```
OW_5 <- OW_5[2:length(OW_5)]
OW_6 <- OW_6[2:length(OW_6)]
OW_7 <- OW_7[2:length(OW_7)]
OW_8 <- OW_8[2:length(OW_8)]
OW_9 <- OW_9[2:length(OW_9)]
OW_10 <- OW_10[2:length(OW_10)]
final <- data.frame(date, pret_SR, port_var_est, port_sd_est, OW_1, OW_2, OW_3, OW_4, OW_5, OW_6, OW_7, OW_8, OW_9, OW_10)

write_xlsx(final,"/Users/fredrikhellesvik/Desktop/Skole/Oslo Met/4. Vår 2021/Masteroppgave/Volatility Master Thesis/R project//4.
          GARCH(1,1)(N).xlsx")

-------------------------------------------------------------------------------------------------------------------------------

# Plot frontier ----

par(mfrow = c(1, 1))
plot(sigma.P, mu.P, xlab="Standard Deviation", ylab="Expected Return", type = "l", xlim = c(0, max(sd.R) * 1.1), ylim =
     c(min(mean.R) * 1.05, max(mean.R) * 1.1), lty = 3, lwd = 2, col = "skyblue")
points(0, mu.free, cex = 5, pch = ".", col = "grey")
sharpe = (mu.P - mu.free)/sigma.P
lines(c(0, 2), mu.free + c(0, 2) * (mu.P[ind] - mu.free)/sigma.P[ind], lwd = 2, lty = 1, col = "grey")
points(sigma.P[ind], mu.P[ind], cex = 4, pch = "*", col = "red")
ind3 = (mu.P > mu.P[ind2])
lines(sigma.P[ind3], mu.P[ind3], type = "l", xlim = c(0, max(sd.R) * 1.1), ylim = c(min(mean.R) * 1.05, max(mean.R) * 1.1), lwd =
      2, col = "blue")
ind = (sharpe == max(sharpe))
options(digits = 3)
ind2 = (sigma.P == min(sigma.P))
points(sigma.P[ind2], mu.P[ind2], cex = 1, pch = "-")
text(sd.R[1], mean.R[1], ".", cex = 4)
text(sd.R[2], mean.R[2], ".", cex = 4)
text(sd.R[3], mean.R[3], ".", cex = 4)
text(sd.R[4], mean.R[4], ".", cex = 4)
text(sd.R[5], mean.R[5], ".", cex = 4)
text(sd.R[6], mean.R[6], ".", cex = 4)
text(sd.R[7], mean.R[7], ".", cex = 4)
text(sd.R[8], mean.R[8], ".", cex = 4)
text(sd.R[9], mean.R[9], ".", cex = 4)
text(sd.R[10], mean.R[10], ".", cex = 4)
legend(0.025, 0.0008, legend=c("Optimal CAL", "Efficient Frontier", "Optimal Portfolio"), col=c("grey", "blue", "red"), pch =
       c("l", "l", "*"), cex = 1, pt.cex = 2, bty = "n")

-------------------------------------------------------------------------------------------------------------------------------
```

References

Advanced Micro Devices Stock Price. 2021. Retrieved from: https://finance.yahoo.com/quote/AMD/history?p=AMD

Altria Group Stock Price. 2021. Retrieved from: https://finance.yahoo.com/quote/MO/history?p=MO

American Tower Stock Price. 2021. Retrieved from: https://finance.yahoo.com/quote/AMT/history?p=AMT

Benninga, Simon. 2014. Financial Modelling. 4th ed. London: The MIT Press.

Bodie, Zvi, Alex Kane, Alan J. Marcus. 2013. *Essentials of Investments*. 9th ed. New York: McGraw-Hill/Irwin.

Bollerslev, Tim. 1986. «Generalized autoregressive conditional heteroskedasticity» *Journal of Econometrics* 31(3):307-327. DOI: 10.1016/0304-4076(86)90063-1.

Bollerslev, Tim. 1990. «Modelling the Coherence in Short-Run Nominal Exchange Rates: A Multivariate Generalized Arch Model» *The Review of Economics and Statistics* 72(3):498-505. DOI: 10.2307/2109358.

Brooks, Chris. 2014. *Introductory Econometrics for Finance*. 3rd ed. Cambridge: Cambridge University Press.

Cont, Rama. 2001. «Empirical properties of asset returns: stylized facts and statistical issues» *Quantitative Finance* 1(2):223-236. DOI: 10.1080/713665670.

Crown Castle International Stock Price. 2021. Retrieved from: https://finance.yahoo.com/quote/CCI/history?p=CCI

Engle, Robert F. 1982. «Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of United Kingdom Inflation» *Econometrica* 50(4):987-1007. DOI: 10.2307/1912773.

Engle, Robert. 2002. «Dynamic Conditional Correlation: A Simple Class of Multivariate Generalized Autoregressive Conditional Heteroskedasticity Models» *Journal of Business & Economic Statistics* 20(3):339-350. DOI: 10.1198/073500102288618487.

Franke, Jürgen, Wolfgang K. Härdle, Christian M. Hafner. 2008. *Statistics of Financial Markets*. 2nd ed. Berlin: Springer-Verlag Berlin Heidelberg.

General Electric Stock Price. 2021. Retrieved from: https://finance.yahoo.com/quote/GE/history?p=GE

Ghalanos, Alexios, Tobias Kley. 2020. *Package 'rugarch'*. Version 1.4-4. Retrieved from: https://cran.r-project.org/web/packages/rugarch/rugarch.pdf

Ghalanos, Alexios. 2019. *Package 'rmgarch'*. Version 1.3-7. Retrieved from: https://cran.r-project.org/web/packages/rmgarch/rmgarch.pdf

Glosten, Lawrence R., Ravi Jagannathan, David E. Runkle. 1993. «On the Relation between the Expected Value and the Volatility of the Nominal Excess Returns on Stocks» *The Journal of Finance* 48(5):1779-1801. DOI: 10.1111/j.1540-6261.1993.tb05128.x.

Hansen, A. Lunde. 2005. «A Forecast Comparison of volatility models: Does anything beat a GARCH(1,1)» *Journal of Applied Economics* 20(7):873-889. DOI: 10.1002/jae.800

Hull, John C. 2018. *Risk management and financial institutions*. 5th ed. Hoboken: John Wiley & Sons.

Javed, Farrukh, Panagiotis Mantalos. 2013. «GARCH-Type Models and Performance of Information Criteria» *Communications in Statistics – Simulation and Computation* 42(8):1917-1933. DOI: 10.1080/03610918.2012.683924.

Lewinson, Eryk. 2020. *Python for Finance Cookbook*. Birmingham: Packt Publishing Ltd.

Markowitz, Harry. 1952. «Portfolio Selection» *The Journal of Finance* 7(1):77-91. DOI:10.1111/j.1540-6261.1952.tb01525.x

Phillip Morris International Stock Price. 2021. Retrieved from: https://finance.yahoo.com/quote/PM/history?p=PM

Pfaff, Bernhard. 2013. *Financial Risk Modelling and Portfolio Optimization with R*. Chichester: John Wiley & Sons, Ltd.

Raychaudhuri, Samik. 2008. «Introduction to Monte Carlo Simulation» *2008 Winter Simulation Conference* 1(1):91-100. DOI: 10.1109/WSC.2008.4736059.

Rojasa, I., O. Valenzuelab, F. Rojasa, A. Guillena, L.J. Herreraa, H. Pomaresa, L. Marquezb, M. Pasadasb. 2008. «Soft-computing techniques and ARMA model for time series prediction» Neurocomputing 71(4-6):519-537. DOI: 10.1016/j.neucom.2007.07.018.

Sahamkhadam, Maziar, Andreas Stephan, Ralf Östermark. 2018. «Portfolio optimization based on GARCH-EVT-Copula forecasting models» *International Journal of Forecasting* 34(3):497-506. DOI: 10.1016/j.ijforecast.2018.02.004.

Sharpe, William F. 1966. «Mutual Fund Performance» *The Journal of Business* 39(1): 119-138. DOI: 10.1086/294846.

StataCorp. 2019. *STATA TIME-SERIES REFERENCE MANUAL*. Release 16. Texas: StataCorp LLC.

Stock, James H., Mark W. Watson. 2020. *Introduction to Econometrics*. 4th ed. United Kingdom: Pearson Education.

S&P 500 Index Price. 2021. Retrieved from: https://finance.yahoo.com/quote/%5EGSPC/history?p=%5EGSPC

Target Stock Price. 2021. Retrieved from:
https://finance.yahoo.com/quote/TGT/history?p=TGT

Tashman, Leonard J. 2000. «Out-of-sample test of forecasting accuracy: an analysis and review» *International Journal of Forecasting* 16(4):437-450.
DOI: 10.1016/S0169-2070(00)00065-0.

Verizon Communications Stock Price. 2021. Retrieved from:
https://finance.yahoo.com/quote/VZ/history?p=VZ

3M Stock Price. 2021. Retrieved from:
https://finance.yahoo.com/quote/MMM/history?p=MMM

3-Month Treasury Bill: Secondary Market Rate. 2021. Retrieved from:
https://fred.stlouisfed.org/series/DTB3#0