Modelling and Forecasting of daily spot prices per barrel of *West Texas Intermediate* Crude Oil

GARCH MODELLING AND RISK MEASURES

Rehareha RAMANANA | M5MF10 Quantitative Risk Management | November 30th 2018

PREFACE

The language that I used and the results that will be exposed in this report, are from 2 notebooks available on my GitHub: https://github.com/big21ray/QRM. I think it's important to say that the notebooks were only a way for me to compute results, but are not properly clean in themselves, I rarely commented and some of the cases are useless. I used two notebooks, because one of them was becoming too large, and I was losing time just loading the notebook. The Python version is 3.65. I decided to do the assignment by myself, first because my attendance rate is not very high meaning that I did not ask anyone to do it with me, and also because from my previous experience, teamwork is important, but I also prefer reasoning and doing the entirety of the project to seize all the meaning behind it.

In Python, I decided to use the following packages: Numpy, Pandas, Arch, Scipy, Seaborne, Matplotlib, Tabulate and Statsmodel.

PART A : STYLISED FACTS AND GARCH MODELLING

Before answering to the questions in the assignment, I first computed the daily log returns. And here is an idea of the first 10 values in the files, with WTI being the price of *West Texas Intermediate* crude oil, 'Daily Log Returns' being the daily log returns of the WTI prices, both of them starting from the November 5th 2008 to November 5th 2018. We have overall, 2519 entries.

	Date	WTI	Daily Log Returns
0	2008-11-05	65.41	NaN
1	2008-11-06	60.72	-0.074402
2	2008-11-07	61.06	0.005584
3	2008-11-10	62.19	0.018337
4	2008-11-11	59.38	-0.046237
5	2008-11-12	55.95	-0.059499
6	2008-11-13	58.31	0.041315
7	2008-11-14	57.18	-0.019569
8	2008-11-17	55.14	-0.036329
9	2008-11-18	54.42	-0.013144

Fig 1: First 10 values available in our dataset

We notice a NaN value in the first line, but we know that this will not be a problem in the future, as we can easily drop it for future computations.

To have an idea of what we are working with, here is a graph showing the evolution of WTI during the period available:

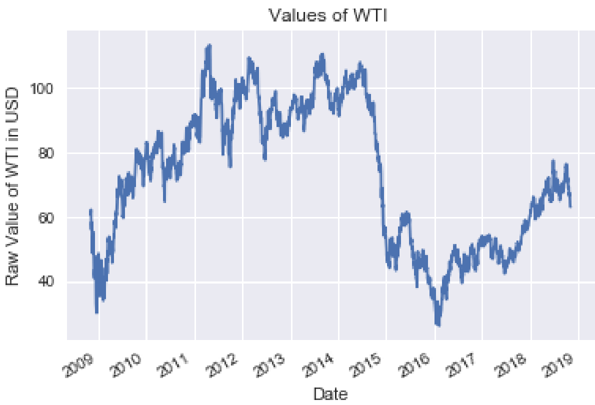

Fig 2: Evolution of the price of WTI from 2008 to 2018

(i) *Compute the mean, standard deviation, skewness, and kurtosis of the log returns. Draw also a histogram including a kernel density estimate and a normal distribution(density) fitted to the log returns. What can you conclude ?*

Here are the values that I computed on Python directly with the Pandas package that give the possibility to directly compute the needed values, knowing that I already dropped the NaN value:

Mean	Standard Deviation	Skewness	Kurtosis
- 1.41 10^{-5}	2.36 10^{-2}	1.63 10^{-1}	4.38

Following are the histograms with a kernel density estimate and then with a normal distribution fitted to the log returns. Note: I plotted normed histogram, and I used Bins of size 50 (empirical choice).

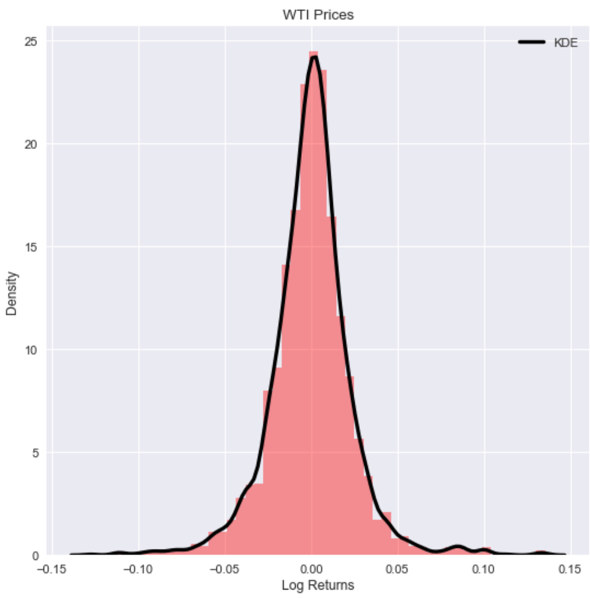

Fig 3 : Histogram of the Log Returns with Kernel Density Estimate(KDE)

And here is the histogram with the normal distribution:

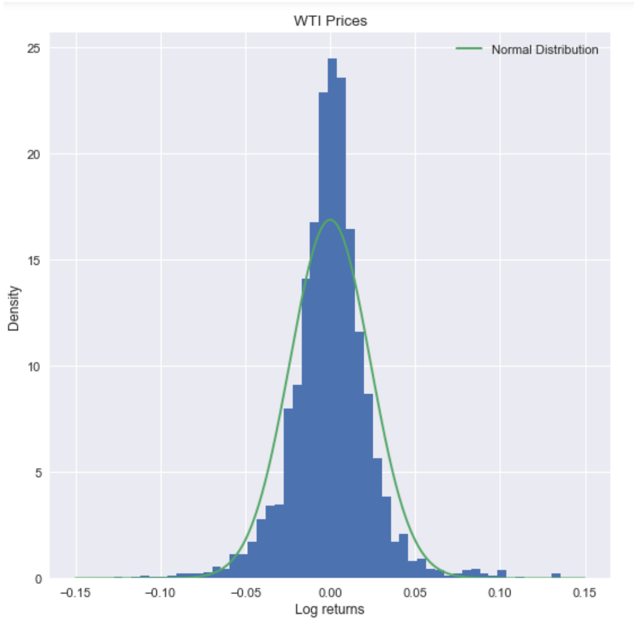

Fig 4 : Histogram of the Log Returns with a Normal Distribution fitted to the Log Returns

Conclusions:

- Similarly to the examples we saw in the courses, the KDE marry well the form of the histogram and on the other side, the Normal Distribution doesn't fit well with the histograms, there is a huge discrepancy on peak part , and the tail is not following correctly the histogram.
- The positivity of the skewness indicates that the right tail is longer and heavier than the left tail.
- The kurtosis indicates that the distribution, is leptokurtic with an acute peak. It is easy to witness on the histograms the acuteness of the peak.

(ii) Draw the empirical ACFs of the log returns, their absolute values, and their squares. What can you conclude ?

Using the ACF function from the StatsModel package, I get the following graph :

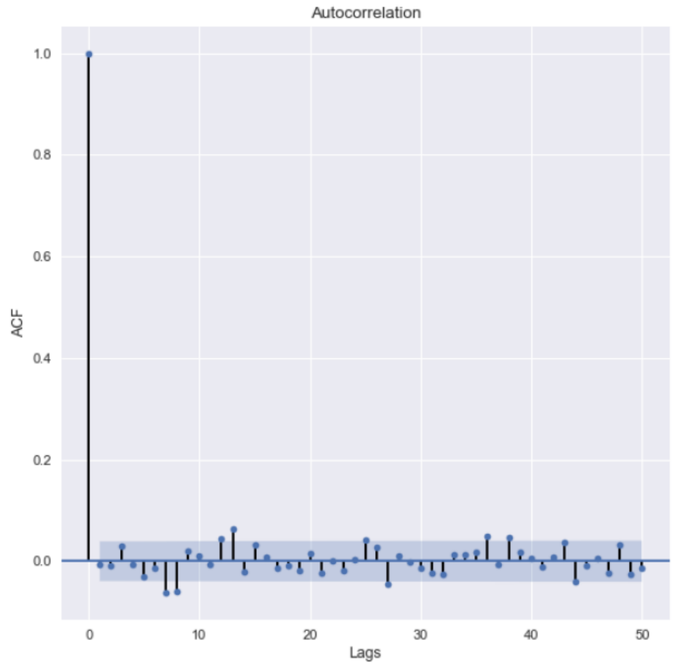

Fig 5 : ACF on the first 50 lags for the WTI Log Returns

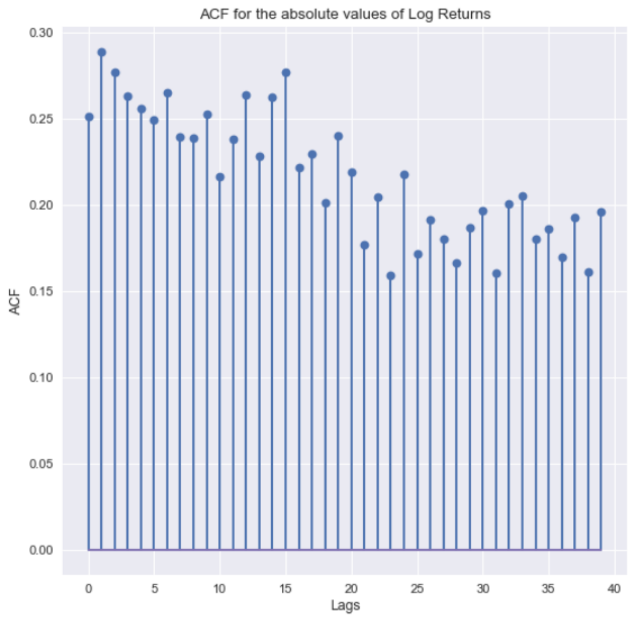

Fig 6 : ACF for the 40 lags of the WTI Absolute Values of Log Returns with the lag 0 removed

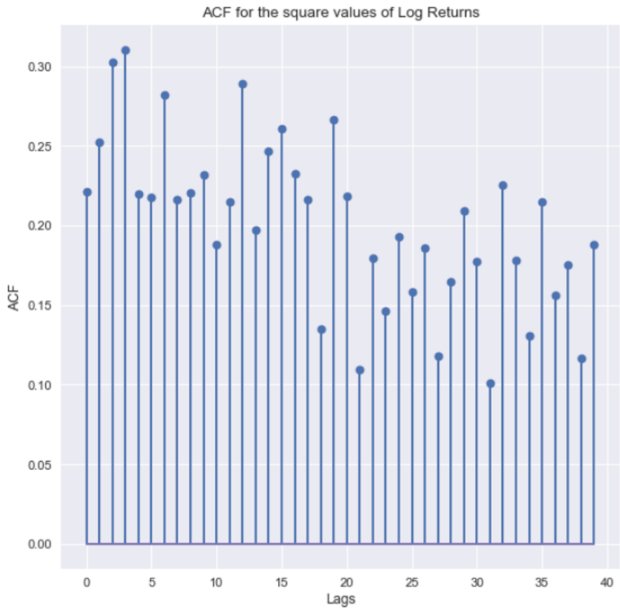

Fig 7 : ACF for the 40 lags with WTI Square Values of Log Returns with the lag 0 removed

Conclusions:

- The first thing I noticed, was the fact that as said in the lectures, the ACF is unreadable with lags close to the number of entries, I did not put the graph to not waste spaces, but the actual graph is very noisy.
- The ACF are very close to 0, so we can say, that the data(log returns) are serially uncorrelated. There are no arbitrage opportunity.
- However, when we look at ACF on absolute values and square values, we witness a slight and slow decay as the lag is increasing, it means that the volatility is persistent, as the ACF is pretty significant(~0.25)

(iii) Fit a standard GARCH(1,1) model, with constant conditional mean and standard normal innovations, to the log return data. Report the estimated parameter values and their standard errors. Plot the absolute returns and fitted volatilites.

With the Arch package on Python, I can specify a GARCH(1,1) model with constant mean and standard distribution. After modelling the log returns, I get the following parameters values:

Parameters Values	Standard Error
$\alpha_0 = 1.160\ 10^{-5}$	$5.202\ 10^{-12}$
$\alpha_1 = 0.1$	$1.797\ 10^{-2}$
$\beta_1 = 0.88$	$1.459\ 10^{-2}$

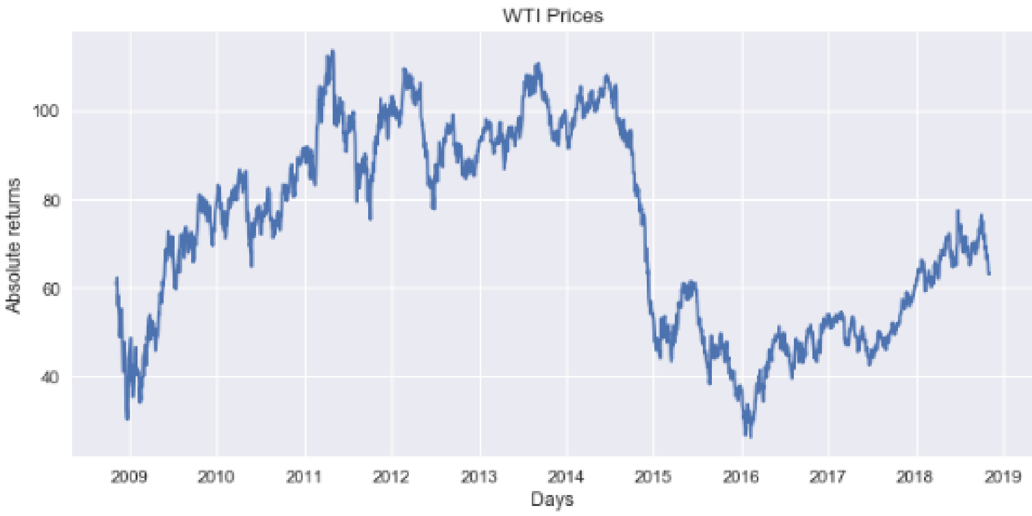

Fig 8 : Absolute Returns over the years per day

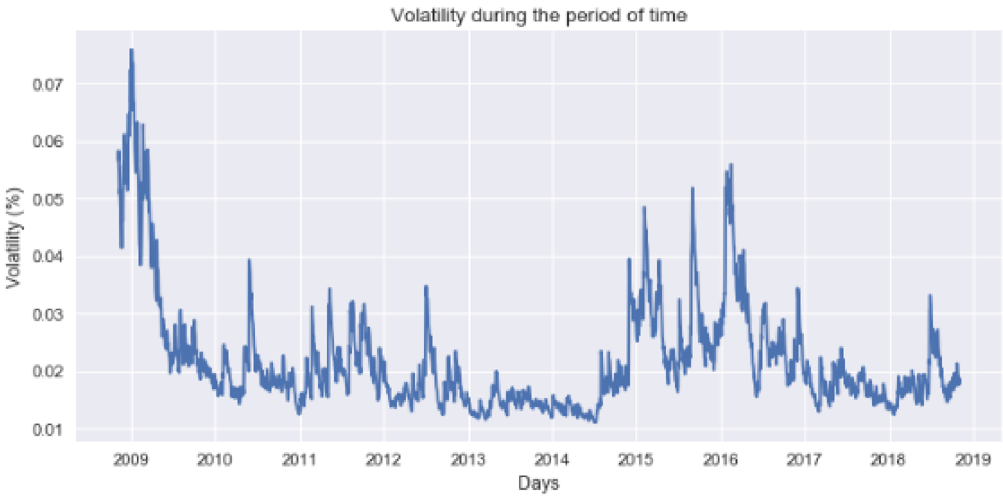

Fig 9 : Conditional Volatility from the GARCH(1,1) model fitted to the log returns over the years per day

(iv) Assess the goodness of fit of the model estimated in (iii). Do the standardized residuals look like and iid sample from the standard normal distribution?

A very quick and naïve answer would be to say that the model estimated in (iii) seems to fit very well because the conditional volatility seems to describe well the original returns. But to make sure, we are going to plot the ACF of the GARCH(1,1) standardized residuals and make sure it's iid. In Python, I took the residuals of my GARCH(1,1) model, divided it by the conditional volatility, squared it and plotted the ACF, here is the graph :

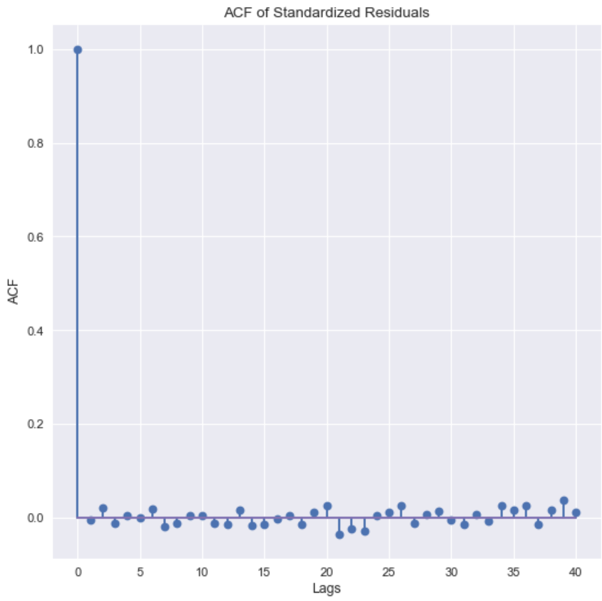

Fig 10 : ACF of the Standardized Residuals from GARCH(1,1) model

The conclusion from this graph is that the standardized residuals are actually iid as the ACF values are very close all-around, zero.

Note: I also plotted in my Python notebook the ACF of the squared values of non-standardized residuals and compared it to the ACF of the squared values of log returns, and they are, very, very, very similar. But I am not sure how to use this information, it can maybe validate that this model seems reasonable.

To follow-up, I decided to plot the histogram of standardized residuals with a fitted normal distribution, and a Quantile-Quantile Plot compared to the normal distribution:

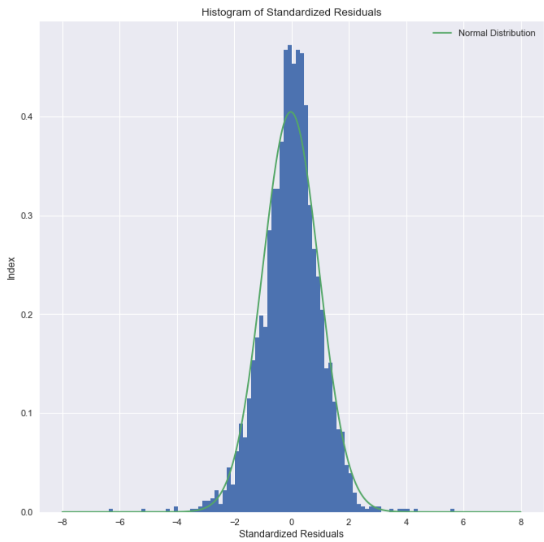

Fig 11 : Histogram of Standardized Residuals with fitted standard normal distribution

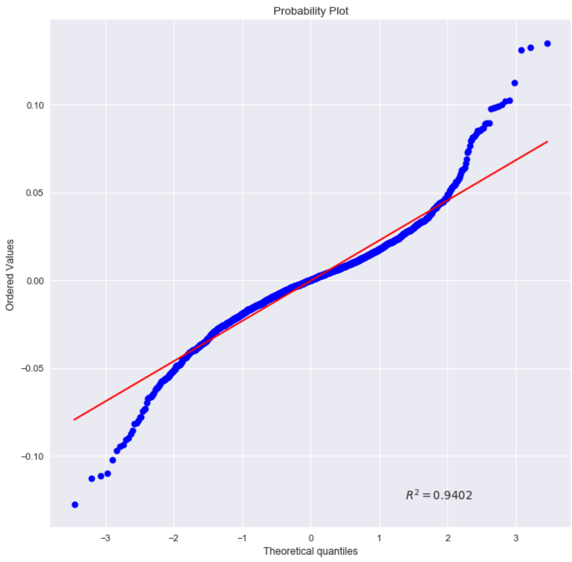

Fig 12 : QQPlot of the residuals compared to a Normal Distribution

So the conclusion seems to be that the model fits very well. The normal distribution does marry the form of the histogram perfectly but it's efficient. The QQPlot gives us $R^2 > 0.9$ so it seems to be an good model.

(v) Fit an ARMA(1,1)-GARCH(1,1) model with Student t-distributed innovations. Does the new specification improve the fit, compared to the model in (iii) ?

Fitting such a model in Python, is not evident, and on a technical side, I could not choose between, fitting an ARMA(1,1) model then fitting a GARCH(1,1) model, and fitting an ARIMA(1,1,1) model then fitting a GARCH(1,1). In both cases, the values of the AIC and BIC are inferior to the one in the model in (iii), so I would say that this model is an improvement.

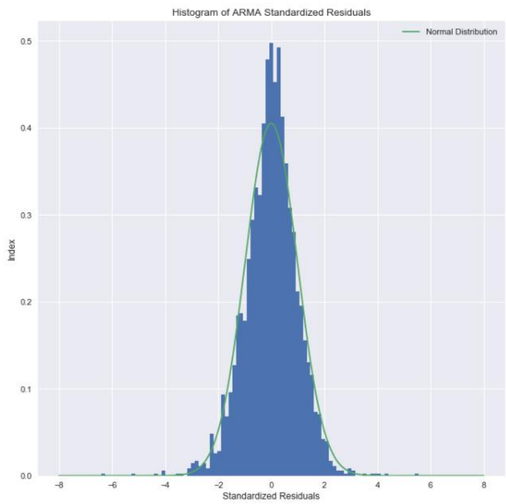

Fig 13 : Histogram with fitted normal distribution

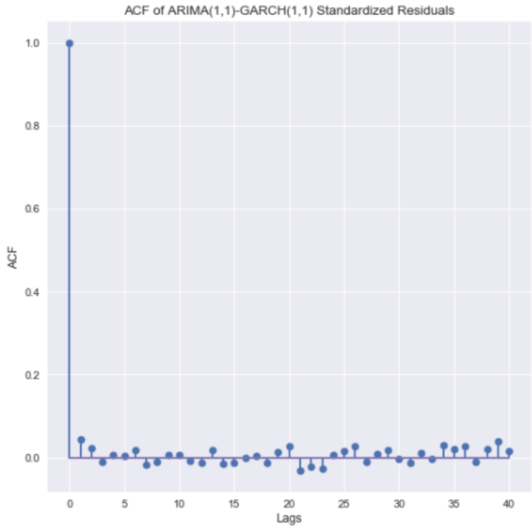

Fig 14 : ACF plot of the Standardized Residuals of ARIMA(1,1,1)-GARCH(1,1)

So it seems like the ARIMA(1,1,1)-GARCH(1,1) model seems to be a slight improvement compared to the previous model, the ACF shows that the standardized residuals are iid, and the normal distribution seems to marry more the form than previous model.

PART B : Risk Measures

This time we work with a new dataset, we work on the *ProShares Ultra S&P500*. We have overall 3022 entries. Here how it looks like:

Fig 15 : Evolution of SSO over the period of time

I then computed the opposite of usual log returns. We are going to forecast starting from November 5th 2008 which corresponds to the 503rd entry.

(i) *Historical Simulation(HS) – use a rolling window scheme, whereby for each forecast you use the data over the past 500 trading days to compute the HS forecasts of VaR and ES.*

As asked, I am going to plot, the forecast Value at Risk (VaR) and Expected-Shortfall(ES) from the 99% confidence interval and 95% interval.

Fig 16 : Historical Simulation with 99% confidence interval

Fig 17 : Historical Simulation with 95 % confidence interval

(ii) *Filtered historical simulation (FHS) with exponentially-weighted moving average (EWMA) — generate first volatility forecasts for the entire time series by applying the EWMA scheme with $\alpha = 0.06$ and $\hat{\mu}t = 0$ for all t. the formulae in Part 3, slide 102 to compute the VaR and ES forecasts. Determine the distribution FZ by applying historical simulation to the standardised EWMA residuals, within a rolling window over the past 500 trading days.*

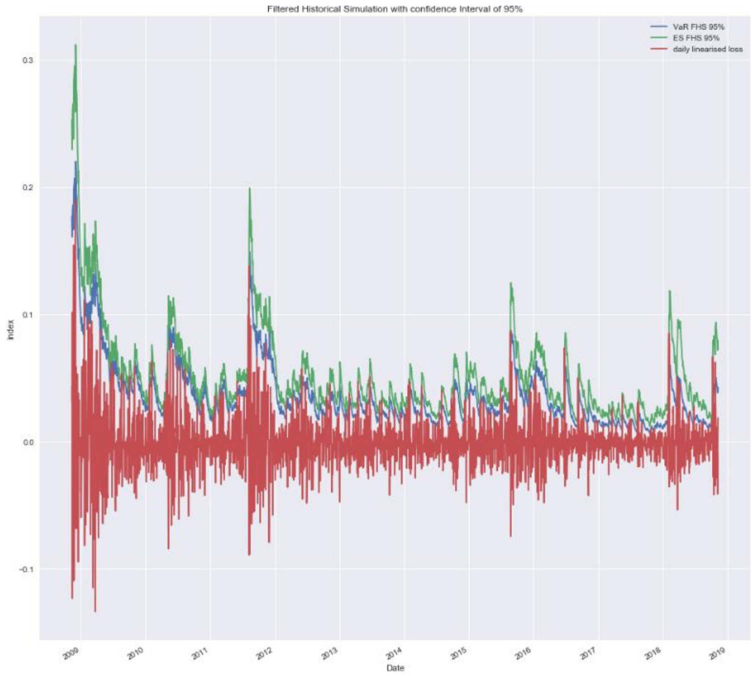

Fig 18 : Filtered Historical Simulation with confidence interval of 95%

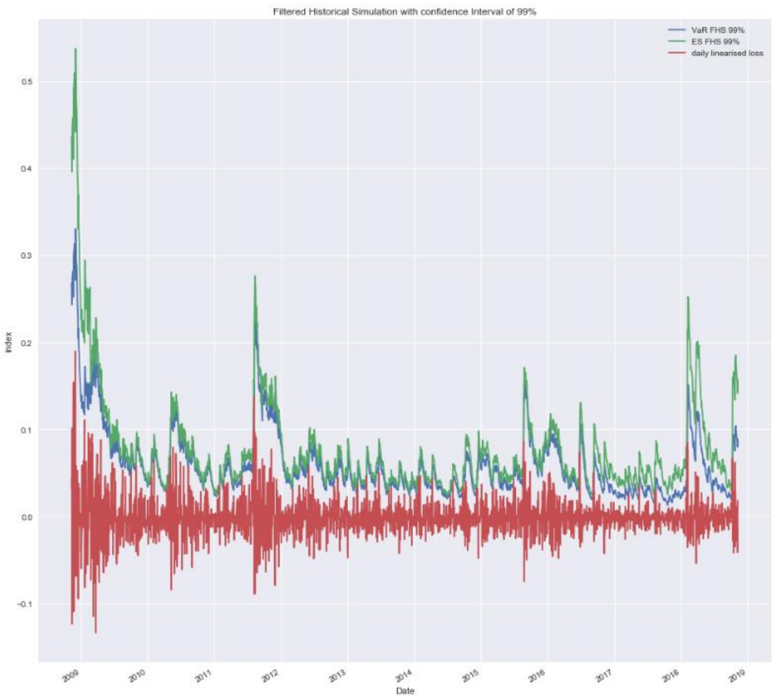

Fig 19 : Filtered Historical Simulation with confidence interval of 99%

(iii) *FHS with GARCH — use a rolling window scheme, whereby you estimate a GARCH(1, 1) model, with constant conditional mean and standard normal innovations, every day using the data over the past 500 trading days. Determine the distribution FZ by applying historical simulation to the standardised GARCH residuals and use then the formulae and the one-day-ahead GARCH volatility forecast to compute the VaR and ES forecasts.*

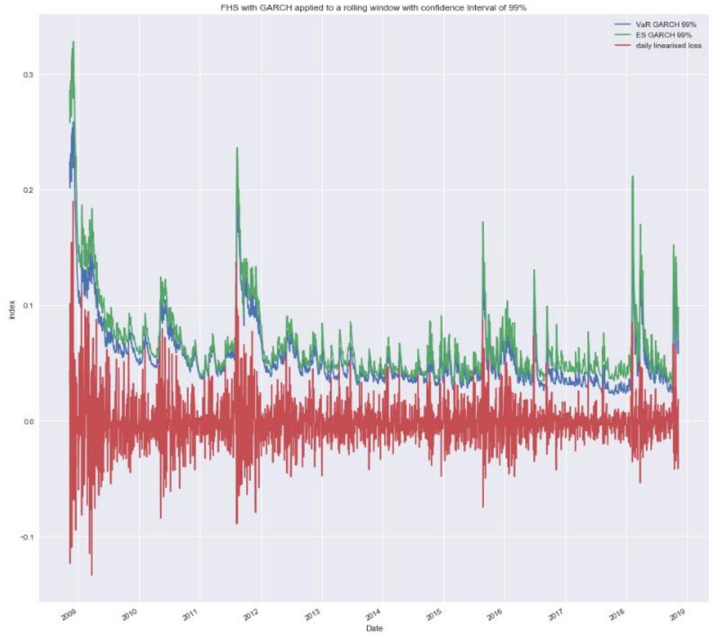

Fig 20 : GARCH Historical Simulation with confidence interval of 99%

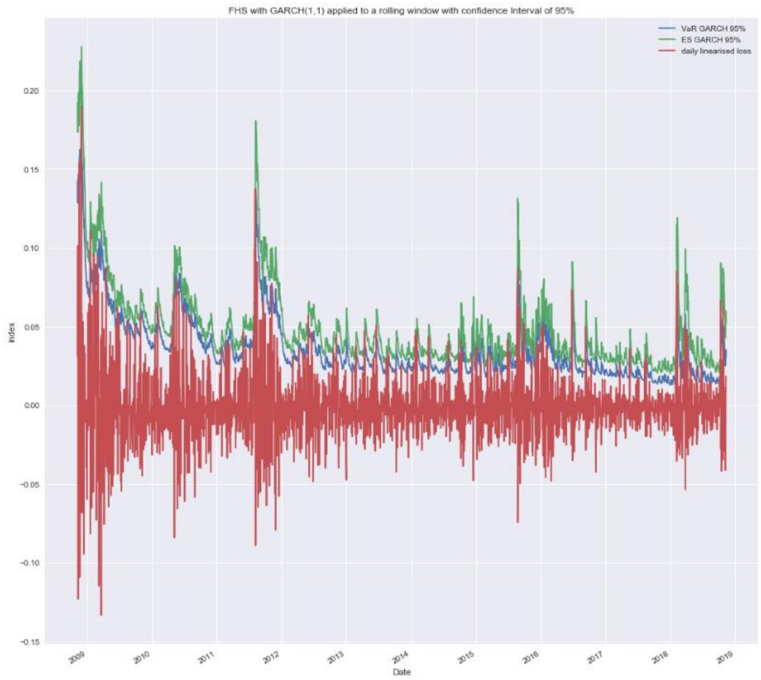

Fig 21 : GARCH Historical Simulation with confidence interval of 95%

PART C : Back testing results from PART B

Here are the results from the unconditional coverage test for VaR :

MODEL	ALPHA	LIKELIHOOD RATIO	P-VALUE
HS	0.99	0.4263	0.5138
HS	0.95	3.156	< 0.0756
FHS	0.99	0.1981	0.6563
FHS	0.95	0.1372	0.7167
GARCH FHS	0.99	0.4263	0.5138
GARCH FHS	0.95	3.507	<0.0610

Here are the results for the joint coverage-independence tests:

MODEL	ALPHA	LIKELIHOOD RATIO	P-VALUE
HS	0.99	12.05	< 0.002
HS	0.95	17.56	< 0.0001
FHS	0.99	5.920	0.052
FHS	0.95	0.1299	0.7185
GARCH FHS	0.99	6.487	0.0109
GARCH FHS	0.95	3.545	0.170

Here are the results for back testing ES:

MODEL	ALPHA	VALUE OF TEST STATISTIC	P-VALUE
HS	0.99	0.3418	< 0.3662
HS	0.95	0.6680	0.2520
FHS	0.99	0.2550	0.3993
FHS	0.95	0.5084	< 0.3056
GARCH FHS	0.99	0.2225	0.0109
GARCH FHS	0.95	0.4708	0.3189

Conclusions:

- For the joint coverage and independence tests, only the FHS model and the GARCH FHS model at 95% seems to be acceptable as the p-value of those models are over 0.1 and better than the HS models. On the graphs we can see that the HS models are in term of efficiency very far away from the ones from FHS and GARCH FHS. So for HS model, we reject the hypothesis that there is no clustering.
- For the back testing of ES, as we saw that the VaR testing was not relevant for the HS model, and somehow the GARCH FHS model is also not perfect as the p-value is not that high for $\alpha = 0.95$ and $\alpha = 0.99$. However, for the FHS model, we saw that only with $\alpha = 0.95$, the p-value of the joint coverage and independence test gave something acceptable, and after back testing ES, we see that the p-value gives us something relevant. We see different p-values acceptable for HS/GARCH/FHS (in green in the table), however as this test is kind of dependent on the VaR back test, we cannot accept those models, as said before.

I advice Pietro to take the

FHS MODEL AT 95 %