Group 6

Total: 10/10

M5MS11 - Introduction to Statistical Finance; Coursework 1

Jerome Wynne, Christopher Stanton, Phillip Murray

Part A: Derivatives Pricing

Consider the trinomial market model with parameter values:

$$\mu = 0.02, \quad h = 0.14, \quad r = 0.01, \quad T = 8 \tag{1}$$

Assume also that $p_+ = \mathbb{P}[\xi_1 = 1]$ *and* $p_- = \mathbb{P}[\xi_1 = -1]$ *satisfy* $p_+ + p_- < 1$.

Part A (i) Find an EMM $\mathbb{Q}$ *by specifying suitable* $q_+ := \mathbb{Q}[\xi_1 = 1]$ *and* $q_- := \mathbb{Q}[\xi_1 = -1]$.

Set $U = e^{\mu+h}, M = e^{\mu}, D = e^{\mu-h}, R = e^{r}$.

From Example 2.30 in the lecture notes we have the following necessary and sufficient condition for the probability measure $\mathbb{Q}$ with parameters q_+, q_- to be an equivalent martingale measure within a Trinomial Model (with $p_+ + p_- < 1$):

$$\mathbb{Q} \text{ is an EMM } \iff q_+\frac{U}{R} + q_-\frac{D}{R} + (1 - q_+ - q_-)\frac{M}{R} = 1 \tag{2}$$

Where $q_+, q_- > 0, q_+ + q_- < 1$. For a solution to Equation 2 to exist, we require that $D < R < U$. We now assume that q_+ is fixed and solve for q_-:

$$q_-\frac{M-D}{R} = q_+\frac{U-M}{R} + \frac{M-R}{R} \tag{3}$$

$$\implies \qquad q_- = q_+\frac{U-M}{M-D} + \frac{M-R}{M-D} \tag{4}$$

We restrict the solution so that the constraints $q_+, q_- > 0,\ q_+ + q_- < 1$ are satisfied:

$$0 < q_- \tag{5}$$

$$\Rightarrow \qquad 0 < q_+\frac{U-M}{M-D} + \frac{M-R}{M-D} \tag{6}$$

$$\Rightarrow \quad \frac{R-M}{U-M} < q_+ \tag{7}$$

And:

$$q_- + q_+ < 1 \tag{8}$$

$$\Rightarrow \quad q_+\frac{U-D}{M-D} + \frac{M-R}{M-D} < 1 \tag{9}$$

$$\Rightarrow \qquad q_+\frac{U-D}{M-D} < \frac{R-D}{M-D} \tag{10}$$

$$\Rightarrow \qquad q_+ < \frac{R-D}{U-D} \tag{11}$$

We conclude that an equivalent martingale measure $\mathbb{Q}$ is defined by any pair (q_+, q_-) that satisfies the constraints:

$$\max(0, \frac{R-M}{U-M}) < q_+ < \frac{R-D}{U-D}, \tag{12}$$

$$q_- = q_+\frac{U-M}{M-D} + \frac{M-R}{M-D} \tag{13}$$

In the specific case given in this question ($U = e^{0.16}, M = e^{0.02}, D = e^{-0.12}, R = e^{0.01}$), we obtain the constraint $0 < q_+ < 0.4296\ldots$. An example EMM can be extracted by setting $q_+ = 0.1$ such that $q_- = 0.1912\ldots$.

ok

Part A (ii) Are there any other EMMs?

It is clear from the Part A (i) that for this particular trinomial model there exist uncountably many choices of q_+ within the constraints, each defining a distinct equivalent martingale measure. There is therefore an uncountable set of EMMs, where each measure in the set is parameterized by a (q_+, q_-) pair that satisfies Equations 12 and 13. As an example, another measure in this set is given by $q_+ = 0.2$ and $q_- = 0.306\ldots$. ok

Part A (iii) Work out an expression for $f(x), x > 0$, given the plot of $f(S_t)$ in Figure 1.

Below is a piecewise expression for the payoff $f(x) : x > 0$:

$$f(x) = \begin{cases} x & 0 < x \leq K_1 \\ K_1 & K_1 < x \leq K_2 \\ x - (K_2 - K_1) & K_2 < x \leq K_3 \\ K_1 + (K_3 - K_2) & K_3 < x \end{cases} \tag{14}$$

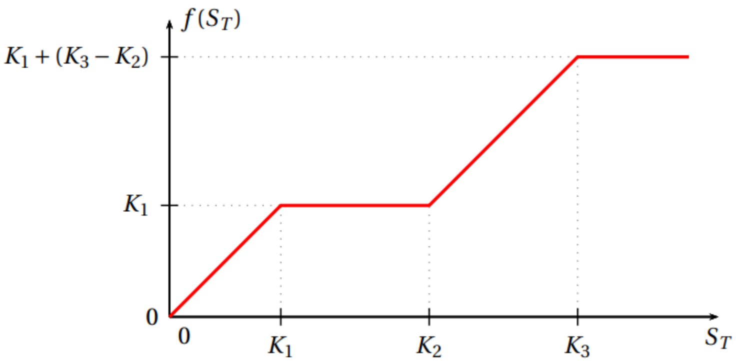

Figure 1: Payoff $f(S_T)$.

Take an EMM $\mathbb{Q}$ from part (i). Set

$$K_1 = 3, \quad K_2 = 7, \quad K_3 = 10. \tag{15}$$

Part A (iv) Using your answer to part (iii), find an arbitrage-free price process $(\Pi_t)_{t=0}^T$ for the payoff $f(S_T)$ in the trinomial market model. Present all values of the price process in the relevant trinomial tree.

Since we showed in part (ii) that the EMM $\mathbb{Q}$ is not unique, the second fundamental theorem of asset pricing implies the market is not complete, meaning that a given payoff X is not necessarily attainable. We therefore resort to risk-neutral pricing. A supplementary Excel workbook is included with this report, 'Options Prices.xlsx', in which all possible paths of the underlying stock price is calculated. Equation 14 and and the recursive expression for the discounted price process of a payoff that is a function of only the final price S_T (Equation 2.48 in the lecture notes) is used to calculate an arbitrage-free price process for $f(S_T)$. All values of the price process are presented in the trinomial tree of Figure 2. For reference, the tree's implementation can be found in cells P23:X39 of the sheet 'New Derivative Price' within the workbook.

Trinomial Tree of the Price Process of f(S_T)								
								6.000
							5.940	6.000
						5.881	5.940	6.000
					5.727	5.843	5.940	6.000
				5.149	5.323	5.517	5.738	6.000
Correct			4.231	4.346	4.473	4.613	4.766	4.930
		3.452	3.498	3.546	3.593	3.643	3.697	3.764
	3.016	3.036	3.053	3.065	3.070	3.066	3.046	3.000
2.831	2.852	2.873	2.892	2.911	2.929	2.948	2.970	3.000
	2.804	2.831	2.858	2.885	2.912	2.941	2.970	3.000
		2.816	2.849	2.880	2.911	2.941	2.970	3.000
			2.822	2.863	2.902	2.938	2.970	3.000
				2.790	2.846	2.901	2.954	3.000
					2.672	2.742	2.821	2.914
						2.430	2.483	2.533
							2.159	2.202
								1.914

Figure 2: Trinomial tree of the price process of $f(S_T)$. Reference: P23:X39, Sheet 'New Derivative Price'.

> *Part A (v) Find a way to express $f(S_T)$ as a combination of positions (long or short) in the underlying and European put and call options. Price those instruments separately and verify that the total value of the trade agrees with your answer to part (iv).*

The following equation expresses the payoff $f(S_T)$ in terms of the underlying, S_T, and three European call options at strike prices $K_1, K_2,$ and K_3:

$$f(S_T) = S_T - (S_T - K_1)^+ + (S_T - K_2)^+ - (S_T - K_3)^+ \tag{16}$$

Equation 16 states that the payoff of $f(S_T)$ can be attained by purchasing 1 underlying, purchasing 1 European call option with strike K_2, and selling 2 European call options: one with strike K_1, and one with strike K_3. Figures 3 - 7[1] present a trinomial tree for the possible price paths of the underlying stock alongside trinomial price models for European call options with strike prices of K_1, K_2 and K_3. These prices are then used to calculate the combined price process of the portfolio as given by Equation 16. The calculations used to produce these trinomial trees can be seen in the workbook: references to each tree within the workbook are given in the figure captions.

[1] Displayed at the end of this section.

Stock								17.983
							15.324	15.634
						13.058	13.322	13.591
					11.128	11.352	11.582	11.816
				9.482	9.674	9.869	10.069	10.272
			8.080	8.244	8.410	8.580	8.753	8.930
		6.886	7.025	7.167	7.311	7.459	7.610	7.764
	5.868	5.986	6.107	6.230	6.356	6.485	6.616	6.749
5.000	5.101	5.204	5.309	5.416	5.526	5.637	5.751	5.868
	4.435	4.524	4.616	4.709	4.804	4.901	5.000	5.101
		3.933	4.013	4.094	4.176	4.261	4.347	4.435
			3.488	3.559	3.631	3.704	3.779	3.855
				3.094	3.156	3.220	3.285	3.352
					2.744	2.799	2.856	2.914
						2.434	2.483	2.533
							2.159	2.202
								1.914

Figure 3: Trinomial tree of the possible price paths of S_t with starting price $S_0 = 5$. Reference: P4:X20, Sheet 'New Derivative Price'.

Trinomial Tree of the Price of a European Call Option, with strike K_1								*14.983*
							12.354	*12.634*
						10.118	*10.352*	*10.591*
					8.216	*8.412*	*8.612*	*8.816*
				6.600	*6.763*	*6.929*	*7.099*	*7.272*
			5.227	*5.361*	*5.499*	*5.639*	*5.783*	*5.930*
		4.060	*4.171*	*4.284*	*4.400*	*4.519*	*4.640*	*4.764*
	3.070	*3.161*	*3.253*	*3.348*	*3.445*	*3.544*	*3.645*	*3.749*
2.231	*2.304*	*2.379*	*2.456*	*2.534*	*2.615*	*2.697*	*2.781*	*2.868*
	1.640	*1.700*	*1.762*	*1.827*	*1.893*	*1.960*	*2.030*	*2.101*
		1.117	*1.164*	*1.214*	*1.266*	*1.320*	*1.377*	*1.435*
			0.666	*0.696*	*0.729*	*0.767*	*0.809*	*0.855*
				0.304	*0.311*	*0.319*	*0.331*	*0.352*
					0.072	*0.057*	*0.035*	*0.000*
						0.003	*0.000*	*0.000*
							0.000	*0.000*
								0.000

Figure 4: Trinomial tree of the price process of a European call option with strike $K_1 = 3$. Reference: P23:X39, Sheet 'Call and Put Options K_1'.

Trinomial Tree of the Price of a European Call Option, with strike K_2								*10.983*
							8.394	*8.634*
						6.197	*6.392*	*6.591*
					4.335	*4.491*	*4.651*	*4.816*
				2.763	*2.883*	*3.008*	*3.138*	*3.272*
			1.503	*1.569*	*1.644*	*1.728*	*1.823*	*1.930*
		0.653	*0.665*	*0.677*	*0.689*	*0.705*	*0.727*	*0.764*
	0.224	*0.214*	*0.201*	*0.183*	*0.159*	*0.125*	*0.076*	*0.000*
0.063	*0.056*	*0.048*	*0.038*	*0.028*	*0.018*	*0.007*	*0.000*	*0.000*
	0.010	*0.007*	*0.004*	*0.002*	*0.001*	*0.000*	*0.000*	*0.000*
		0.001	*0.000*	*0.000*	*0.000*	*0.000*	*0.000*	*0.000*
			0.000	*0.000*	*0.000*	*0.000*	*0.000*	*0.000*
				0.000	*0.000*	*0.000*	*0.000*	*0.000*
					0.000	*0.000*	*0.000*	*0.000*
						0.000	*0.000*	*0.000*
							0.000	*0.000*
								0.000

Figure 5: Trinomial tree of the price process of a European call option with strike $K_2 = 7$. Reference: P23:X39, Sheet 'Call and Put Options K_2'.

Trinomial Tree of the Price of a European Call Option, with strike K_3								7.983
							5.424	5.634
						3.256	3.422	3.591
					1.519	1.589	1.681	1.816
				0.496	0.471	0.432	0.371	0.272
			0.126	0.106	0.082	0.056	0.027	0.000
		0.027	0.020	0.013	0.007	0.003	0.000	0.000
	0.005	0.003	0.002	0.001	0.000	0.000	0.000	0.000
0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		0.000	0.000	0.000	0.000	0.000	0.000	0.000
			0.000	0.000	0.000	0.000	0.000	0.000
				0.000	0.000	0.000	0.000	0.000
					0.000	0.000	0.000	0.000
						0.000	0.000	0.000
							0.000	0.000
								0.000

Figure 6: Trinomial tree of the price process of a European call option with strike $K_3 = 10$. Reference: P23:X39, Sheet 'Call and Put Options K_3'.

Price Process of: Stock - Call(K_1) + Call(K_2) - Call(K_3)								6.000
							5.940	6.000
						5.881	5.940	6.000
					5.727	5.843	5.940	6.000
				5.149	5.323	5.517	5.738	6.000
			4.231	4.346	4.473	4.613	4.766	4.930
Correct		3.452	3.498	3.546	3.593	3.643	3.697	3.764
	3.016	3.036	3.053	3.065	3.070	3.066	3.046	3.000
2.831	2.852	2.873	2.892	2.911	2.929	2.948	2.970	3.000
	2.804	2.831	2.858	2.885	2.912	2.941	2.970	3.000
		2.816	2.849	2.880	2.911	2.941	2.970	3.000
			2.822	2.863	2.902	2.938	2.970	3.000
				2.790	2.846	2.901	2.954	3.000
					2.672	2.742	2.821	2.914
						2.430	2.483	2.533
							2.159	2.202
								1.914

Figure 7: Trinomial tree of the price process of a long position in 1 underlying stock and 1 European call option with strike $K_2 = 7$ and a short position in 2 European call options with strikes $K_1 = 3$ and $K_3 = 10$. Reference: E23:M39, Sheet 'New Derivative Price'. We can see that this process is consistent with the process obtained by computing the price recursively (i.e. the process shown in Figure 2).

Part A: 5/5 marks

Part B: Stylized Facts

> *Part B (i) Compute first log returns for each of the three price series. Analyze whether each return series exhibits the stylized facts (SF_1-SF_3) introduced in Section 4.1 of the lecture notes.*

We are given three datasets - Apple Inc. stock prices, GBP/USD exchange rate and the gold fixing price, with each data set spanning from January 2000 to January 2019. We first compute log returns for each of these time series, that is, given prices $s_0, s_1, ..., s_T$ we calculate $r_t = \log s_t - \log s_{t-1}$ for $t = 1, \ldots, T$ and then multiply by 100 to convert to a percentage.

```
AAPL_data   <- read.csv("AAPL.csv")
gold_data   <- read.csv("Gold.csv")
GBPUSD_data <- read.csv("GBPUSD.csv")

logreturns  <- function(x) diff(log(x))*100 # in %

AAPL_returns   <- logreturns(AAPL_data[ , 2])
gold_returns   <- logreturns(gold_data[ , 2])
GBPUSD_returns <- logreturns(GBPUSD_data[ , 2])
```

We now want to assess whether each time series exhibits the following three stylised facts:

SF1 The unconditional distribution of returns (at a daily or shorter time scale) is markedly non-normal with heavy tails and possibly mild asymmetry.

SF2 Returns are serially uncorrelated.

SF3 Volatility is clustered and persistent.

Stylized Fact 1

To check the first stylised fact we need to calculate the kurtosis κ and skewness β of each time series, and compare these values to the normal values where $\kappa = 3$ and $\beta = 0$. In particular, we are looking for kurtosis values greater than three and nonzero skewness, in order to meet SF1. To do this we use the sample kurtosis:

$$\hat{k} = \frac{\frac{1}{T}\sum_{t=1}^{T}(r_t - \bar{r})^4}{(\frac{1}{T}\sum_{t=1}^{T}(r_t - \bar{r})^2)^2} \tag{17}$$

Which can be implemented in R as follows:

```
Kurtosis <- function(rt, r=mean(rt)) mean((rt - r)^4) / mean((rt - r)^2)^2
```

We also estimate the skewness using the sample skewness, given by

$$\hat{b} = \frac{\frac{1}{T}\sum_{t=1}^{T}(r_t - \bar{r})^3}{(\frac{1}{T}\sum_{t=1}^{T}(r_t - \bar{r})^2)^{3/2}} \tag{18}$$

This can be written in R as follows:

```
Skewness <- function(rt, r=mean(rt)) mean((rt - r)^3) / mean((rt - r)^2)^(3/2)
```

We can now apply our functions to our three time series to obtain estimated values for the kurtosis and skewness:

```
Kurtosis(AAPL_returns)

## [1] 120.3551

Kurtosis(gold_returns)

## [1] 11.81545

Kurtosis(GBPUSD_returns)

## [1] 14.47611
```

As can be seen, all three distibutions are notably heavy tailed, with $k > 3$, indicating that the distributions are leptokurtic and thus deviate from normality in terms of tail weight. The sample skewness of each time series is calculated to be:

```
Skewness(AAPL_returns)
## [1] -4.335316
Skewness(gold_returns)
## [1] 0.04006626
Skewness(GBPUSD_returns)
## [1] -0.7644839
```

Here we can see that Gold returns are symmetric but that Apple and GBP/USD returns are both negatively skewed with Apple seeming to have the largest negative skew. From the plots on later pages we can see that Apple returns are strongly influenced by one value, the 29/09/2000 where the stock price of Apple crashed and lost more than half of its value, equating to a log return of -73%. So considering this, we should also measure the uncertainty of our estimate. One way would be to calculate the standard error of our measures using the bootstrap, and a function of the form:

very good point

```
BootstrapSE <- function(B, data, func) sd(replicate(B, func(sample(data, replace=T))))
```

Perhaps a better method would be to use a block bootstrap approach to obtain a confidence interval for our estimates of kurtosis and skewness. To do this we first construct a function that generates a Nonoverlapping Block Bootstrap sample, using blocks of length l, from a time series:

```
BlockBootstrap <- function(l,data){
  N <- length(data)
  bootdata <- c()
  for (i in sample(1:(N/l), replace = TRUE)){
    bootdata <- c(bootdata, data[((i-1)*l + 1):(i*l)])
  }
  bootdata
}
```

Then we can use this to generate our bootstrap samples and then construct a studentized confidence interval from them. The block bootstrap method should work better for the dependent data we have here as it should preserve some of the structure of the data.

```
BlockBootstrapConfidenceInterval <- function(B, l, data, func){
  val <- func(data)
  bootvals <- replicate(B,{
    bootdata <- BlockBootstrap(l,data)
    (func(bootdata) - val)/sd(bootdata)
  })
  c.star <- quantile(bootvals, c(0.025,0.975))
  CI <- c(val - c.star[2]*sd(data), val - c.star[1]*sd(data))
  names(CI) = c('2.5 %', '97.5 %')
  CI
}
```

block bootstrap is a great idea!

Applying this function to our three returns datasets, using $B = 10,000$ resamples and blocks of length $l = 20$, to obtain bootstrapped confidence intervals for kurtosis and skewness yields:

```
set.seed(69)
bootsize <- 1e3 # 1e4 originally
BlockBootstrapConfidenceInterval(bootsize,20,AAPL_returns,Kurtosis)
##      2.5 %    97.5 %
##   43.48624 250.39056
BlockBootstrapConfidenceInterval(bootsize,20,gold_returns,Kurtosis)
```

```
##      2.5 %    97.5 %
##  7.582449 16.450266

  BlockBootstrapConfidenceInterval(bootsize,20,GBPUSD_returns,Kurtosis)

##      2.5 %    97.5 %
##  4.100079 24.345733
```

```
  BlockBootstrapConfidenceInterval(bootsize,20,AAPL_returns,Skewness)

##       2.5 %      97.5 %
## -9.3343969 -0.5284195

  BlockBootstrapConfidenceInterval(bootsize,20,gold_returns,Skewness)

##       2.5 %      97.5 %
## -0.6204181  0.6415220

  BlockBootstrapConfidenceInterval(bootsize,20,GBPUSD_returns,Skewness)

##      2.5 %    97.5 %
## -1.592475  0.251750
```

As can be seen here, the confidence intervals are fairly wide, but suggest that the kurtosis is above three, certainly for Apple and Gold returns, and most likely for GBP/USD returns also. There also seems to be some evidence of slight skew too. Overall, all three distributions appear to exhibit SF1.

Stylized Fact 2

We now check that there is no evidence of serial correlation in the time series. For this, we use the empirical autocorrelation function (ACF), given by

$$\hat{\rho}(k) = \frac{\sum_{t=k+1}^{T}(r_t - \bar{r})(r_{t-k} - \bar{r})}{\sum_{t=1}^{T}(r_t - \bar{r})^2} \tag{19}$$

for lags $k = 1, ..., T - 1$. This is easily implemented and plotted in R using the `acf` function. As can be seen by the plots, for $k \geq 1$, the lag value of $\hat{\rho}(k)$ is not significantly different from zero. Of course some lag have autocorrelation values outside of the confidence interval, but we would expect this at most 5% of the time under the null (no autocorrelation), which is evidenced here by all three time series. Therefore we can conclude that all three time series exhibit SF2.

Stylized Fact 3

From the plots of the autocorrelation functions of absolute returns, we can see that volatility is indeed clustered and persistent for all three time series, since the correlations are all significantly non-zero for lags up to at least 35. Therefore we can conclude that all three time series exhbit SF3.

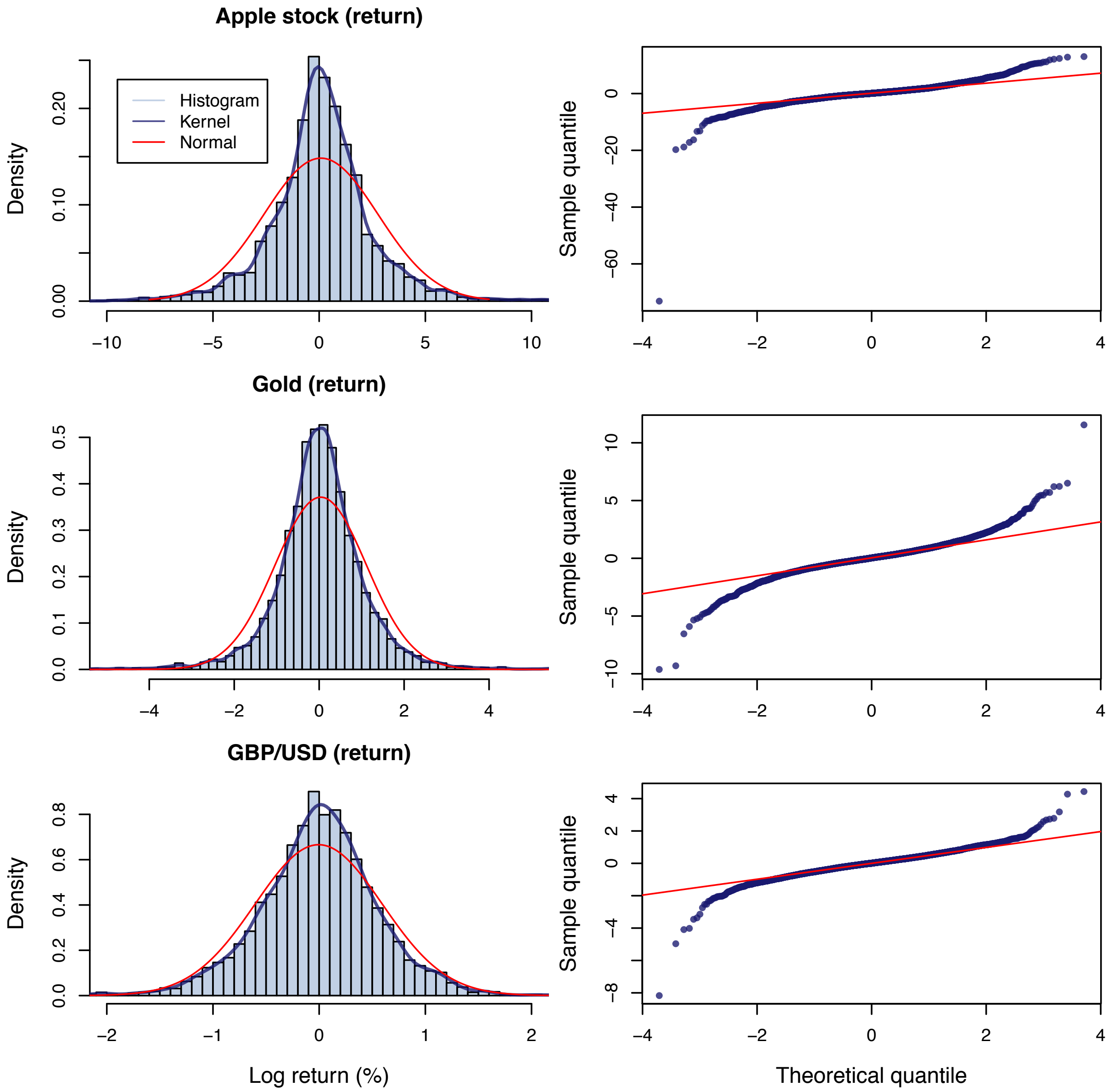

Figure 8: Comparison of log-returns with normal disribution density and quantiles. We can see that the distribution of returns is markedly leptokurtic for all three assets.

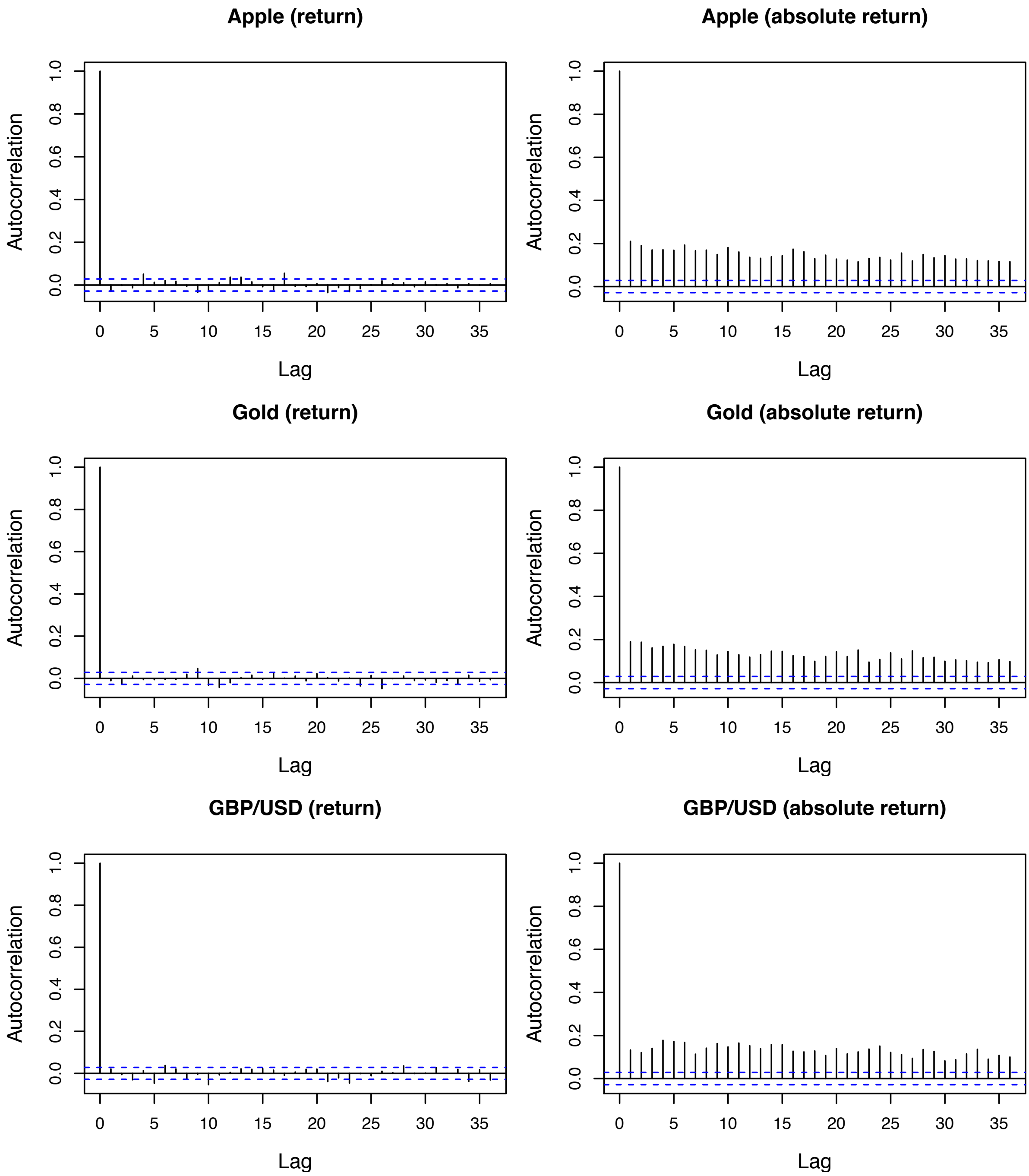

Figure 9: Sample autocorrelations for returns (left) and absolute returns (right).

Part B (ii) For each series, find a distribution that describes well the unconditional distribution of returns. Fit the distribution to the data and assess its goodness-of-fit.

The analysis in Part B (i) indicates that a suitable model for the distribution of returns must be unimodal with non-zero skew and heavy tails relative to the normal distribution. The stable distribution is capable of manifesting all three of these properties and is a commonly chosen in practice for modeling unconditional returns [1].

Stable distributions are, by definition, distributions that result from summing independent and identically distributed random variables. They are unimodal and can be both asymmetric and leptokurtic. Both the Cauchy and normal distributions are stable distributions, but in general there is no analytic expression for the distribution's density.

A stable distribution has four parameters - a stability index $\alpha \in (0, 2]$, a skewness parameter $\beta \in [-1, 1]$, a scale parameter $\gamma > 0$, and a location parameter $\delta \in \mathbb{R}$. The stability index controls the rate at which the disribution's tails diminish. The normal disribution is the stable distribution with the thinnest tails, since it has $\alpha = 2$[2]: for $\alpha < 2$, the distribution's variance is infinite. As α decreases, the distribution becomes increasingly leptokurtic. The parameter β affects skewness - $\beta > 0$ implies that the distribution is skewed right, $\beta < 0$ that it is skewed left, and $\beta = 0$ that it is symmetrical. γ and δ are analagous to scale γ and location x_0 parameters for the Cauchy distribution [3].

For this question we fit both a stable distribution and a Cauchy distribution to the returns datasets to evaluate whether expanding the model space to encompass all stable distributions is necessary.

No closed-form solution exists for the maximum likelihood estimates of the Cauchy parameters. Instead, the likelihood function must be optimized numerically. We do this using the `MASS` package's `fitdistr()` function, as implemented below. It was not possible to obtain maximum likelihood estimates of the stable distribution parameters. The optimization routine used in the `StableEstim` package, `MLParametersEstim`, a procedure described by Nolan [2], was prohibitively slow and was sensitive to the choice of parameter initialization. Instead, we estimate the stable parameters using McCulloch's quantile method [3]. The empirical performance of this algorithm suggests that it is acceptable.

```
suppressMessages(require(MASS))
suppressMessages(require(StableEstim))
suppressMessages(require(stabledist))

AAPL.cauchy.params   <- fitdistr(AAPL_returns, densfun='cauchy')[[1]]
gold.cauchy.params   <- fitdistr(gold_returns, densfun='cauchy')[[1]]
GBPUSD.cauchy.params <- fitdistr(GBPUSD_returns, densfun='cauchy')[[1]]

AAPL.stable.params   <- McCullochParametersEstim(AAPL_returns)
gold.stable.params   <- McCullochParametersEstim(gold_returns)
GBPUSD.stable.params <- McCullochParametersEstim(GBPUSD_returns)
```

The estimated parameter values for the Cauchy and stable distribution parameters are presented in Tables 1 and 2. The maximum likelihood estimates (MLEs) are accompanied by estimates of their standard errors, computed by inverting the observed information matrix evaluated at the MLE.

Figure 10 presents the probability density functions (pdf) and cumulative distribution functions (cdf) of the fitted fistributions alongside the data's histogram and empirical cdf. We can see that the stable distribution is more consistent with the data's histogram than the Cauchy distribution, which has excessively

[2]The normal distribution is a stable distribution with $\alpha = 2, \beta = 0, \gamma = 1, \beta = 0$.
[3]The Cauchy distribution is a stable distribution with $\alpha = 1, \beta = 0, \gamma = \gamma, \delta = x_0$.

Parameter estimated	AAPL	Gold	GBP/USD
x_0	0.0934 (0.0243)	0.0247 (0.0111)	0.0160 (0.0069)
γ	1.113 (0.0213)	0.495 (0.0094)	0.3044 (0.0057)

Table 1: Maximum likelihood estimates (MLE) of Cauchy distribution parameters using log-returns datasets. Estimates of the MLE's standard deviation are shown in parentheses.

Parameter estimated	AAPL (S)	Gold (S)	GBP/USD (S)	AAPL (C)	Gold (C)	GBP/USD (C)
α	1.464	1.587	1.649	1*	1*	1*
β	-0.005	0.0130	-0.0770	0*	0*	0*
γ	1.227	0.5430	0.3432	1.113	0.495	0.3044
δ	0.0877	0.0297	0.0118	0.0934	0.0247	0.0160

Table 2: (S) Quantile-based estimates of stable distribution parameters using log-returns datasets. (C) Maximum likelihood estimates of Cauchy distribution parameters, presented in terms of the corresponding stable disribution parameters. The asterisk * indicates that $\alpha = 1$ and $\beta = 0$ are fixed a priori.

Excellent idea to study log density histograms

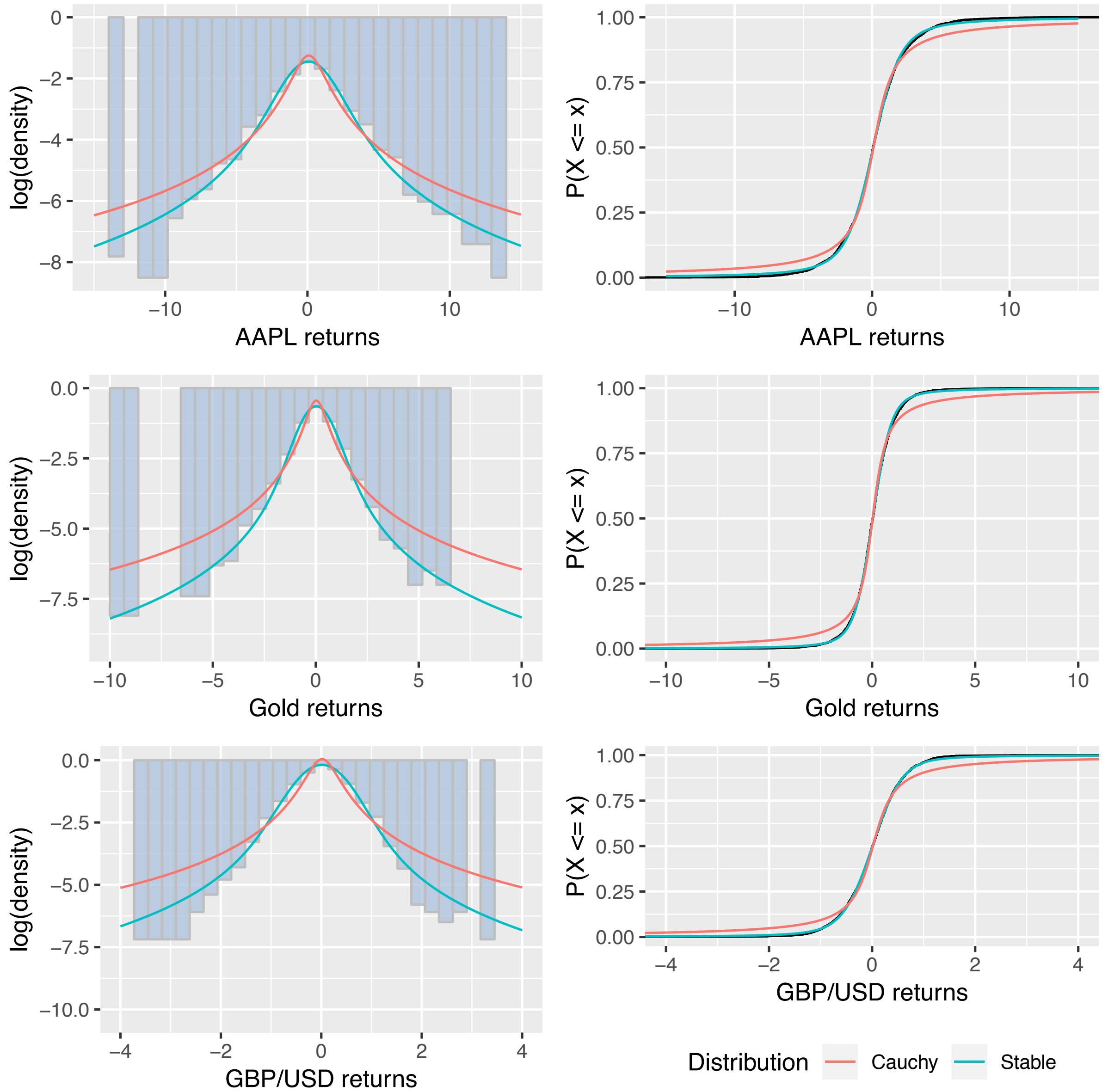

Figure 10: (Left) Log-density histograms for each dataset with the Cauchy pdf for maximum-likelihood parameter estimates and the stable pdf for quantile-based parameter estimates. (Right) Empirical cumulative distribution function for each dataset (black) with Cauchy and stable cdfs.

fat tails and too narrow a mode (i.e. it is too leptokurtic). By contrast, the stable distribution agrees closely with the data in all three cases, as illustrated by the fact that the ecdf and fitted stable cdf are essentially collinear.

We can see in Table 2 that similar estimates for γ and δ are obtained for the Cauchy and stable distributions. Furthermore, all three skewness parameter estimates $\hat{\beta}$ are approximately equal to 0, indicating that their associated distributions are roughly symmetrical. The difference between the Cauchy and stable distribution fits lies exclusively in α, the stability index. The Cauchy distribution takes $\alpha = 1$, but the stable fit suggests a value in the vicinity of 1.5 is more appropriate. This value corroborates our previous observation that the fit stable distribution has thinner tails than the fit Cauchy distribution.

We evaluate the quality of fit statistically by computing the probability that the Kolmogorov-Smirnov (KS) test statistic would be at least as extreme as its observed value under the null hypothesis that the data was sampled from the reference distribution (either the Cauchy or stable distribution). The KS test relies on the assumption that the reference distribution is continuous, which implies that the probability any two observations are exactly equal is zero. The returns data does not satisfy this assumption: there are 28 entries in the Apple log-returns that are equal to 0, probably because the support of the stock's price is not the real numbers, but the countable subset of terminating decimals of six decimal places[4]. To make the KS test applicable, we perturb the data by a value on the order of 10^{-2} so that no two values are equal.

```
set.seed(666)
pstable_ <- function(x, pm) return(pstable(x, pm[[1]], pm[[2]], pm[[3]], pm[[4]]))
pcauchy_ <- function(x, pm) return(pcauchy(x, pm[1], pm[2]))
eps.AAPL <- rnorm(length(AAPL_returns), mean=0, sd=0.001)
eps.gold <- rnorm(length(gold_returns), mean=0, sd=0.001)
eps.GBPUSD <- rnorm(length(GBPUSD_returns), mean=0, sd=0.001)

ks.test(AAPL_returns+eps.AAPL, function(x) pstable_(x, AAPL.stable.params))$p.value

## [1] 0.1467438

ks.test(AAPL_returns+eps.AAPL, function(x) pcauchy(x, AAPL.cauchy.params))$p.value

## [1] 0

ks.test(gold_returns+eps.gold, function(x) pstable_(x, gold.stable.params))$p.value

## [1] 0.314372

ks.test(gold_returns+eps.gold,
        function(x) pcauchy(x, gold.cauchy.params))$p.value

## [1] 0

ks.test(GBPUSD_returns+eps.GBPUSD,
        function(x) pstable_(x, GBPUSD.stable.params))$p.value

## [1] 0.7001488

ks.test(GBPUSD_returns+eps.GBPUSD,
        function(x) pcauchy(x, GBPUSD.cauchy.params))$p.value

## [1] 0
```

The p-values of 0 indicate that the observed log-returns are not consistent with the Cauchy fit. The null hypothesis is rejected at the 5% level. The values 0.15, 0.31, and 0.70 suggest that all three datasets are plausibly consistent with the stable fit, so the null hypothesis is retained. These results indicate that a stable distribution is appropriate for modeling unconditional returns, while the Cauchy distribution is not. This agrees with advice given in the literature [1] [2] [3].

Exemplary work – well done!

Part B: 5/5 marks

[4] i.e. we model a discrete distribution over log-returns using a continuous distribution.

References

[1] Unconditional and Conditional Distributional Models for the Nikkei Index, Stefan Mittnik et al, Asia-Pacific Finanical Markets, 1st May 1998, Volume 5, Number 2.

[2] Fitting Data and Assessing Goodness of Fit with Stable Distributions, John P. Nolan, American University, Washington, June 1999. pp. 6-10.

[3] Simple Consistent Estimators of Stable Distribution Parameters, John H. McCulloch, Communications in Statistics - Simulation and Computation, Volume 15, Number 4, pp. 1109-1136, 1986, Taylor & Francis.