Ch 12: Modelling financial price data

Overview I

- Subject is inherently quantitative but is a core part of finance
- The list of potential applications includes econometric studies, options pricing, risk management etc
- Often a core part of dissertations
  - Need to model percentage price changes
  - Usually simplified in practice to modelling the log-returns
  - Log-returns are the first differences of the log-price

Overview II

- The modelling of price data from cryptocurrencies is a live topic of academic research (see e.g. Katsiampa, 2017)
- Topic extends from the classical study of statistical models for stock price data
- This is not an ideological judgement, in itself, that cryptocurrencies are more of a speculative asset than a genuine currency

Which data do we model?

- **We almost never model the price index directly**
- Usually more informative to look at the percentage change in price.
- **Returns**

$$R_t = \frac{P_{t+1} - P_t}{P_t}$$

- In practice it is usually easier to look at the log-returns
- Define $X_t = \ln\ P_t$ and analyse
- **Log-returns**

$$\Delta X_t = X_{t+1} - X_t = \ln\left(\frac{P_{t+1}}{P_t}\right)$$

Why percentage price-changes are usually more informative

- **Some simple examples**
  1. "**The price today is £100**"
     - This piece of information does not make sense in isolation
     - Was the price yesterday £5 or £300?
  2. "**The difference between today's price and yesterday's price** $P_t - P_{t-1}$ **is £0.1**"
     - This piece of information does not make sense in isolation
     - Was the price yesterday £5.50 or £0.5?

Percentage price-changes give an added sense of scale and direction

- E.g. $R_t = 0.0003$ means the price has increased by 0.03% compared to yesterday's value
- E.g. $R_t = -0.0002$ means the price has decreased by 0.02% compared to yesterday's value
- Especially when they are calculated over short-time horizon's like days and weeks stock market returns tend to show quite low values unless the market is extremely volatile
- For comparison, Black Monday October 19th 1987 would have resulted in a value of $R_t = -0.2261$ as the Dow Jones Industrial Average index lost 22.61% of its value

Where do the log-returns come from?

- **It clearly makes sense to look at returns but the usual convention is to instead look at the series of log-returns**
- There are several reasons for this
  1. Tractability and consistency with standard mathematical finance models
  2. The log-returns series are typically approximately stationary and so easier to model statistically
  3. The log-returns series are typically approximately uncorrelated and so easier to model statistically
- Note: There is usually not much difference between looking at the returns and the log-returns
- Note: Being uncorrelated is not the same as being independent

Log-returns consistent with standard mathematical finance models – e.g. Black-Scholes (non-examinable)

- Black-Scholes (options pricing) model

$$
\begin{aligned}
dP_t &= \mu P_t dt + \sigma P_t dW_t, \\
dX_t &= \left(\mu - \frac{\sigma^2}{2}\right) dt + \sigma dW_t
\end{aligned}
$$

- The log-returns $\Delta X_t = X_{t+1} - X_t$ are then independent and normally distributed with mean

$$
\int_t^{t+1} \left(\mu - \frac{\sigma^2}{2}\right) du = \mu - \frac{\sigma^2}{2}
$$

and variance

$$
\int_t^{t+1} \sigma^2 du = \sigma^2
$$

Differences between returns and log-returns very small (non-examinable)

- Compare the exact return $r_t = \frac{P_{t+1} - P_t}{P_t}$ with the log-return $\Delta X_t = \ln\left(\frac{P_{t+1}}{P_t}\right)$ that is used to approximate it

$$\begin{aligned} \ln\left(\frac{P_{t+1}}{P_t}\right) &= \ln\left(\left(\frac{P_{t+1} - P_t}{P_t}\right) + 1\right) \\ &= \left(\frac{P_{t+1} - P_t}{P_t}\right) + O(r_t^2) \\ &= r_t + O(r_t^2) \end{aligned}$$

Outline of the rest of the lecture

1. Computational work with price data
2. The random walk model
   - Serves as interesting historical background and motivation
3. Stylised empirical facts
   - Since stock price data is so widely studied
   - A range of stylised empirical facts typically shared by stock price data from around the world are widely documented
   - Forms a natural yardstick with which to compare data from Bitcoin and cryptocurrencies and have supervised dissertations on this topic in the past
4. Some tests of stylized empirical facts
5. References

1.1. Computations with cryptocurrency price data

- Usually obtain cryptocurrency data from `coinmarketcap.com`
- usually focus on the log-returns calculated from each day's closing price – so only need a subset of the available data
- Example using Bitcoin prices shown on Canvas. The raw excel file `BitcoinData.xlsx` and the abridged `.txt` version `BitcoinData.txt`
- I would recommend you read the data in from `.txt` format using the command `read.table`
- You need to get rid of any commas in the `.txt` file using the `Edit`⟶`Replace` function
- If you directly copy the log-returns from e.g. MS excel into R this can be a hidden source of rounding error

1.2 Bitcoin example

- **Read in the data using the read.table command**

```
>BitcoinData<-read.table(``E:BitcoinData.txt")
```

- **Clarify how many columns are in the downloaded dataset. In this case the result tells you there are four columns**

```
>ncol(BitcoinData)
4
```

- **The price then has to be identified as the last (fourth) column**

```
>price<-BitcoinData[,4]
```

- **Need the data to be in chronological order from oldest to newest. May have to reverse the data if it isn't already in this format. In R the command to do this is rev**

```
>price<-rev(price)
```

1.3 Bitcoin example (continued ...)

- R's command line structure offers various time and efficiency savings compared to inferior alternatives such as MS excel
- In R calculate the log-returns as the first-difference of the log-prices
- This can be a bit hard to see at first but can be achieved by creating two series

1. Series One with the first observation deleted
2. Series Two with the last observation deleted

- The log-return can then be calculated as

$$\text{Log-return} = \text{Series One} - \text{Series Two}$$

1.4 Bitcoin example (continued ...)

- Calculate the log-return using
  1. Series One with the first observation deleted
  2. Series Two with the last observation deleted
- The log-return can then be calculated as

$$\text{Log-return} = \text{Series One} - \text{Series Two}$$

- In R use

```
>length(price)
2482
>logreturn<-log(price[-1])-log(price[-2482])
```

- **In R the command length tells you how long the series is and the number corresponding to the last observation. The minus sign indicates that you delete the 1st and 2482nd observations**

1.5 An R function to calculate the log-returns

- In financial econometric work in R I have personally found the following function helpful to calculate log-returns based on a price series x listed in chronological order from oldest to newest

```
gradrel<-function(x){
n<-length(x)
logreturns<-log(x[-1])-log(x[-n])
logreturns}
```

- **In R you could then equivalently calculate the log-returns as**

```
logreturns<-gradrel(price)
```

2.1 Overview of the random walk model

• The random walk model is the simplest possible financial model that can give you "reasonable answers"
• The random walk model has links with both the Efficient Markets Hypothesis and to Corporate Finance though it is not always presented in this way
• **Finance is inherently quantitative ...**
• The random walk model is the lens through which we can see how stock market prices really behave!

2.2 The random walk model (non-examinable)

- Mathematically, a random walk is defined as

$$S_n = \sum_{i=1}^{n} X_i,$$

where the X_i are independent and identically distributed (not necessarily normally distributed!)

- Under the Black-Scholes model the log-price X_t can be constructed as

$$X_t = \sum_{i=1}^{t} \Delta X_i,$$

where the X_i are normally distributed with mean $\tilde{\mu} = \mu - \frac{\sigma^2}{2}$ and variance σ^2

2.3 Background to the random walk model (non-examinable)

• The random walk model has a rich history and hints at close links between physics and finance (Weatherall, 2013)

- Originally used as an options pricing model by Bachelier (1900)
- Predates Einstein's work on Brownian motion by 5 years

• For various reasons there was a growth in mathematical finance in the 1950s-1960s

- Osborne improves upon Bachelier's original model
- Other important contributions made by Mandelbrot and Thorp
- First tests of the random walk model and the Efficient Markets Hypothesis by Fama

2.4 Additional historical background (non-examinable)

• 1973. Seminal options-pricing papers published by Black-Scholes and by Merton. Chicago Board Options Exchange established.
• Early 1970s Physics research funding dramatically cut in the aftermath of the space race. Period coincides with increased use of quantitative computer-driven models in financial industries.
• **Finance becomes increasingly quantitative – and will probably continue to do so!**
• 1980s+. Developments in time series econometrics such as ARCH/GARCH in response to empirical failings of the random walk model
• 1990s+. Increases in computer power and data availability occur alongside developments in computational modelling.

2.5 Black-Scholes model

- Under the Black-Scholes model the log-returns are normally distributed and are independent
- **Markowitz interpretation**

  - The mean of the log-returns provides a measure of the rate of return on investment

  - The variance of the log-returns provides a measure of the rate of risk associated with an investment
- For the daily Bitcoin log-returns discussed earlier the mean log-returns is 0.001716242 and the variance of the log-returns is 0.001801658
- In R use *mean(logreturn)* and *var(logreturn)* to calculate these

2.6 Summarising the random walk model

• The key advantage of the random walk model is that it is conceptually interesting and tractable – especially in regard to devising numerical Options-Pricing models

• **However, it is important not to view the random walk model as a purely theoretical devices**

- The random walk model lays the foundation of more advanced and accurate study of financial time series via widely documented **stylised empirical facts**

• The random walk model is also not restricted into having a normal distribution.

- heavy-tailed multivariate random walk models can lead to fruitful risk management applications including analysing contagion.

3.1 Stylised empirical facts – Cont and Tankov (2004) – Chapter 7. See also Cont (2001)

• **The random walk model is not just theoretically interesting**

- Black-Scholes model is used as a baseline from which stylised empirical facts are defined

• **Financial time series are widely studied, are obviously important, and the results of these studies are widely documented**

- Stylised empirical facts use historical data to describe how real stock market prices truly behave

3.2 Stylised empirical facts

1. Heavy tails – higher probabilities of extreme events than under the normal distribution
2. Log-returns are approximately uncorrelated
3. Log-returns are not independent
4. Volatility clustering
5. Central Limit Theorem – returns calculated over a longer time horizon (e.g. days, weeks, months) are closer to a normal distribution
6. Leverage effect – volatility negatively correlated with asset returns
7. Volume positively correlated with volatility

3.3 Stylised empirical facts – a note of caution

- Stylised empirical facts are general rules rather than mathematical laws of nature
- **There may be exceptions to every rule**
- Stylised empirical facts typically formulated for large efficient and liquid stock markets
- **May observe differences for thinly traded, less efficient and less liquid developing and emerging markets**

3.4 Cryptocurrencies and stylised empirical facts

• We would naturally expect cryptocurrency price data to share much in common with these generic stylised empirical facts
• However, if we compare cryptocurrencies to e.g. a developing stock market index we might expect

- some auto-correlation in asset returns, see e.g. empirical work in Katsiampa (2017)
- Very heavy tails as a reflection of extreme price risks. This stylised empirical fact may be especially true for cryptocurrencies

4.0 Tests of stylised empirical facts

- In this section we discuss graphical and numerical tests for stylised empirical facts 1-5
- Stylised empirical facts 6-7 require separate estimates of volatility. Whilst this is possible, e.g. from recently established derivative markets for Bitcoin, this is more involved so we omit this here

4.1.1 Heavy tails

- In R test for the normality of a series using the command `shapiro.test`
- The null hypothesis is that the data is normally distributed
- In finance rejection of the null hypothesis will usually mean that the data has heavy tails (a higher probability of extreme events than under the normal distribution)
- This is the conclusion from our Bitcoin example since `shapiro.test(logreturn)` gives

```
data:  logreturn
W = 0.88778, p-value < 2.2e-16
```

4.1.2 Graphing heavy tails

- Want to see how the normal approximation breaks down not just that it is an inaccurate model
- The easiest way to do this is to use a kernel density estimate which is a special kind of histogram
- The kernel density plot gives us the best estimate of the probability density function of the log-returns
- We can then see how the fitted normal distribution compares to this kernel density estimate

4.1.3 Kernel Density Estimates in R

- In R a kernel density estimate can be constructed using the function `density` applied to the log-returns series:
`dens<-density(logreturn)`
- This produces a grid of x values over which a corresponding y value (kernel density estimate or histogram value is calculated)
- It is easiest to compare this with the y-values that would correspond to the normal distribution
- In order to do the comparison the R function for the normal probability density is `dnorm`
- You also need

1. The mean of the log-returns series
```
mean(logreturn)
0.001716242
```
2. The standard deviation of the log-returns series
```
sd(logreturn)
0.04244594
```

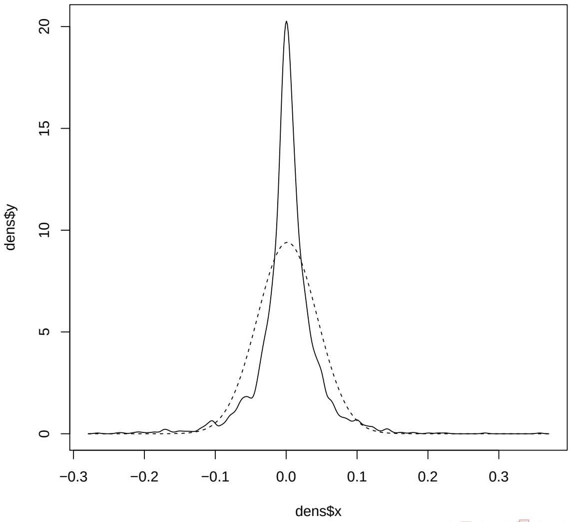
4.1.4 Graphing Probability Density Estimates

1. Plot the kernel density estimate using the x and y co-ordinates of the kernel density estimate
   `plot(dens$x, dens$y, type=‘‘l")`
2. Overlay a line showing the fit of the corresponding normal distribution
   lines(dens$x, dnorm(dens$x, 0.001716242, 0.04244594), lty=2)

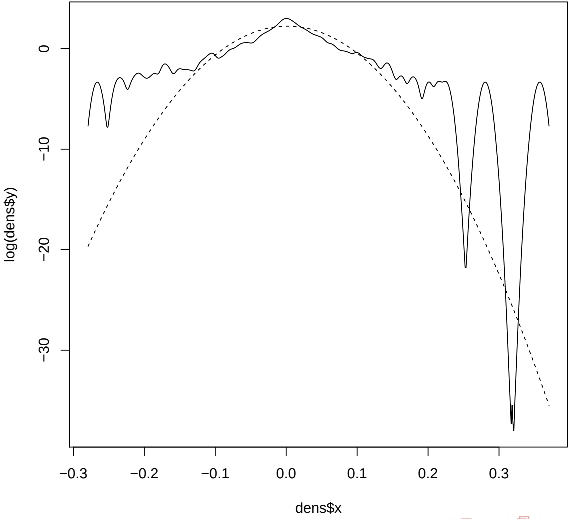
Note

1. Should show much heavier tails in empirical financial data compared to the normal distribution
2. Sometimes the effect is better shown using a plot of the log density

4.1.5 Probability Density Plot

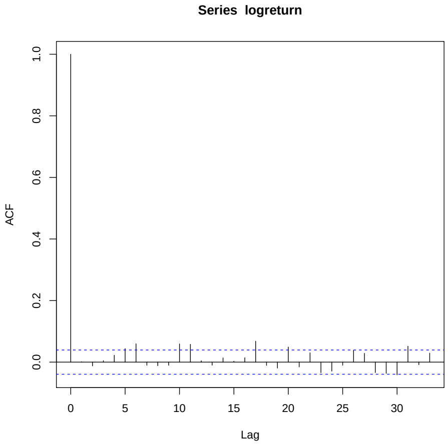


4.1.6 Log (Probability Density Plot)



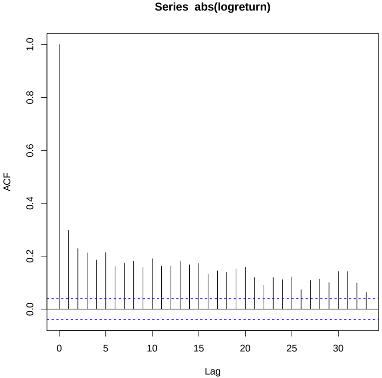
4.2 Log-returns approximately uncorrelated

- If this stylised empirical fact is true then the ACF plot constructed should have all the points within the “tramlines”
- In R use `acf(logreturn)`



4.3 Log-returns are not independent

- Log-returns are not independent
- This feature is also sometimes described as long-range dependence in volatility
- The ACF of the absolute value or modulus of the log-returns should suggest autocorrelation. In R use `acf(abs(logreturn))`



4.4.1 Volatility clustering

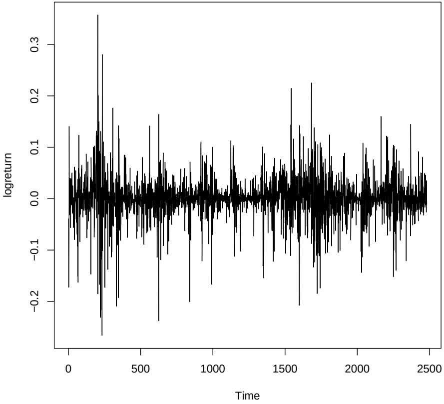
• We will discuss ARCH/GARCH modelling to account for volatility clustering in the next lecture

• Whilst ARCH and GARCH models give a formal statistical test for volatility clustering some important points to bear in mind are as follows

- Purely graphical measures of volatility clustering are still useful
- The behaviour of price volatility will be richer (and inevitably more dangerous) than any mathematical or statistical model can describe

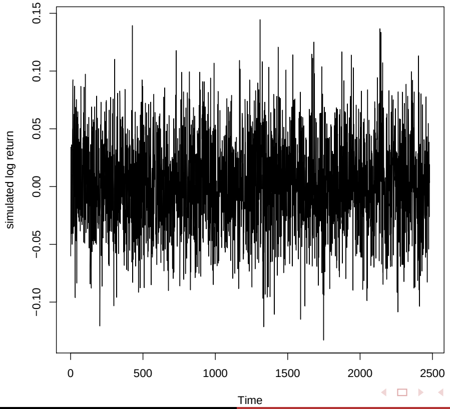
4.4.2 Volatility clustering

- In contrast to how simulated data from the normally distributed random walk model prices clump together around groups of large spikes



4.4.3 Comparison with simulated data

- Simulated data from a normal random walk model looks too smooth compared to real price series
- This may not be easy to see. The other thing to look at would be the scale on the *y*-axis



4.4.4 R-code to test for volatility clustering

- In R the time series plot is produced using `ts.plot(logreturn)`
- In R to produce the simulated data plot you need to know

1. **The length of the series**

   `length(logreturn)`

   `2481`

2. **The mean of the series**

   `mean(logreturn)`

   `0.001716242`

3. **The standard deviation of the series**

   `sd(logreturn)`

   `0.04244594`

- A plot of the simulated data can then be constructed using

```
ts.plot(rnorm(2481, 0.001716242, 0.04244594),
ylab=``simulated log return")
```

4.5 Central Limit Theorem effect

• A typical finding is that as the return horizon increases prices should become closer to a normal distribution
• Simple examples about how the effect should manifest itself include

1. Returns calculated over a day should be closer to a normal distribution than returns calculated every 15 minutes
2. Returns calculated over a week should be closer to a normal distribution than returns calculated over a day
3. Returns calculated over a month should be closer to a normal distribution than returns calculated over a week
4. Returns calculated over a year should be closer to a normal distribution than returns calculated over a month

5. References

Cont, R. (2001) Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance* **1** 223-236.

Cont, R. and Tankov, P. (2004) *Financial modelling with jump processes.* Chapman and Hall/CRC, Boca Raton London New York Washington D.C.

Katsiampa, P. (2017) Volatility estimation for Bitcoin: A comparison of GARCH models. *Economics Letters* **158** 3-6.

Weatherall, J. O. (2013) *The physics of finance. Predicting the unpredictable: Can science beat the market?* Short Books, London.