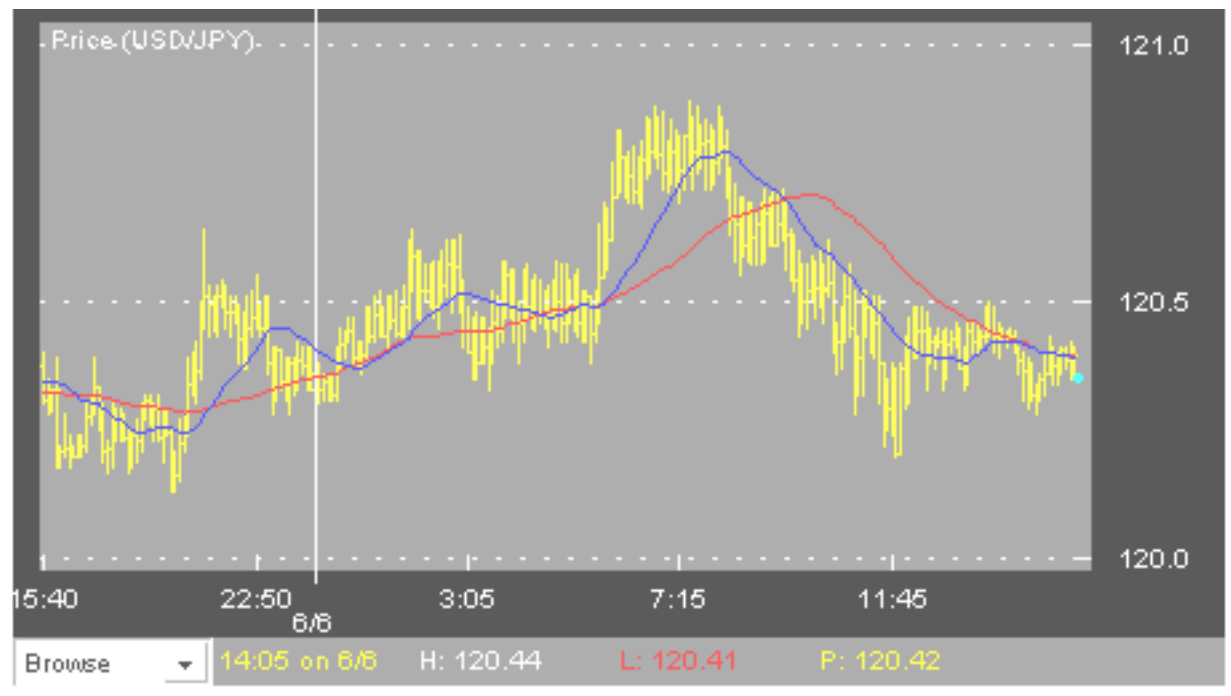


Trend Following Strategies in Trading

By Peter Elliott

00858111



Supervisor: Prof. Peter Richmond

Abstract

This report examines the interesting results found by J. James in a paper that appeared in *Quantitative Finance* and attempt to reproduce them. This entailed using a moving average, for a given pair of currencies, to determine when to change money (trade signals) from one currency to the other. The number of days n used to calculate the moving average is found by seeing which value of n gave the best profit over the last 10 years. To do this, I wrote a program in C and found I was able to exactly reproduce the results of James. I also investigated how many trade signals were produced for each value of n between 10 and 100, and found that when an average is taken over 12 currency pairs, we find that this obeys a power law. I also looked into a slightly different method for calculating the trade signals where the moving average value n to be used is updated each day. The number of preceding days used to recalculate n is optimised to give the best result and I found that I could achieve reasonably good values but this method does not appear to be viable as the optimised value for one range of data does not yield sizable profits for the subsequent data points.

Acknowledgements

I would like to thank the following people for their help with this project. Prof Peter Richmond, Lorenzo Sabatelli, Jessica James and my project partner Zora Law.

Trend following strategies in trading *P. Richmond*

A recent paper by James describes how trend following strategies can be used for profitable trading in currencies. The idea is to automate the buying or selling process depending on the position of the price relative to a long time moving average value. What are the optimum points at which buying and selling can be done profitably over a long time period? The objective of this project is to write code that reproduces the work of James and if time is available test out the model for other asset prices.

Ref: J. James, QF Vol 3 No 4 August 2003 C75

Contents

<u>Introduction</u>	1
<u>Background Theory/Method</u>	2
Part I	2
Part II	4
Part III	5
<u>Results</u>	6
Part I	6
Part II	7
Part III	9
<u>Discussion/Conclusion</u>	10
<u>References</u>	11
<u>Appendix</u>	12
Graphs	12
Program Description	13

Introduction

The underlying idea behind this project is that you can move money back and forth between a pair of currencies and make a profit. For example, this time two years ago €1 would get you \$0.90 whereas now €1 will buy \$1.25, if two years ago you had \$100 and you changed it all into Euro, then if you moved it back today, you would have \$139. So it is clear to see that substantial profit can be made if you know when to move your money. That, of course, is the problem, without knowing when to move, you can just as easily make a loss.

A simple method for calculating these buy and sell points was published by J. Jamesⁱ in 2003, here she uses a moving average and its value in relation to the currency ratio to say whether you should move your money. Despite the simple nature, James claims that it leads to interesting results and can give an average of 5.57% profit when you invest in a portfolio of these currency pairs.

The purpose of this project is to verify the results in James's paper and experiment with possible ways of improving it. The majority of this project was writing code in C that could calculate the moving average, calculate the crossover points where you change from one currency into the other and then to calculate what profit is made by using this strategy. With the short time left after doing this I investigated two areas. The first was how the number of trades for a given moving average differs from currency to currency and the second was whether calculating a new moving average for each day gave a better return than using a single moving average for that time period.

Background Theory / Method

In part I of this section, I shall detail the ideas behind each part of the main program and explain how and why there are done in this way, part I .

I shall then explain the idea behind the additional investigations, parts II and III.

Part I

Overview of Program

The program can essentially be split into the following pieces; the words in italics are explained in more detail below:

- Read in the data from a file.
- Calculate the *Moving Average* for a given value of n.
- Find the *Crossover Points* which tell you to trade.
- Calculate the *Profit* each day and subtract *Costs* if there is a trade.
- Run this for many values of n and find which value gives the largest profit.

Moving Average

As the name suggests, this is the average of the values over the last n days. It is calculated as follows:

$$Av_n(t) = \frac{1}{n} \sum_{j=0}^{n-1} x[t-j]$$

where t is the day number.

So for each day greater than the number of days needed to get an average, we have a moving average. This can be plotted on a graph to see the crossover points

Crossover Points

If we plot the moving average and the currency price, figure 1, we can see that there are crossovers on days 23, 57, 73, 78, 79, 91 and 99. Depending on whether the price crosses going up or down, we know which way to move the currency.

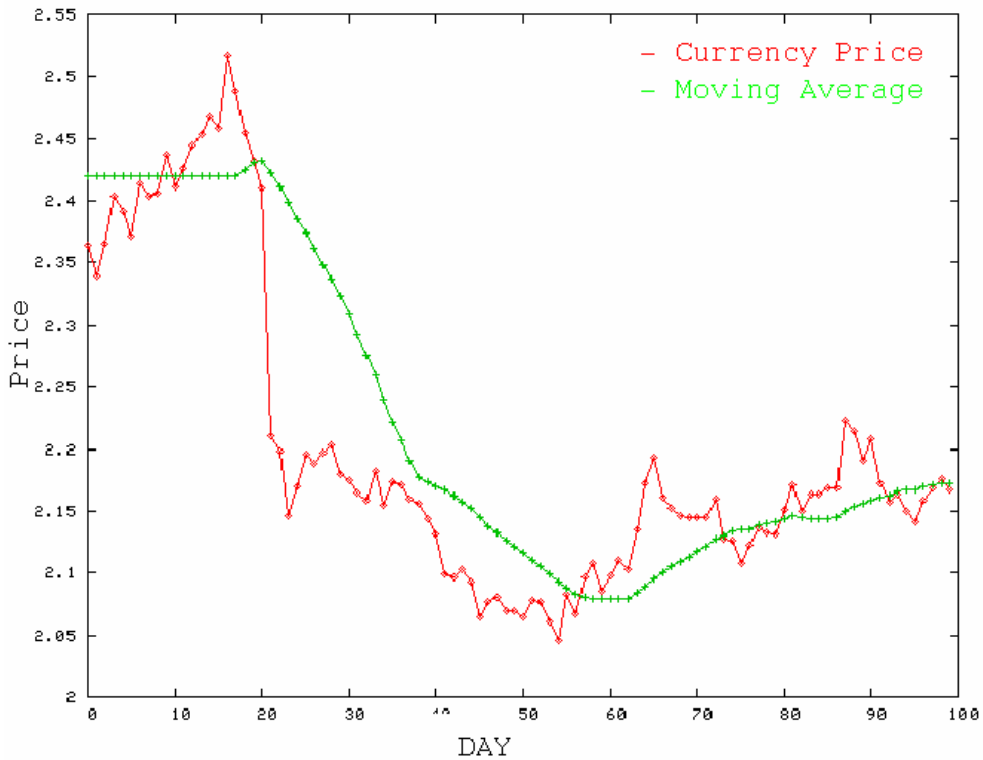


Figure 1. Graph of Moving Average and Currency Price

In the program, these are calculated in the following way.

On each day, if the moving average is larger than the price, then a variable called the signal is -1 and if the moving average is less than the price, then it is +1.

If on one day its was -1 and the next it is +1, then a crossover must have occurs and the money invested is moved into the other currency. Figure 2. shows why this is true.

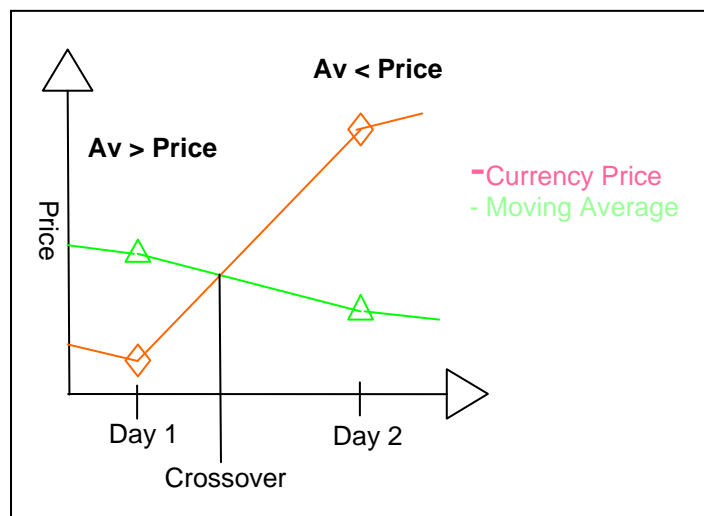


Figure 2. Graph Illustrating Crossover Point

Calculating Profit

The profit from one day is calculated as follows

$$\text{Profit} = 100 \times \text{trade} \times \frac{x[i] - x[i-1]}{x[i-1]} \quad \text{where } \text{trade} = \pm 1$$

This is the percentage loss/gain on money invested the day before at price $x[i-1]$. The trade signal comes from finding the crossover points, it tells you which currency you are in. This is important as if you move out of a currency before it crashes, then this constitutes a profit. However this would be a loss if you had stayed in that currency. Another point here is that a crash in one currency looks the same as the other currency doing very well, from the point of view of the data read in, i.e. the ratio.

We then add up all the day to day percentages to find the total profit. We do it in this way as it lets us see the underlying pattern rather than compounding the return and making it dependence on the start date.

Trading Costs

Each time you change money from one currency to another, you incur a trading cost of 0.03%, in the program this mean that at each crossover point, 0.03% is subtracted from the total profit.

All of these will give you a working program that will loop over all values of n and find which one gives the best return. This value will then be compared with the value found by James.

In James's paper, the Information ratio is calculated for each currency pair. This is defined as

$$IR = \frac{(\text{average annual return}) \times \sqrt{12}}{(\text{std. deviation of monthly returns})}$$

This is quite easily calculated in our program but I decided not to evaluate it and concentrate on part II and III. The same is true for the volatility adjustment which in principle should only require a few extra lines of code. I decided that the little time I had after completing the main program and testing it, would be better spend on the following.

Part II

For each moving average, there will be a number of crossover points. These were calculated for each currency pair to see if there are any similarities.

Part III

A new method for calculating the values of the variable *trade* and hence the crossover points was tested here.

Each day the optimal moving average over the last d days was calculated, where d is some value greater than 100. From this value, the *trade* signal is calculated and used to calculate the profit. We loop over the all the values of d and see if which value gives the best return.

This is done as I thought that just because a value of $n=69$ (say) for the USD/JPY gave the best profit if it had been used over the last 10 years say, it doesn't necessarily mean that it will give the best profit for the next 10 years. Of course, it may well do and this is the basic for making any money using the program, but it does not seem obvious to me that it should be, so I tried a method that actively updates the moving average as it goes along. The idea being that perhaps the new moving average works in the short term but not over 10 years.

Results

Part I

The program written gets exactly the same answers as James. For example, the USD/JPY

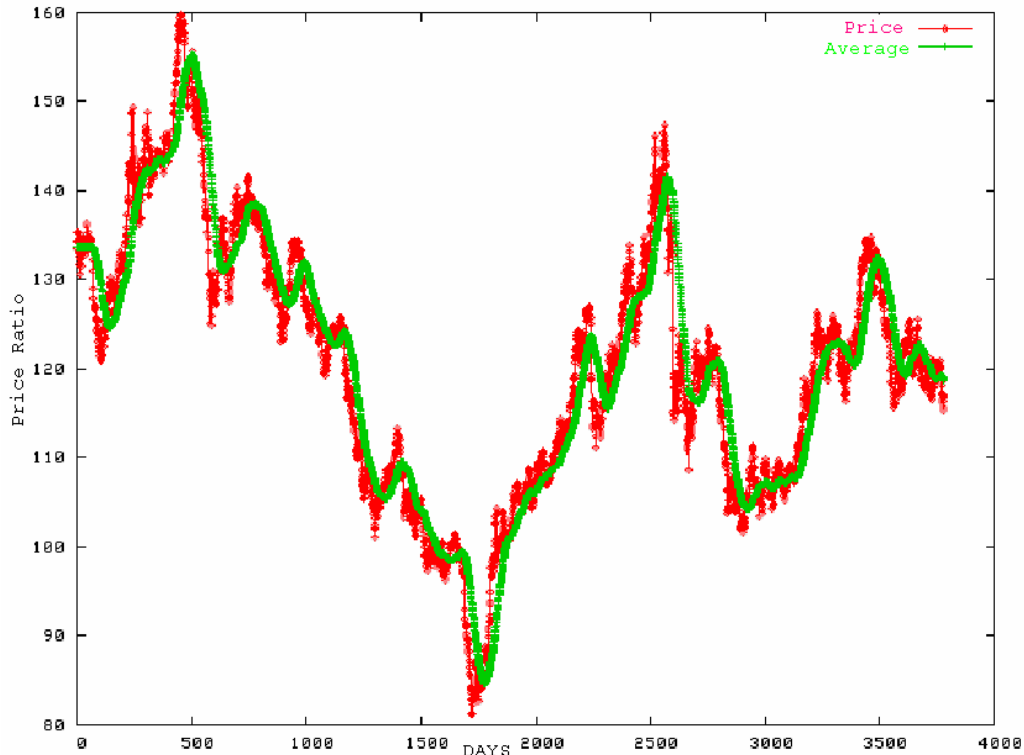


Figure 3. USD/JPY over 3780 days with moving average of 69 days

When my program cycled through all values of n between 10 and 200, it found the best profit was tot 133.744457% for 3780 data points (days).

$$\begin{aligned} \text{Average Annual Return} &= 12 \times \text{Av. monthly return} = 12 \times \frac{\text{Return} \times (\text{Days in a month})}{\text{Days}} \\ &= 12 \times \frac{133.74 \times 21}{3780} = 12 \times 0.743 = 8.92\% \end{aligned}$$

This is exactly as James finds.

Here are the results for the other currency pairs

Currency Pair	Best Moving Average	Average Annual Return	James' Values	
			Moving Average	Av. Annual Return
USD/JPY	69	8.92%	69	8.92%
USD/CHF	64	5.05%	64	4.95%
USD/AUD	173	2.94%	23	1.09%
EUR/USD	89	5.12%	89	4.76%
EUR/JPY	71	6.83%	71	6.60%
EUR/GBP	80	3.91%	80	3.71%
EUR/CHF	116	0.79%	116	0.77%
GBP/USD	19	3.02%	19	2.79%
GBP/JPY	126	5.06%	126	5.13%
CAD/USD	23	0.39%	23	-0.04%

As you can see the results agree nearly perfectly. In actual fact the final figure is exactly the same however there are slight differences in the Average Annual Return due to a difference in the calculation of these values, the important point is that both ways agree on the moving average to use for maximum profit. One small difference is that I have a value of 173 for the USD/AUD case, James's program gets the small result if the moving average is allowed to take this value.

Another slight difference is that James finds the value n that gives the greatest IR, in most cases this is the same as the value that gives the greatest profit, however in the GBP/USD, a value of $n = 7$ gives the greatest IR but 2.86% annual return, less than my value. However, a greater IR will mean that over more time, the $n = 7$ value will do better.

Part II

When the various values for each currency pair are plotted on the same graph, the result is the following

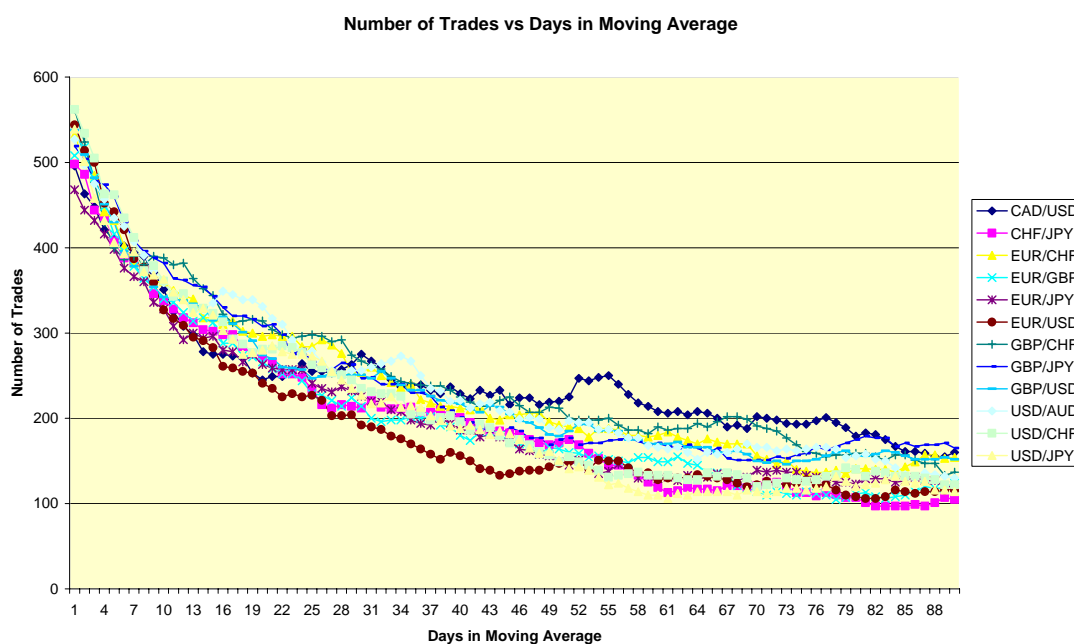


Figure 4. Graph of number of trade signals generated for a given value of n for 12 currency pairs

This is a very surprising result. One would not expect that the shape and approximate values for each slope would be so similar. They all seem to have roughly the same values for a given slope number of days in the moving average.

However, it occurred to me that this might be caused by using a moving average in the first place. It might be an inherent property of the moving average that it will cross a certain number of times for fluctuating data (obviously if the data just goes monotonously downward, the moving average will never cross).

To test this, I created a program to generate a random walk, with a 50:50 chance of going up 1 unit or down 1 unit. This data was treated like a currency pair and the result is shown in figure 5

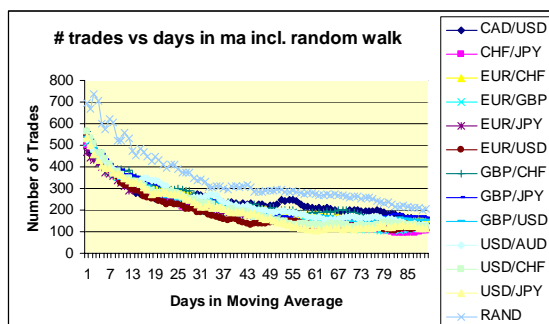


Figure 5. Fig. 4. with random walk data

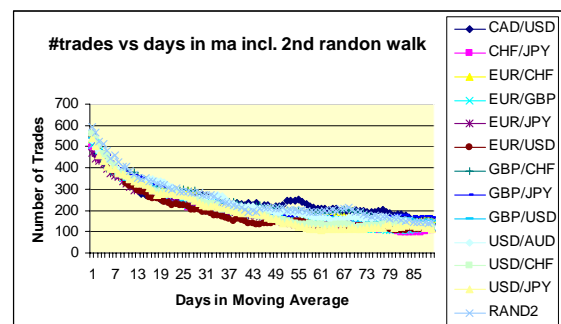


Figure 6. Fig 4 with second type of random walk

As you can see in Fig 5 the normal random walk has many more trades than the currency pairs however the basic shape is roughly the same. This leads me to suspect that this may be a common feature to all currency types.

As this point I decided to modify the first random walk slightly. In the second type, if the random number (between 0 and 1) was above 0.5 two times in a row, the value increased by three instead of two, and vice versa for less than 0.5 twice. I did this as I thought that if an upward trend was continued even just a small bit, the moving average would cross less and maybe have values around the range of the currency pairs. The result is Fig 6 and it is clear that is certainly looks similar but doesn't appear quite right.

I then decided to leave the random walk and just plot the average of all the values for each value of n (days in moving average). This is plotted in Fig 7 and the data is fitted with a power law.

As you can see the results all seem to lie on the line:

$$y = 2419.3x^{-0.62302}$$

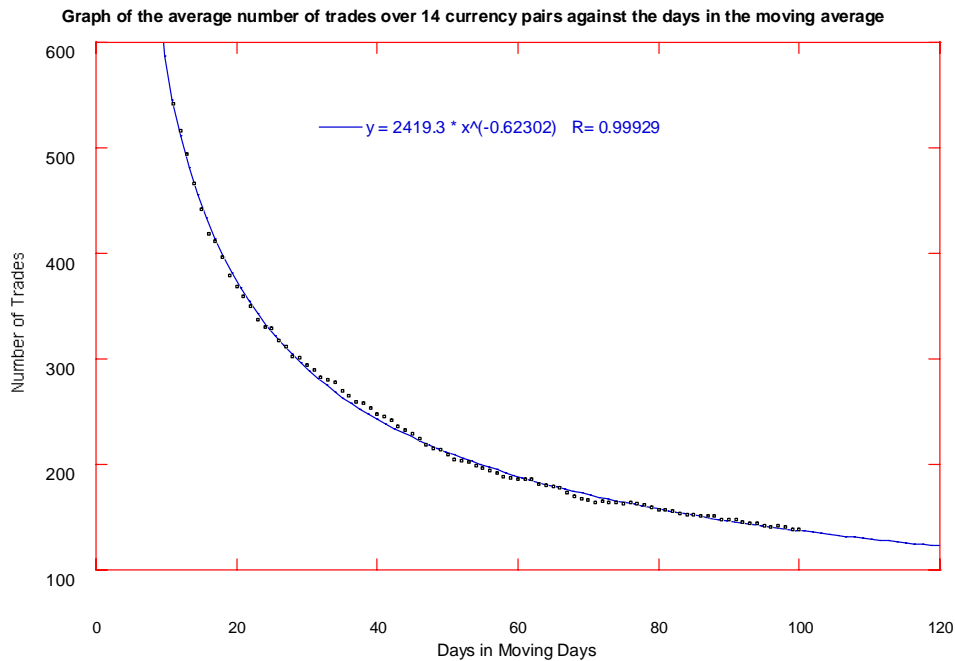


Fig 7 Graph of Average Number of Trades vs Days in Moving Average

Part III

Due to time constraints (it takes 8 hours to run each time), this program was only run twice. The first was 3000 data points for the USD/JPY currency pair.

The result was a total profit of 93.21% by using a value of 659 preceding days over which the best moving average was calculated each day. If we calculate the average annual return for this, remember profit only started accumulating after 659 days, we get a value of 10.03%. This figure is slightly better than James's method however when the program is run for 3780 days, the annual return drops to 6%, so this method seems to be far less reliable and I would predict that the IR is much lower for this case.

The currency pair CAD/USD was also tried as it has a very low IR and may benefit from having a changing moving average.

The best average annual return was 0.48% using 152 days. Again this is slightly better than the simple case however I suspect that again it wouldn't lead to greater profits if taken over longer time periods.

Discussion / Conclusions

Overall the project was successful in that I was able to completely reproduce the results of James's which was the main objective. However I still have doubts over the validity of the method. I would have liked to use more data and see whether the best moving average can change. That is whether a value of n calculated over the last 5 years say will give the best profit for the following 5 years. However from looking at my data, as long as the value of n doesn't change by a large amount, values near to the best value also make reasonable profits. So I would conclude that this method is a good way to use past data to see what may give a profit in the future. Also when the best currencies are selected for a portfolio like in James's paper, a steady profit can be made.

In Part II, the number of trades for a given moving average value does appear to follow a power law. If we take the standard deviation to be the error for each point (see graph in Appendix), we can see that nearly all the currencies tend to follow this law. It does not appear to be due to using a moving average and may be a property of Currencies. Unfortunately I did not have time to fully investigate this, but it is certainly something worth further study.

In Part III, I tried a different method to see if I could improve on James's method. At first it appears that it may have done slightly better but as I said I doubt it would work better in the long run and the end value appears to be quite sensitive to the value over which the new moving average is found. As mentioned about, a number near the optimum value for the moving average still gives a sizable return however in this new method, there can be a large change. So if the program is run over an extra two years, the best value for d will be different while if the previous value for d had been continued to be used, the profit doesn't necessarily continue in the way it would predict of gaining an average value per year. However I feel that this method also deserved further consideration for the reason that it is able to adapt for changes in the moving average. It also has the theoretical advantage of always having the best moving average for that period. So perhaps with some adjustments that stop the moving average varying wildly, a lot of the time it stays around a certain value but every couple of days it might jump but then return back to that value. So with some fine tuning this method may be able to do as well if not better than James's method.

Overall, I would deem this project a success as not only did I achieve the primary objective but I was also able to investigate two areas that looked interesting.

References

- [1]ⁱ J. James, QF Vol 3 No 4 August 2003 C75
- [2] J James QF Vol 3 No 5 October 2003 C97
- [3] M.Ausloos, K. Ivanova, arXiv:cond-mat/0201587 v1 31 Jan 2002