Data Analysis and Programming, CA2
This CA is worth 20%

John Burns

November 24, 2017

Submission date/time: 17:00 15 December 2017

1 Stock performance

Imagine you are a stock market analyst who believes that what goes up must continue to go up. Your task is to identify stocks whose average daily gain in a certain time period is higher than the overall average daily gain of the entire stock exchange in that time. We will use a sample of this data set containing the daily gain (or loss) made by ten technology stocks for the last 90 days. Your solution to this task should access the data in one of the three formats provided and print out the names of those stocks whose gain is higher than the average. Your solution should allow the time frame to be restricted to the last d number of days, in other words should compute which stocks beat the overall average when only the most recent d days are considered. The data is provided in three formats: A simple csv file (`stocks.csv`), a numeric-only csv file in which the stock names are replaced by id numbers (`stocksNumeric.csv`) and an SQLite database (`stocks.sqlite`). In all cases the columns are

1. day: the day number from 1 to 90, where 1 is the most recent day
2. stock: the name or id number of the stock
3. open: the opening price
4. close: the closing price
5. gain: the gain or loss for that day in relation to the opening price

The stocks are

Ticker Symbol	ID
AAPL	1
GOOG	2
ORCL	3
INTC	4
SYMC	5
FB	6
CSCO	7
XRX	8
IBM	9
MSFT	10

So the following row in the data means that two days ago Apple's stock price rose by about 1.9%: `2,"AAPL",84.05,85.66,0.0191552647233789` You may like to begin by computing the average for the whole stock exchange (ie all ten stocks). You can put the number of days d into an R variable which is then visible to your functions. You may also like to solve the task in a single-threaded, in-memory way to make sure your solution is logically correct. The goal of this exercise is to demonstrate that you can provide a scalable solution to this task, so the ideal solution will be both parallelised and use an out-of-memory data source.

1.1 Notes

1. Your solution should contain a textual description (ca. 1 page pdf) of your approach to the problem, including any assumptions you make.
2. Your code must be runnable, in other words it should be possible to source your script in a brand new R session with the data set in the working directory. If you use any R libraries (you may or may not), add a comment stating which functions you are using from that library.
3. You will get more marks if your code is readable, well-indented, and above all well-commented.
4. You will get more marks for appropriate use of the built-in functions of R and its libraries.

1.2 Marks

1. Correctly solve the problem for the 90 day data: **up to 50 points**
2. Be able to limit the analysis to a certain number of days: **10 points**
3. If your solution is parallelised: **extra 20 points**
4. If your solution uses out-of-memory data: **extra 20 points**