Programming for Big Data: Assignment 3

Mark Buckley
mark.buckley@dit.ie

Submission date: 28 May 2013

1 Tax underpayment (20 points)

Imagine you are an analyst with the Revenue Commissioners. Part of your job is identifying characteristics of people who are likely to have underpaid tax. Imagine you have been asked to design a solution to automate the process in R and generate regular reports.

You can assume that

- Revenue has a record of everyone's tax information, such as historical tax payments, income, what income tax band they are in, whether they have paid their property tax, whether they have underpaid tax before, etc.
- Revenue has detailed personal information about each taxpayer, such as age, full address, family status, employment status, profession and so on.
- Revenue maintains all of its data in one giant multi-gigabyte database table with one row per taxpayer.

Note: This exercise is about the data analytics process with R, but you do ***not*** need to write any R code in your answers.

Part A. Suitable Analyses Propose at least three hypotheses about the data which may help find cases of underpayment of tax. For instance you might consider whether the self-employed pay less tax than PAYE workers, or the relationship between the amount of tax paid and other characteristics of each taxpayer. You can assume any attributes you like are available in the database, so be creative.

Which graphical plots and which statistical tests that R provides are suitable to investigate each hypothesis you have proposed? Give concrete examples.

Part B. Data storage Comment briefly on the suitability (or otherwise) for this task of each of the out-of-memory data access methods which we have seen in the lecture (SQL, bigmemory, RHadoop).

Suggest some ways in which the data set could be broken down into chunks for incremental processing with an out-of-memory storage technology such as MySQL or bigmemory.

2 Stock performance (65 points)

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

Stock name	ID
AAPL	1
GOOG	2
ORCL	3
INTC	4
SYMC	5
FB	6
CSCO	7
XRX	8
IBM	8
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.
Notes:

- Your solution should contain a textual description (ca. 1 page pdf) of your approach to the problem, including any assumptions you make.
- 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.
- You will get more marks if your code is readable, well-indented, and above all well-commented.
- You will get more marks for appropriate use of the built-in functions of R and its libraries.

Marks

- Correctly solve the problem for the 90 day data: up to 20 points
- Be able to limit the analysis to a certain number of days: 5 points
- If your solution is parallelised: extra 20 points
- If your solution uses out-of-memory data: extra 20 points

3 R and Hadoop (15 points)

Note: You are ***not*** required to run or test any RHadoop code in this exercise, however code sketches should be provided.

Your task is to outline an alternative solution to the previous problem which uses RHadoop.

Part A Specify the types of your mappers and reducers, and state informally what each one does. Ensure that your MapReduce specification delivers data in the form that it can be further processed in R after you retrieve it.

Part B Sketch what a call to the rmr2 function "mapreduce" should look like. You don't have to include all details of the map and reduce functions themselves, but you should write pseudocode explaining what happens in each function.

4 Notes

- Late submissions will be penalised at 10% per day