Data Management for Big Data Analytics

23 - 31 July
PKU
What do data analysts do?
PROGRAMMING LANGUAGES

- SQL: 70%
- R: 57%
- Python: 54%
- Bash: 24%
- Java: 18%
- JavaScript: 17%
- Visual Basic / VBA: 13%
- C++: 9%
- Scala: 8%
- C#: 8%
- C: 8%
- SAS: 5%
- Perl: 5%
- Ruby: 3%
- Go: 1%
- Octave: 2%
- MATLAB: 9%

SHARE OF RESPONDENTS

Source: O'Reilly Data Science Salary Survey 2016
Two exceptions were “Natural Language/Text Processing” and “Networks/Social Graph Processing,” which are less tools than they are types of data analysis.

One hundred and fourteen tools were present on the list, but over 200 more were manually entered in the “other” fields.

Figure 1-10. Most commonly used tools (used by at least 10% of sample)

Just as in the previous year’s salary survey, SQL was the most commonly used tool (aside from operating systems); even with the rapid influx of new data technology, there is no sign that SQL is going away.
Correlations were tested using a Pearson’s chi-square test with $p=0.05$. That SQL/RDB is the top bar is no surprise: accessing data is the meat and potatoes of data analysis, and has not been displaced by other tools. The preponderance of R and Python usage is more surprising—operating systems aside, these were the two most commonly used individual tools, even above Excel, which for years has been the go-to option for spreadsheets and surface-level analysis. R and Python are likely popular because they are easily accessible and effective open source tools for analysis. More traditional statistical programs such as SAS and SPSS were far less common than R and Python.

By counting tool usage, we are only scratching the surface: who exactly uses these tools? In comparing usage of R/Python and Excel, we had hypothesized that it would be possible to categorize respondents as users of one or the other: those who use a wider variety of tools, largely open source, including R, Python, and some Hadoop, and those who use Excel but few tools beside it.

Python and R correlate with each other—a respondent who uses one is more likely to use the other—but neither correlates with Excel (negatively or positively): their usage (joint or separate) does not predict whether a respondent would also use Excel. However, if we look at all correlations between all pairs of tools, we can see a pattern that, to an extent, divides respondents. The significant positive correlations can be drawn as edges between tools as nodes, producing a graph with two main clusters.

Source: O'Reilly Data Science Salary Survey 2013
Future data scientists’ favorite tools
wrangling

analytics
(up to 80%) wrangling

analytics
Challenges

• **4 Vs of big data**

• **Volume** - data is large

• **Variety** - data comes in different formats

• **Veracity** - data is not there/uncertain/dirty

• **Velocity** - speed of change
Volume challenges

• Even scanning data can take hours/days/weeks
• How to get to the right data?
• Precise answers to queries are impossible
• Hence need to approximate
Variety challenges

- Different data formats
- Still much of the data is stored in relational DBMSs
- But other models catch up
- Most active these days is **graph data**
- We will look at it a lot
Veracity challenges

• How to deal with uncertainty or incompleteness? Relational databases (SQL) are really bad at it

• What to do if data is structured under a different schema, or different bits of data reside in different databases? (Data exchange and integration)

• What to do if data is supplemented with additional knowledge, e.g., an ontology, to compensate for missing data?
Course structure

• A quick of reminder of the basics of relational databases

• SQL: how well do we understand it?

• Volume: Conjunctive queries (many-way joins)
  • optimisation
  • approximation
• Volume: scale independence
• Variety: graph databases
  • theoretical languages
  • property graphs in Neo4j
• Variety: RDF data
• Variety: tree-structured data (XML) - depending on time
• Veracity: incomplete information and correct answers

• Veracity: data integration and exchange

• Veracity: answering queries with the help of ontologies
Evaluation

• Project

• There is a long list of papers on the web

• Choose one of them

• The goal is to write an essay, 6-8 pages

• It must present a summary of the paper that would be understood by someone who has not read the paper

• It should also provide some of your own ideas or further investigation about the paper
• Examples:

• analysis of the followup literature to see how these ideas were used

• ideas on improving algorithms in the paper, perhaps in some special cases

• an implementation of a theoretical algorithm to see how it performs
2-stage process

• Stage 1: this week (and weekend), choose the paper, read it quickly, and decide what you want to do in addition to summarising its ideas

• Have a quick presentation Tuesday 31 July (during the last 2 hours of the course)

• The do the proper writeup and email it to me, libkin@gmail.com, before To Be Announced