Processing UK Prescribing Data Using SparkSQL : CS784 Project

Derek Paulsen

April 25, 2019

1 Introduction

2 Motivation

Prescription medication has been the center of many controversial discussions. From sky high pricing in the US, to unethical prescribing practices, prescription drugs have been a hot topic. Because of this we choose to examine the prescription drug trends. In particular we choose to look at the drug prescribing trends in the UK. This decision was made for two reasons. First, because the UK has socialized healthcare, there is much open data that can be found, which is both complete and reliable. This is crucial since we endeavor to try to infer causal relationships, having confidence that the data is free of noise (as much as it can be) is necessary. Second, since the UK has socialized healthcare (including prescription drug coverage) we wondered if we could find differences in trends compared to the US. In particular we wondered if factors which have been shown in the US to affect healthcare and availability of prescription drugs, such as income, would have the same effect in the UK.

3 Related Work

3.1 SparkSQL [1]

Interactive querying of big data is very common task in nearly any data intensive field. In the past this kind of data analysis was done with a RDBMS (for example, PostgreSQL). While this provides interactive querying, SQL is not well suited for complex data analysis for a few reasons. First, most RDBMS's lack easy integration with common scripting languages (such as Python), that is, a user must read a table into memory and then process it with their scripting language. SparkSQL provides a fully featured relational model for interactive data processing while allowing for the execution of arbitrary snippets of code, making it much more flexible for doing data exploration, especially when more advanced data processing is required.

Second, many databases are meant to handle OLTP from multiple users. OLTP workloads have much different requirements that OLAP works loads, leading to a lot of unnecessary overhead in processing (such as a strong consistency model and locks). Additionally since RDMBS's are meant to handle multiple users they frequently don't take advantage of all the compute and memory resources available. PostgreSQL for example only recently added parallelism into query execution and doesn't use all available threads.

Finally, RDMBS's are built with the assumption that RAM is very limited and the data is very large, hence their memory foot print is very small even for large tables but vastly slower. SparkSQL on the other hand makes the opposite assumption, that there is plenty of RAM and hence persists everything in memory by default.

3.2 Coarsened Exact Matching [2]

A common problem in statistical analysis is that it is often not possible or too expensive to obtain experimental data for a subject of inter-

est. To address this issue, observational data is often used, due to its availability and price. Using observational data however doesn't allow for the control of confounding factors due to the lack of randomized assignment. Coarsened Exact Matching address this problem by performing a kind of fuzzy matching between the ćontrolánd t́reatmentǵroups in the data and then doing analysis based on this matching.

For our purposes, this is a principled way to control for confounding variables, by providing a way to group data and then compare from with in groups. For example, we could stratify our data points into those which come from areas with high crime rates and low crime rates and then perform linear regression on each group and see if there is a statistically

3.3 ZaliQL [3]

ZaliQL is a framework for doing causal inference on large observational datasets using PostgreSQL. To do this, they apply fuzzy matching between the control and treatment datasets using possible confounding variables and then do casual inference. The contribution of this paper is that this procedure (which is very common in statistical anaylsis) is implemented as a PostgreSQL package allowing it to scale to billions of observations.

4 Data

4.1 Sources

We found that there were many open sources of data provided to the public by the UK government such as . Accessing these data sources was as simple as opening a web browser and using their web interface to download data sets of interest. In fact, we found there to be an over abundance of data that we could access, making the main problem sifting through irrelevant or unusable data sets. Given our end goal of casual inference we applied the following criteria.

- Must be from the years 2010 to 2017 (the years which we have for the prescribing data)
- Must be for England and Wales
- there must be an intuitive idea as to why it would be related to drug prescribing trends in the UK
- Have geographic granularity which can be normalized and linked with the prescribing data

Applying this criteria (and a lot of googling and coffee) we extracted the datasets in table

4.2 Preprocessing

Our main data set that we extracted we estimated to be about 125GB uncompressed. At this size even running a three node cluster of cloudlab with 150GB of ram per node, preprocessing the data is no feasible due to Spark's memory usage. To deal with this issue we took advantage of the fact that the dataset was divided into zip file for each month. Hence we coarsened the data on a per month basis and then combined the results using pyspark. The rest of outer datasets were considerably smaller, however all of them required combining multiple tables. Hence we used python and Pandas to combine the data into one table and then normalize the data to the same geographic granularity as the prescribing dataset. Additionally we found that there this allowed use to quickly check for corner cases in the data. After we completed the preprocessing we were left with 4 datasets which had a spider web schema, that is, everything could be joined with everything else.

5 Methods

5.1 Identifying interactions of interest

Even with our fairly limited datasets, there is a huge number of different directions to look for interactions. Performing in depth analysis of all possible interactions computationally would be difficult and completely infeasible in terms of human effort. Not to mention the possible issues with statistical validity of testing massive

numbers of hypotheses. To address this challenge we applied a few techniques to try to reduce our search space and computational complexity. First, we needed to be able to have an easy to compare, interpret, and computationally cheap measure to identify interactions of interest. For this we used, the pearson's correlation coefficient. While this may not be a statistically valid way of quantifying relationships between our variables (due to possible differences in distributions), since we used it simply as a filter for where do more in depth analysis.

Even with a relatively cheap measure for where to look, given that there are over 38,000 unique BNF codes in our dataset, looking at each individual drug compared to even a

Even with a relatively cheap measure for where to look, given that there are over 38,000 unique BNF codes in our dataset, looking at each individual drug compared with a few casual variables would still take far too much time, hence we took advantage of the fact that BNF codes are order in a heirarchical manner to further coarsen the data. That is, each code is prefixed with a section number and sub section number which groups drugs which have similar use cases (e.g. treating hypertensive diseases), e.g. '0103' is part of section 1 sub section 3. We used this is group the data greatly reduce the computation required. We were surprised to find that even this was not sufficient as we still had issues with running out of memory on the cluster.

Finally, we simply choose to ignore some of the classes of drugs which are not ínteresting(e.g. not related to major health problems). This reduced our search space to a much more manageable size which we could handle with our computation resources.

5.2 Controlling for Confounding Variables

Doing any sort of reasonable causal inference for a complex phenomena like prescription drug use requires some method for controlling for confounding variables. The first method that we looked at was Coarsened Exact Matching (CEM). CEM is a way of stratifying data into groups such that comparisons within groups account for the potential confounding variables. For continuous variables (which all of our data is) CEM requires that the confounding variables be descritized and to do the grouping. Various methods exist for doing so include . We found that our data made doing this procedure difficult for a few reasons. First, we had at least seven possible confounding variables for any given interaction that we decided to investigate. If we were to stratify each variable into three categories, we would have 2187 different groups. With Postcode District granularity of our data we had 1974 unique geographic locations, and 6 years of data which we all of our datasets overlapped, that would mean we would expect to have 96 data points (at the month granularity) for each group, which is far too small. Furthermore, it isn't clear that three categories per confounding variable makes sense, it is likely that for some variables more strata makes much more sense.

Given these issues with doing CEM we opted for the simpler multiple linear regression using the 10% rule of thumb for judging whether there was a significant confounding effect. We used a basic linear model because while some of our variables were measurements that correspond to real world entities, most of our variables (i.e. the Indexes of Deprivation scores) are an abstract score which is meant to be compared within the category but not across. This being the case it was unclear what kind of model should be tested (e.g. is the relationship between crime and hypertension medication quadratic?) and allowing the model to be tweaked arbitrarily would almost guarantee spurious findings as we did not control for the number of hypotheses we tested.

6 Results

6.1 Distribution of Drugs

We first looked that the overall distribution of drugs being prescribed (by number of items). The first thing that stood out was that the histogram looked to be following a power law. Note that this for ALL drugs, not a particular

category. This means that there are a few drugs that account for a significant portion of the total number of drugs being prescribed for any given year. For space reasons we only include the plot for 2015, however the same pattern was found in all years.

Next we filtered the data to just include heart medication (BNF section 2) and created the same plot. We again note that the distribution looks very similar to a power law graph. Hence we performed a best fit on the histograms above using the formula $ax^b = y$ where x is the rank and y is the percentage of the items that the drug accounted for.

6.2 General Trends

Next we looked at the overall trends in drug prescribing over time. We again start by looking at the prescribing rate of all drugs.

Here we see a very clear increasing trends in the number of drugs being prescribed.

Next we looked at the drug prescribing trends for heart medications.

No surprise here, it roughly follows the overall trend.

We then wondered how this might compare to the trends in mortality from heart related causes.

We were quite surprised to find that there was a *decreasing* number of heart related deaths overall. This contrasts quite a bit to previous observations that we made while doing exploratory analysis which found that the norm was for there to be a positive correlation between the number of drugs prescribed to treat a illness and the number of deaths from that illness. For example, we observed this trend in drugs used to treat dementia.

6.3 Per drug trends

Given the interesting overall trend in heart medication we decided to look at trends on a per drug basis. Initially we simply graphed the drugs to see if there was anything abnormal. We immediately noticed that there were some quite dramatic increases and decreases in prescribing rate for some drugs. We then wondered how homogeneous these trends where. That is, does this trend hold at the Postcode District level? To measure this we took calculated the correlations between the prescribing of a particular drug with the overall trend. We choose to use correlations again for two reasons. First, the amount of computation to do this was quite large. Even after filtering out the drugs which had less than 1M items prescribed from 2010 to 2017 running the script took roughly 20 hours of CPU time. Second, while we could have performed linear regression on the time series and compared the slopes, we were interested in the any kind of deviation from the trend. For example if the overall trend looks like a sin wave and the trend for a particular district was out phase, we wanted to detect that as well differences in the slope of the fitted line (which would just be registered as a negative correlation). Our analysis found that for many of the drugs there was very little deviation from the overall trend, looking at the histogram of the correlations it looked like

However we also found some drugs where this was not the case at all.

We note here that many of the correlations plotted in the histograms had a p-value that was greater than .001 and hence wouldn't be statistically significant. However since we know that the two variables are in fact related (one is used in the calculation of the other) we included these.

6.4 Causal Inference

Our previous observations then (we think) beg the question why is there a difference in the trends? To try to answer this question we performed both simple linear regression and multiple linear regression to assess whether or not there was a significant relationship between drug prescribing and if that relationship was significantly confounded by other factors. Unfortunately, we found that there was significant confounding in all relationships we tested. This being the case it would be misleading to say that we have been able to make any sort of statement about causality in our data. We do note that we found that the proportion of the pop-

ulation that was 65 and old was the best predictor of the rate of prescribing for heart medication. This of course has a very straight forward interpretation that as people get older it is much more likely that they will have heart and circulatory problems, requiring medication. While this is not terribly surprising, it is useful in that population proportions are easy to predict and hence can be useful in predicting the demand for medications.

6.5 Confounding Variables and Correlations

Failing to draw any sort of causal relationships in the data, we endeavoured to explain why this was so difficult. Below we provide a table of correlations between our variables. We note that there are *very* significant correlations between may of the variables. In fact, finding not statistically significant correlations between our predictive variables was an exception, not the norm (see table 1). We believe this to be a major cause of our difficulties in the

7 Conclusion

7.1 Data Cleaning and Preprocessing

For the most part our datasets were quite clean and easy to handle. Despite this, preprocessing and integration still took significant amount of time and domain knowledge to execute. For this we found that for any particular dataset preprocessing was fairly simple but each dataset contained it's own edge cases that needed to be dealt with accordingly, which amounted to a lot of hard coding. Given that these scripts are ran at most a hand full of times, we found that the time taken to run the scripts was of very little concern and instead the time it takes to write the scripts were far more pressing. From we found that libraries like Pandas and PySpark to be invaluable due to their flexibility and ease of use. This is particularily true given that code reuse is difficult in the best case for these applications as anticipating was is going to be need in the future is difficult if not impossible. While our work flow and processes has been laid out in a sequential manner, the reality is that the processes is far from that. We found that we were gathering and cleaning new data up until constantly in parallel while doing analysis and running queries.

7.2 Rigorous Casual Inference

Finding statistically significant correlations/relationships in data has become easier than ever due to the ease with which massive amounts of data can be processed with only basic knowledge of computer programming and easy access to computing resources through cloud based platforms. With this power comes a different problem though. Without controlling for the number of hypotheses tested, p-values are misleading in many cases. Given enough computing resources and data, you will find *something* that is statistically significant. We attempt to control for confounding variables in our analysis however this isn't sufficient, as the power to test thousands of hypotheses still poses a major issue for

References

[1] Michael Armbrust et al. "Spark SQL: Relational Data Processing in Spark". In: SIGMOD '15 (2015), pp. 1383–1394. DOI: 10.1145/2723372.2742797. URL: http://doi.acm.org/10.1145/2723372.2742797.

[2] Stefano M. Iacus, Gary King, and Giuseppe Porro. "Causal Inference Without Balance Checking: Coarsened Exact Matching". In: *Political Analysis* 20.1 (2012), pp. 1–24. URL: https://www.cambridge.org/core/journals/political-analysis/article/causal-inference-without-balance-checking-coarsened-exact-matching/5ABCF5B3FC3089A87FD59CECBB3465C0.

[3] Babak Salimi et al. "ZaliQL: Causal Inference from Observational Data at Scale". In: *Proc. VLDB Endow.* 10.12 (Aug. 2017), pp. 1957–1960. ISSN: 2150-8097. DOI: 10.14778/3137765.3137818. URL: https://doi.org/10.14778/3137765.3137818.

	heart_related	crime	employment	health	housing	income	density	latitude	over_65	over_45
heart_related	1.000	-0.191	-0.076	-0.050	0.110	-0.121	-0.160	-0.055	0.259	0.239
crime	-0.191	1.000	**0.641**	0.595	0.007	**0.761**	0.598	0.030	**-0.660**	**-0.701**
employment	-0.076	**0.641**	1.000	**0.897**	-0.119	**0.898**	0.259	0.209	-0.268	-0.307
health	-0.050	0.595	**0.897**	1.000	-0.170	**0.820**	0.262	0.260	-0.282	-0.332
housing	0.110	0.007	-0.119	-0.170	1.000	0.108	0.253	-0.201	-0.119	-0.145
income	-0.121	**0.761**	**0.898**	**0.820**	0.108	1.000	0.537	0.132	-0.530	-0.592
density	-0.160	0.598	0.259	0.262	0.253	0.537	1.000	-0.057	**-0.626**	**-0.684**
latitude	-0.055	0.030	0.209	0.260	-0.201	0.132	-0.057	1.000	0.012	0.015
over_65	0.259	**-0.660**	-0.268	-0.282	-0.119	-0.530	**-0.626**	0.012	1.000	**0.962**
over_45	0.239	**-0.701**	-0.307	-0.332	-0.145	-0.592	**-0.684**	0.015	**0.962**	1.000

Table 1: The pearson's correaltion coefficient's between the independent variables