**Introduction**

This textual analysis is meant to serve as a case study for application to MDRs (medical device reports) submitted to CDRH (center for devices and radiological health), a center within the FDA. For this case study on Essure, a voluntary permanent sterilization device, I will explore the advantages and disadvantages of using text mining and text analysis methods for obtaining relevant information within over 20,000 MDRs. For each application, I will compare the utility and risk to the others. My main objective is to improve upon methods used to efficiently analyze a large number up reports, while also not losing valuable information that may also prove useful. In the long run, I feel that this method can be used to identify the problem faster, and start asking the right questions sooner. The choice to analyze Essure specifically is in no way meant as an attempt to “expose” or otherwise damage the brand. Though it has been met with public controversy, to comment on this is not the point of my analysis. Data is in the form of the event description text within each MDR, and graphs for visual aid will also be provided.

Specifically, an MDR is the report of an adverse event “submitted to the FDA to report any undesirable experience associated with the use of a device”. This includes serious side effects, product quality issues, and misuse. The FDA requires MDRs to be submitted by manufacturers, device user facilities, and importers, while other parties, such as health care professionals and patients, are strongly encouraged to do so[[1]](#footnote-1). Though the FDA recognizes the drawbacks of such data collection methods, it is but one of many data sources. Others include clinical trials, and post market surveillance. Despite its apparent downfalls, the reports combined with other data sources are major contributors to improvement of patient safety.

**Data Extraction and OpenFDA**

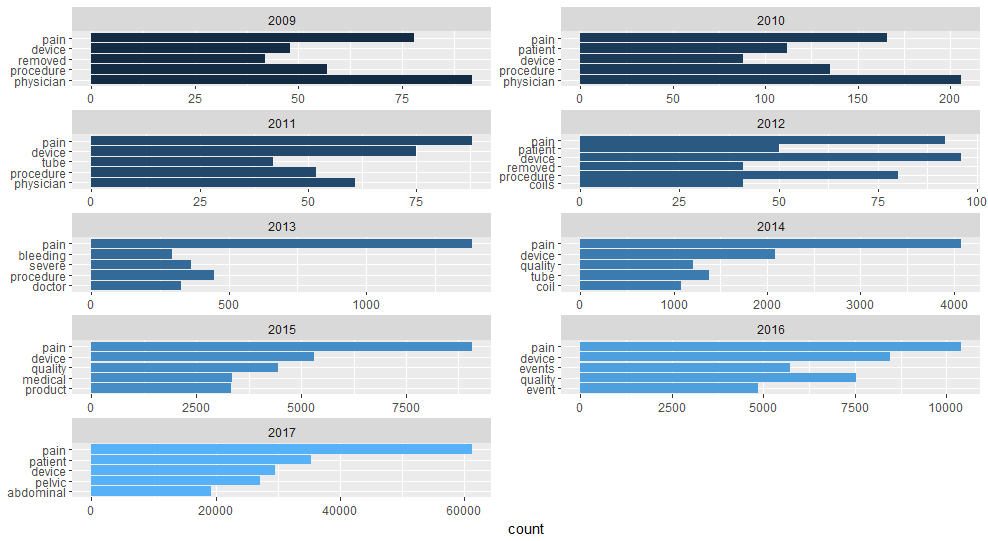
As does every FDA page regarding openFDA, I will also begin with the following disclaimer: “Do not rely on openFDA to make decisions regarding medical care. Always speak to your health provider about the risks and benefits of FDA-regulated products[[2]](#footnote-2).”

OpenFDA is a publicly available API created in 2013 with the goal to “create easy access to public data, create a new level of openness and accountability, ensure privacy and security of public FDA data, and ultimately to educate the public and save lives[[3]](#footnote-3).” It allows novice (like me) and expert data scientists alike the opportunity to apply their skills for information retrieval. I should not spend too much time explaining the data retrieval process, but since it took a significant amount of time learning and improving how I went about obtaining the text data, I feel it is worth briefly mentioning.

The method I used for obtaining MDR data was through using the GET command, found in the “httr” package in R. To obtain the desired url, you must find the API link[[4]](#footnote-4), and include your search parameters (search; limit; skip). One huge obstacle I faced was the fact that the FDA’s API only lists and allows download for 100 MDRs at a time, but I was looking to obtain at least 20,000 (out of the total 28,00 or so). In short, this required me to make use of the “skip” parameter, create a loop that skipped every 100 listings, and then extract the text/date data which I wanted from each MDR. From there, I could tiblify and tokenize the data as I needed.

**Viewing Simple Tokens**

Simple tokens can certainly come in handy as a way of getting familiar with content of a text. You can take the word counts and word frequencies (after taking out stop words) to reveal some common theme or get a general idea of what’s going on. This, however, I would argue, is not such a great application when it comes to MDRs:



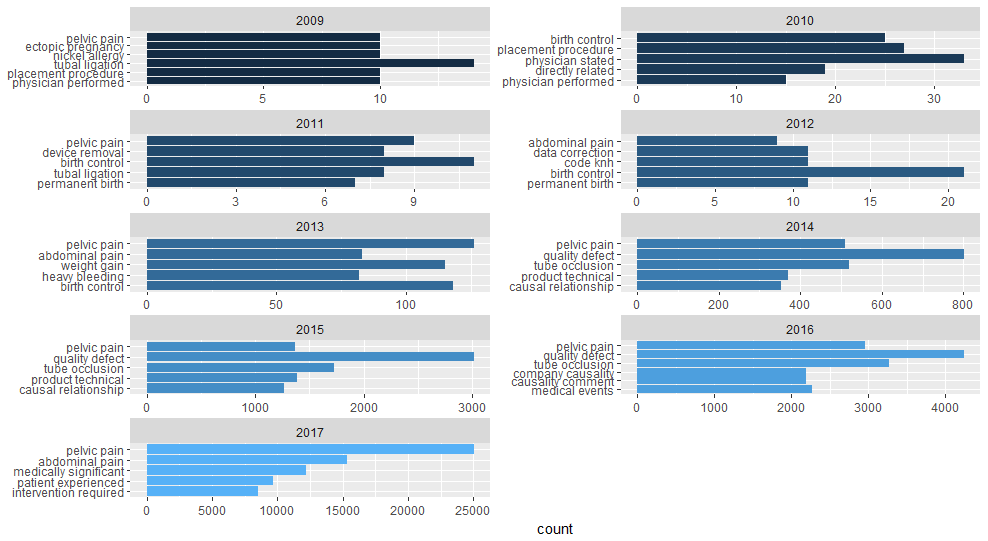
Of course, we can get *some* meaning from this: pain. There appears to be a lot of pain associated with the adverse events. This, however, is perhaps what we would expect from an *adverse* event. So, just knowing that there is pain involved, does not provide us with much to discuss if we wanted to start asking specific questions to people who might have answers, or, could give us leads. Simply put, it does not suffice to say, “patients experience a lot of *pain* in the adverse events”. If we can tell an expert what type of pain, that might be more useful.

One risky thing with naively taking the count of a unigram to analyze an MDR is how misleading one word could be. Let’s take the most serious case for example: death. By taking the raw count of how many times death appears, we can see that in November of 2017, there were 2,615 counts of the word death. This would be astonishing, and sad, if it actually meant that 2,615 people died in November alone from this device. Not only that, but I doubt this product would be on the market still if this were the case. In fact, if you were to plot the “frequency of death” over the years, which I did, you would see that it increases over time. Not to be alarmed, as the overwhelmingly vast majority of these MDRs actually say “no reported death”, or something to that effect. But what explains the increase? According to the FDA’s reporting, there was a flip in the majority from patients to manufacturers submitting the adverse event reports[[5]](#footnote-5), so it would make sense that we would see the lack of death being reported more often, as manufacturers likely want to emphasize this point.

Aside from the vagueness of pain, and the danger of death, we also see several other common words throughout the years- “device”, “patient”, “medical”, “bleeding”, “doctor”, etc. If I didn’t know better, I would likely be able to deduce what this report is about, and that, most likely, the situation did not pan out well for the patient. Nevertheless, this information does not help to achieve what MDRs are meant to do. By nature of the report, we already know something bad has happened, so it would be much more fruitful to know, for example: what sort of pain; bleeding from where; or what is it about the device we should know? Certainly, some, but perhaps not all, of these questions can be answered by going beyond the unigram.

**Going beyond the Unigram: Bigrams**

Knowing that simple unigrams don’t quite cut it when trying to extract important and valuable information from a medical device report, while they do have their place in the world, we could do much better just by adding a second word. The addition of this second word is how we not only get a bigram, but also get a much better understanding of what is going on with these people and their medical devices:



This already appears to be a much better representation of what the people submitting these adverse event reports are experiencing, and just by adding one word and taking the most common bigrams, we start to learn what all that pain was about. We can see that, practically across the board, pelvic pain is the most frequent side effect reported. Not too far behind, we also see that abdominal pain is frequently reported as well. One might criticize that this is not the most rigorous of analyses, and that may be the case, however I would also argue that the findings are, nevertheless, crucial and important. Over the years, these symptoms are widely reported and instead of sifting through thousands of reports, a bit of tidy text code can be applied in a relatively short amount of time to obtain this information. Beyond the top five that I’ve chosen to show, for the sake of keeping the graphs of reasonable size, the list certainly goes on.

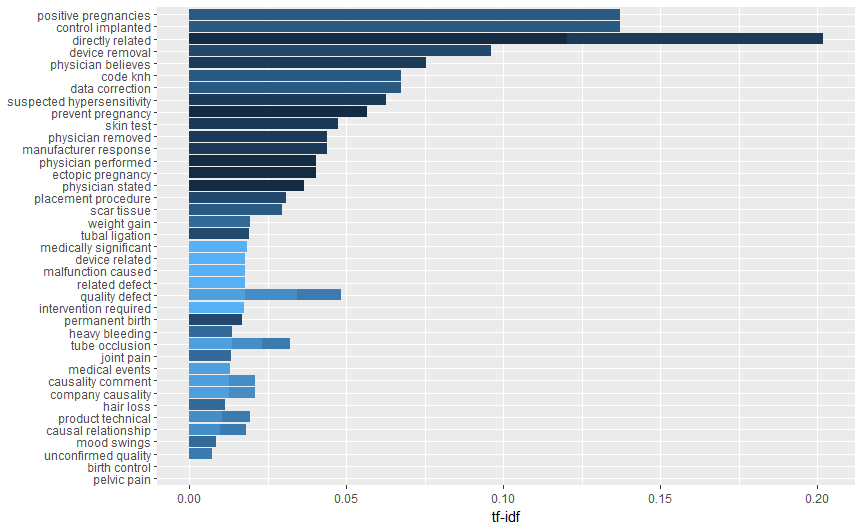
One other thing is clear though, and that’s the *noise.* Rather than random error, though, the noise I refer to are the most common words of “insignificant” meaning. For example, “physician performed”, “birth control”, “directly related”, and others, still do not help very much in terms of figuring out what is going on. But, like any method, this one is not flawless. If I really didn’t like the words I was seeing up at the top, I could easily go back to the code, and add words like “birth” to the stop word list (which I’ve done for several other words). The drawback here, as well as my point for not doing so, is that this word may be important in a different way. For example, perhaps there are many birth defects reported, but they just don’t appear on the top five, or top ten. I wouldn’t want to sacrifice this crucial finding just to clean up the top row, leaving only “meaningful” bigrams!

Now that we can note the most common reported side effects, we could still look at a few less common. Still, even in just the top five, we see a bit of a variety: “heavy bleeding”, “weight gain”, “ectopic pregnancy”, “quality defect”, “nickel allergy” for starters. With this information, now we might be able to strike up a conversation about how the body reacts to certain, in this case, metal implants, with the experts (immunologists, toxicologists, etc.). Also, as a regulatory agency, the FDA might have a word or two about the cause of the “quality defect” with the manufacturers. Of course, accidents happen, and some rare cases are anomalies.

Another, and perhaps the most significant, drawback of this, is that a full sentence could still read “there was absolutely NO heavy bleeding”, and yet we are left with just a bigram. Therefore, such a method should not be taken at face value, and we should always dive deeper to find out the whole situation (in this case, find the whole report, and read it.) There are methods of combating this drawback, but to go over everything is beyond the scope of this project. One simple, yet perhaps not ideal way, would be to add on several more “ngrams”. Though no cause for immediate alarm, these bigrams certainly give us reason for further investigation.

**Applying TF IDF**

TF IDF, short for term-frequency inverse document-frequency, is the product of the term-frequency and inverse document-frequency, where we take the log of the inverse document-frequency[[6]](#footnote-6). I will not go into detail about the mechanics, but what it essentially does is gives more weight to less frequent words, and significantly decreases the weight of super common words, like the words in my previous top five bigram list. The idea is that a word that isn’t so frequent is not necessarily unimportant, so the method tries to find the balance between common, and not *too* common. I believe this is very much applicable to adverse event reports, as there may still be some important adverse symptoms, but they are bogged down by the high frequency of the most common ones.



Now we are seeing a few new terms that, while not the most frequent, are still important for information gathering purposes. A few new terms include “joint pain”, “mood swings”, “suspected hypersensitivity”, and “positive pregnancy”. For a device, which is supposed to ensure permanent sterilization, we know there is something wrong when the user is still getting pregnant. With the additional insight brought about by applying TF IDF, we are better equipped to consult our proverbial “experts”.

An additional object of interest to point out is “pelvic pain”, which, as we’ve previously seen, is absolutely one of our most common symptoms. It now has a whopping value of 0. This is the effect that the TF IDF weight has on ultra-frequent terms. Remember that TF IDF is the *product* of term-frequency and the logged inverse term-frequency. Since we can assume that pelvic pain is appearing in all adverse events, we have an inverse document-frequency of N/n = 1, the log of which is 0. Does this pose a problem?

Much how the pure term count brings down other important insights, using TF IDF may also weigh down equally, if not more, important terms, essentially having the opposite effect. In my mind, this is not a major issue, as long as the analyzer is not limiting their self to one approach. What is still a real issue is the fact that we are dealing with bigrams, which are just as subject to being preceded with a negating term, such as “no” or “not”. Once again, there are ways to combat this drawback, but it goes beyond the scope of this project. That said, I fully intend to implement such ways in the future. With what I’ve done here, I already see large improvements from what I’ve experience sifting through reports in excel sheets.

**Conclusion**

These methods will not uncover everything, of course, and even with more advanced applications, I would still emphasize the need to take a closer look at the reports. Finding the most common ngrams, frequencies, and using TF IDF weights are what is supposed to get the conversation, and perhaps investigation, started. Taking the results for face value is dangerous, particularly when dealing with simple unigram tokens, as we’ve seen.

Aside from the innate drawbacks of the methods discussed, there is still a question of data quality. Medical device reports are not, and were never meant to be, the sole indicator of a problem or symptom. We use this data to supplement more rigorous and time-tested methods, such as clinical trials and post-market surveillance. Despite it’s questionable quality for drawing conclusions, it can certainly be useful for information gathering purposes, and allows the public to notify the FDA how adverse events from medical devices are affecting them. An improved method, of which I hope to aid in the development, may be able to improve the usefulness and quality of this data[[7]](#footnote-7). Tidy Text, and a bit of Data Camp on APIs, has helped me to make that attempt. The work is certainly not yet “complete”.

1. https://www.fda.gov/MedicalDevices/Safety/ReportaProblem/default.htm [↑](#footnote-ref-1)
2. https://open.fda.gov/about/ [↑](#footnote-ref-2)
3. https://open.fda.gov/about/ [↑](#footnote-ref-3)
4. https://api.fda.gov/drug/event.json?search=essure [↑](#footnote-ref-4)
5. https://www.fda.gov/MedicalDevices/ProductsandMedicalProcedures/ImplantsandProsthetics/EssurePermanentBirthControl/ucm452254.htm [↑](#footnote-ref-5)
6. tfidf.com [↑](#footnote-ref-6)
7. There are lots of other graphs and plots, but I saved those for the presentation! [↑](#footnote-ref-7)