PhUSE 2017

Paper PD07

Dipping our fingers into the world of data representation - a visual journey using Encapsia™ Clinical Data Suite

Minola Ilica, CMED, Timisoara, Romania

ABSTRACT

The technical advances of the past two decades have brought a significant change in the way the world looks at information. Where previously we would associate data with extensively manual and/or fixed point statistically programmed reviews, the increasingly graphical gadgets used in our daily lives have introduced a need for a more interactive approach to data review.

Encapsia™ Analytics provides live visualizations of the clinical data based on statistical techniques in order to identify unusual data and patterns. Statistical programmers have now been allowed the opportunity of using their background to apply new techniques on real time data to support cleaning and monitoring activities, while also being exposed to numerous new technologies.

The purpose of this paper is to look back on the various aspects entailed by transitioning from traditional statistical reporting into the new role of Data Programmer while exploring the capabilities of data visualization techniques using Encapsia™.

INTRODUCTION

Visual representations of data may not be a new concept generally and we are quite used to being exposed to increasingly visual experiences in our day to day environment. Whether it is in a cinema, in front of our TV sets, using our smart phones or tablets we are daily bombarded with visual content as various studies have shown that visuals are better processed by our brains and marketing research has shown that we are more likely to respond to images rather than plain text.

In this era of ever growing exposure to information, the pharmaceutical industry has so far remained quite reliant on massive data listings and tabulation. The capabilities and advantages of a more graphical approach have only recently started being explored.

Encapsia™ Analytics is looking to provide a faster way of understanding data and identifying patterns via real time visual representations that the Data Programmers would help configure using web programming concepts.

Finding programmers with experience in web design and visualization techniques is not a considerable challenge in today's world, especially since technical knowledge we are looking for with Insights programming does not necessarily surpass the usual requirements for web programming related positions. However the responsibilities entailed by more senior Data Programming roles do require a good knowledge of the clinical trial process and of the clinical data which can only be found within other areas the Clinical Research field like Biostatistics, Database Programmers, etc. This offers the unique opportunity of very specialized programmers working in niche areas of programming to branch back out to the more traditional programming world without effects to their seniority.

BACKGROUND

Whilst this year has marked the beginning of the first real study using all the capabilities of Encapsia™ as a standalone database and the birth of the Data Programming department at Cmed, the concept of Insights is not new to the Biostatistics department. The first contact I had with this concept was a couple of years ago, when we were asked to pool together some ideas regarding how statistical methods could be used to identify issues during study conduct, including site performance, data quality, safety review, remote based monitoring and even potential fraud. A few months later a team consisting of a biostatistician and myself, a statistical programmer, was working on the first demo version on what would later become the Insights module, proving that statistical background can be useful in designing new techniques to support cleaning and monitoring on real time data within the clinical database. But while the statistics department was helping expand the views on data review, the developmental nature of the work also exposed them to numerous new technologies and programming languages.

DESIGN VISUALIZATIONS

As most types of charts and graphs are based on descriptive statistical methods and as they have been used occasionally as part of the statistical analysis it was only natural that the Biostatistics group was involved with the design and scope of the initial Insights.

Clinical trial data can be very specific, particularly when dealing with oncology or rare disease studies, and presenting aggregate data can become difficult when, for example, each patient has a different visit schedule. While using the study day relative to dose for reports across tine might seem like a natural approach if one is familiar with statistical reporting, what I have found is that for team members coming from other backgrounds it was less obvious why this approach would be so much more correct than just using the visit number.

However unlike the statistical analysis, where the reports would have been used to highlight the study outcomes and results at specific points during study conduct (Interim of Final Analysis), with Insights the graphical representations are meant for more various audiences and purposes (medical review, data cleaning, risk based monitoring, etc.) and are meant to be used at any time during the study.

The real-time nature of the Insights implies the need to be correct on an ongoing basis. More like with DSMB type deliveries, design cannot rely on how clean data should look, but rather what data can look like at any point in time based on CRF structure and restrictions. Further issues might be caused with reports being run an all patients rather than different analysis populations. For the example above where results would be summarized across time relative to dose, the report cannot be restricted to Safety Population as patients that have yet to be dosed should not be excluded from data cleaning. Rather than focusing on what the data should be, we now need to learn how tailor reports to whatever the data can be, while also finding ways to visually highlight where the data does not fall in the first category.

In order to provide appropriate visual representation of the data that both contains all the necessary information but also uses methods that are intuitive enough for each role to understand, the data programmer is expected to liaise with the intended recipients to understand what the purpose and scope of each reviewer is. Sometimes this may be as simple as adding a footnote to a current design to clarify what is presented, similarly as would have been done in TFL Shells, however at times the entire output might need to be redone– for example, while for someone with statistical background a boxplot might seem like the most reliable method of spotting unusual data, it is like that medical personal would be much more fond of scatter plots.

Traditionally different stakeholders on Clinical Trials have been quite encapsulated in terms of what their role entail a and with Data Visualizations being a quite a new development within the industry there is an expectation for a certain degree of back and forward between the data programmer and different roles across study. However with more studies and more types of reports being worked on it is likely that the requirements and expertise of the Data Programmer role will need to expand to a good enough understanding of the overall high-level Clinical Trial process in order to diminish the dependence on study team feedback.

PROGRAMMING LANGUAGES

Encapsia™ Analytics module is quite flexible in the sense that it allows programming of study visualizations to use a large range of languages including but not limited to: Python®, R®, JavaScript®, HTML®, SQL®, Jinja®. A study visualization would frequently make use of a multiple of these to define the required behavior, often making use of various external templates available for use online (Google Charts® for example).

Generally in technology related areas it is very common to switch between programming languages and between industries in general (the same company can potentially service both entertainment media while also providing database frameworks for banks). Statistical Programming in comparison is less about programming and more about clinical data knowledge and therefore it is not unusual for statistical programmers to work within the same industry (pharma) and, due to its the highly regulated nature, use the same programming language (SAS®) for most of their careers.

Probably the biggest technical challenge I faced, in expanding my role, was moving back from five years in SAS to different programming environments that I had previously known and loved a number of years a ago while in college. What I have found when initially learning SAS after previously being used to programming languages like C or Java is that it is conceptually different to any programming languages out there. While Python, R, Java, Java Script may have considerable different syntaxes, they are all object oriented and based on the same principles which makes them easier to switch between. A few of the basic concepts you would come across for all four of these are variables, objects, classes, functions, function packages, inheritance, encapsulation etc.

SAS on the other hand is quite a high level programming language closest to SQL than any of the lower level programming languages mentioned above. The most important concepts for SAS are data steps and PROCs. A variable represents a column in a datagroup rather than referencing an individual data point. One distinguishing feature for SAS is that it does not allow the definition of functions. Similar functionality can be achieved by defining macros and using macro code, which in fact can be quite strong as it can both be used to mimic function behavior but can also be used as a sort of 'code generator' to allow code to dynamically be created based on the data. The advantage of a high level programming language is that is easy to learn which makes it easy to switch to for both programmers used to different languages (like myself when I started within statistics) and individuals without any programming background. The downside of this, along with the very specific syntax, makes switching from SAS to a different language can be quite tedious especially from the latter category. In my case, although I was surprised of how much I could forget in the time that has passed since college, I think I probably had it easier in the sense that I simply had to remember some of the concepts (like classes for example) rather than learn them from scratch. Particularly due to the mix of technologies and techniques used by my current role, I expect it will take a bit more time before I can call myself "fluent" in all the programming languages. However I did find that the internet holds a lot more examples and tutorials that I can use than what I was used to finding for SAS.

On a final note, I have to mention that to my personal discontent, what does seem to be consistent with every programming language out there, including SAS, is the focus on indentation and use of comments. Indentation is especially important with Python, where it is actually used to delimit blocks of code.

SCOPE OF DELIVERY AND VALIDATION

Another consequence of working on real data reports is that a change of mindset is needed in terms of what the actual "delivery" is and how this needs to be validated. With statistical deliveries the end products are the Table, Figures and Listings, therefore the validation is focused on checking the outputs are correct for that specific delivery. As live data visualizations are meant to be run at any time during the study, rather than at specific times when the data is expected to be clean, focus shifts on checking that the program used for obtaining the reports is correct and would also yield an expected result regardless of the state of the data. To that end, test cases need to be thought of in such a way to anticipate where issues might arise and to ensure the behavior is as expected regardless of the data entered (any value that can be entered in the database is appropriately handled).
Similarly, documentation is no longer focused on the accuracy of each delivery, but towards the quality and robustness of the actual report code. We are essentially moving from validating a result to testing an application in a similar way that would be done in a testing department of any technology company.
To put it in simpler words, when testing a report it is my opinion that one should essentially identify all the ways that could potentially break it and ensure they don't. Unfortunately what I have found to be the immediate consequence of this is that validation time is significantly higher than what was originally estimated due to the need to enter test data for specific scenarios. Therefore one challenge I think we might currently be facing within the group is identifying the proper timing of the reviews.

PROGRAMMING ENVIRONMENT

As data visualizations are designed to be ran directly from the user interface, they are essentially a part of the database itself. In order for the reports to be available for real time use, all the programs need to be installed directly on the database server. In order to copy configuration items from one server to another (as each project would have multiple servers like the configuration server, the User Acceptance Testing server and the live server) we use Trial Installation Packages (TIP). And in order to ensure traceability at all times we use versioning control to store program versions to a repository. Both the TIP and versioning are done via command line and require minimal knowledge of versioning systems and servers. Version control and management of TIPs would be part of the role of the Data Programmer.

Within my experience as a Statistical Programming I have had contact with version control, however this is has usually been less technical and did not involve direct contact with servers. Similarly, my experience with working directly on the command line within my former role was rather minimal. Although the process became straight forward after using it for a few times, the initial contact was quite challenging and eventually lead to an increase in skills and exposure to more technical fields that would otherwise only have be acquired by moving through multiple companies and over a larger period of time.

STANDARDIZATION VERSUS CUSTOM REPORTS

Similarly to Statistical Programming where a great area of interest is the standardization of programs and data submission, Insights developers will also be focusing their future efforts in trying to develop a standard library of visualizations that can subsequently be used on a large number of studies with minimal customization. The shift towards CDISC standards in data collections is a major benefit to this area as naming conventions are standardized across studies, and we can expect general domains like Adverse Events, Concomitant Medications for example to be very similar. However where collection guidelines may help with the review of safety domains, the visualizations required for review can be very diverse depending on the study drug indication or trial designs, particularly since the visualizations are meant to be multi-purpose and to satisfy the needs of multiple players (Data Management, Medical Review, Performance Review, etc.).
Regardless of how many or how well thought out a standard library might be, we can never expect to be able to satisfy every need that may arise with any clinical trial. That should not stop us however to try to minimize study specific work. A challenge will be to identify areas of interest that are less likely to be different across studies so that we can ensure resources are invested most efficiently (a suite to analyze the fluctuation of numeric measurements and results within and between subjects can be useful for medical review but also for data cleaning). On the opposite end, a specialized chart allowing the review of parameters involved with the disease response on an oncology trial might be the central item on that particular trial, however it may be that it will be used only on this one trial.

CONCLUSION

Data visualization in general is quite a new concept within the very conservative pharmaceutical industry and current uses are usually restricted to data cleaning and review activities. However, with the recent advancements in artificial intelligence and big data analytics it is difficult to say how far visualizations can go. There are many justified reasons why these new techniques should not be used for statistical reporting or any decision making process, but these might be surpassed in the future. With artificial neuron networks being used for support systems for medical

diagnosis it is no longer impossible to imagine a world where similar tools could be used in running and analyzing clinical trial data.

While it is difficult to say how, when, or even if artificial intelligence will play a part in clinical trial data analysis, we can say with certainty that due to the particularities of the pharmaceutical studies, any such change will involve personnel working on more traditional fields within the industry.

It is my opinion that data visualizations in their current form are only a very shy first step towards this shift. With a mixed background of artificial intelligence and automation in college, followed by working in clinical biostatistics I feel that this new role offers me the opportunity to both learn more about clinical trials while also developing on the a developing on the technical side, allowing me to potentially move on to other industries.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Author Name: Minola Ilica
Company: Cmed SRL
Address: Str. Coriolan Brediceanu, Nr. 10, City Business Centre Building A, Second Floor
City / Postcode: Timisoara, Timis, 300011, Romania
Work Phone: +40 (0)356 731 007
Fax: F: +40 (0)356 004 364
Email: milica@cmedresearch.com
Web: https://www.cmedresearch.com/