Utilizing D3: creating interactive data templates for experimental and computational statistics and results for MINERvA

Sheri Lopez
University of New Mexico-Los Alamos

Science Undergraduate Laboratory Internship (SULI)
Fermi National Accelerator Laboratory
Batavia, Illinois 60510

Abstract

Physics today is a consortium of data that has been collected, analyzed, and distributed among the masses. However, the graphical representation of this data is a bombardment of numbers, lines, and error bars displayed in a singular graph. Utilizing this approach to the distribution of results can be difficult to read quickly even for the most seasoned physicists, and beyond confusing to the layman. However, there have been advancements in code that are able to use and manipulate data in simple and straightforward graphs that are able to separate data and values into animated graphs such as D3. Utilizing D3 can ease the display of multiple data sets, which are related, by showing them on one graph that is animated to transition between all the points. The MINERvA group will be utilizing a D3 library built and loaded onto GitHub in order to animate and display the data in an interactive and relatable interface for users of all levels. Using this animated graphics library will create a more user-friendly interface for the MINERvA group that will be capable of explaining physics and experimental results to, not only, the science community, but to anyone interested in neutrino physics.

I. Introduction

An abundance of science today, particularly physics, is based on results of experimentation. Oftentimes, these results are relayed to an audience in graphical form, which displays statistics, errors, the correlation between the two, and expected outcomes. While delivering these results, it is crucial that everyone being exposed can understand and retain the information being given to them. However, the projected statistics and outcomes are usually displayed in vast amounts and in a singular graph, which can be overwhelming to the audience, and lead to the loss of information retention. The misunderstanding of valuable statistical analysis can result in losses for physics in the form of funding, interest, and at the very worst, important data. However, there are ways to improve the delivery of results.

In order to break through informational boundaries and bring a greater understanding of science, efforts are being made to create more user-friendly networks of experimental data. There are a host of graphical code-modules being designed over the Internet. To date, there are approximately eight different code types which can ease the scrutiny of overwhelming data. These include, but are not limited to, the following: Data Wrapper, Flot, Google Chart Tools, gRaphael, and HighchartsJS. As comprehensive as these new code designs are, they still lack the ability to properly correlate statistical data in an orderly and comprehensive fashion. Yet, there has been a recent development in coding that is elegant, interactive, and comprehensive, and that code is D3.

The D3 library is a combination of SVG (Segmented Vector Graphics), JavaScript, HTML (HyperText Markup Language), XML (Extensible Markup Language), and CSS (Cascading Style Sheet), which are all utilized in correlation with each other to create transitional data that is formatted in a style that is clean and easy to read. Since D3 utilizes JavaScript, the SVGs, such as circles and lines, are easily manipulated and graphed with tooltips that are highly functional and easy to read. The removal of multiple statistical points creates a cleaner graph that everyone is able to read clearly. The removed data can then be placed back into the graph in the form of tooltips (pop-up boxes) that correlate to individual data points and carry all the information in a discrete manner. D3 is also a perfectly acceptable data code that can be utilized by the scientific community to ease the viewer into the complex results the experiments or simulations often yield. D3's graphing ability has been gaining recognition since publication in 2013, and has recently caught the attention of the Main INjector ExpeRiment for v-A, or the MINERvA group.[1]

The scope of this paper will discuss the creation and utilization of a D3 library that can be distributed and used by the MINERvA group. The topics covered will explain why D3 is an advantage to the physics community and especially to the MINERvA group.

II. MINERvA

The MINERvA experiment focuses on neutrinos, their oscillations, and their interactions within heavy nuclei, such as lead, in order to determine how neutrinos interact with our universe. MINERvA is the "first of its kind in the world to use a high-intensity beam to study neutrino reactions with nuclei of five different target materials, creating the first side-by-side comparison of interactions."[1]

In order to conduct an experiment such as MINERvA, computer simulations calculate the statistical results that physicists are looking for prior to the experiment being run. These simulations expel immense amounts of computational data that help to form the hypothetical measurements for what the experiment will actually encounter. This data is then used and weighted against the actual experimental data that is collected. When the two data sets are compared, the statistical data and the actual data are usually displayed in a singular graph, which can appear very convoluted to the audience. Figures 1 and 2 show an example of statistical data that has errors and experimental measurements included.

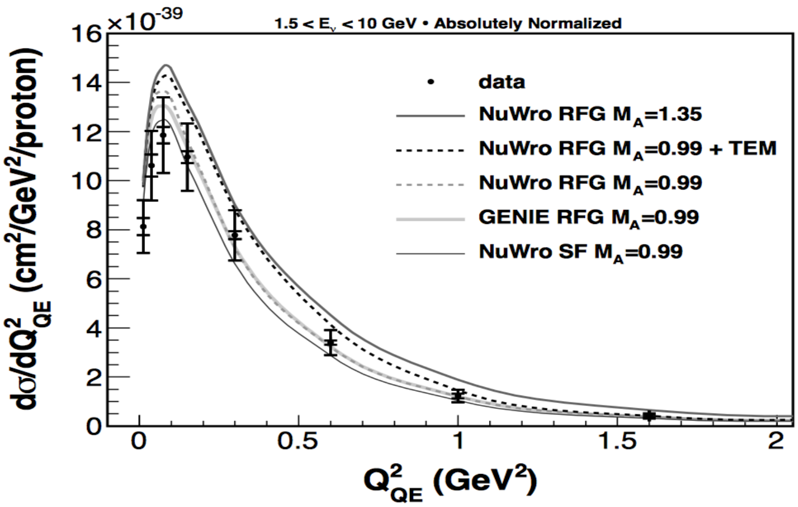


FIG. 1. MINERvA: measurement ofmuon antineutrino quasi-elastic scattering on hydrocarbon target. [2]

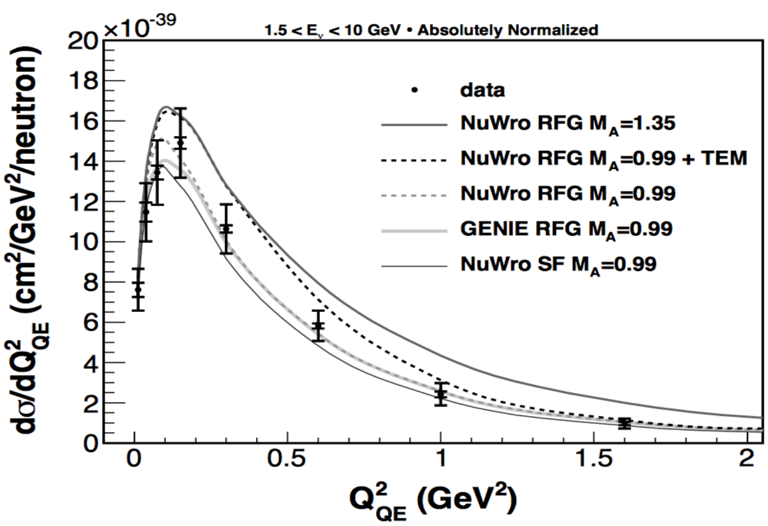


FIG. 2. Meausrement of muon neutrino quasi-elastic scattering on hydrocarbon target [3]

Graphs like these can be extremely difficult to read to the untrained eye. Yet, with the aid of D3, the MINERvA group can display their results in an orderly and more manageable fashion.

III. D3 Code

Learning and utilizing the D3 code will become a valuable tool for the MINERvA group since the code can create display results, such as the examples given above in a clean, streamline, and presentable format. When utilized correctly, the D3 data library can create beautiful and transitional graphs, animations, and information that can captivate an audience, hold their attention, maintain the ability of the audience to retain information, and bring more excitement to the physics field.

Since D3 maintains the ability to manipulate individual segmented values, the presentation of data can show only what is needed and then transition into other specific points of data that must be represented. All of this can occur in one individual, animated graph. The ability of D3 to make this happen occurs because of the code's ability to manipulate many different coding types into one coherent design that is able to carry information better than regular HTML or JavaScript alone.

D3 works by utilizing HTML to build the basic web document, XML to carry the messages, SVG to create the lines, bars, and dots that are the graph overall, CSS to stylize and color the graph, and finally JavaScript to animate, manipulate and tie the different data values together. Yet, in order to fully graph how D3 works, we have to explore the individual components involved in the code itself.

A. HTML

HyperText Markup Language, more commonly known as HTML, is a code that is used in basic web design, and is often identifiable by its tags or nodes (<>, </>).

Example:
```
<!DOCTYPE html>
<head>
    <title></title>
</head>
</html>
```

HTML has been around since 1989 and was developed at CERN[4] by Tim Berners-Lee, so scientists could communicate results in a swift manner. Since then, HTML has morphed into a standard used to create all webpage designs.

Without HMTL a webpage would have no structure in which to add elements, divs, styles or paragraphs. Most of the use of HTML in the D3 code is to ensure that there is a proper frame to hold the rest of a web-based code in place such as JavaScript, CSS, and SVG.

B. XML

XML is the code that is found between the nodes of HTML. Oftentimes XML are message carriers that are only viewable on the server side of a code, meaning the customer cannot see any of the XML on a webpage. XML works by relaying information from one set of nodes to another. Within the HTML tags XML can also assign classes and ids to specific sets of information that can later be called using CSS. Using the example above, XML are the words "head," "title," and "body" that are written between the HTML nodes.

Example: <title></title> where XML=title

C. SVG

SVG or Scalable Vector Graphics is the code implemented in D3 to create lines, triangles, circles and any other shape to be manipulated in the graph. SVG are oftentimes created within HTML nodes or can be manipulated using JavaScript. SVG's role in D3 is to create the design elements that will ultimately make a graph such as the axis, data points, and labels.

Example: <svg width= "100" height = "100">
<circle cx= "50" cy= "50" r= "40" stroke= "green" stroke width= "4" fill= "yellow"/>
</svg>

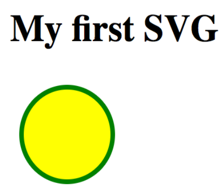


FIG. 3. The coded SVG element as it appears on a webpage.

D. CSS

CSS, or Cascading Style Sheet, is the code that is used to stylize a webpage, add a background, color text, borders, or background, and any other elements. CSS uses the call functions given by XML such as "class" and "id" and stylizes the specific class or id called. For example: If we have the following,

```
<div class= "axis"></div>
```

we can use our CSS to call the class axis to then stylize the div. The code to do this would look as follows:

```
.axis {
        fill: red;
        stroke: black;
      }
```

When our axis is stylized according to the call, we will have a red axis line with a black outline. CSS is capable of doing this for every element within a webpage and is key in D3 because it will ultimately color and move the elements in our webpage around the page itself.

E. JavaScript

JavaScript is a language that is used in conjunction with HTML to create interactivity in webpages. JavaScript allows for client-side script trough which the user can interact, such as with games or pop-up windows. JavaScript is identifiable by its use of loops, functions, and variables. In D3, JavaScript is the key component animating the SVG and tying in transitions between two sets of data or multiple SVG elements.

Example:
```
var speed = 65;
// Complete the condition in the ()s on line 4
if (speed>80) {
// Use console.log() to print "Slow down"
console.log("Slow down")
} else {
// Use console.log() to print "Drive safe"
console.log("Drive safe")
}
```

IV. Utilizing the Code to Create Animated and Transitional Graphs

Utilizing all of the codes stated, a library was built for the MINERvA group utilizing results taken from the MINERvA homepage. Figures 1 and 2 are the graphs I replicated using D3. I managed to combine the two graphs into one. Using a transition and radio button, the user is able to easily switch between the antineutrino quasi-elastic scattering results and the neutrino quasi-elastic scattering results. Below are previews of the graphs without animations.

FIG. 4. Graph utilizing MINERvA data pre-transition.

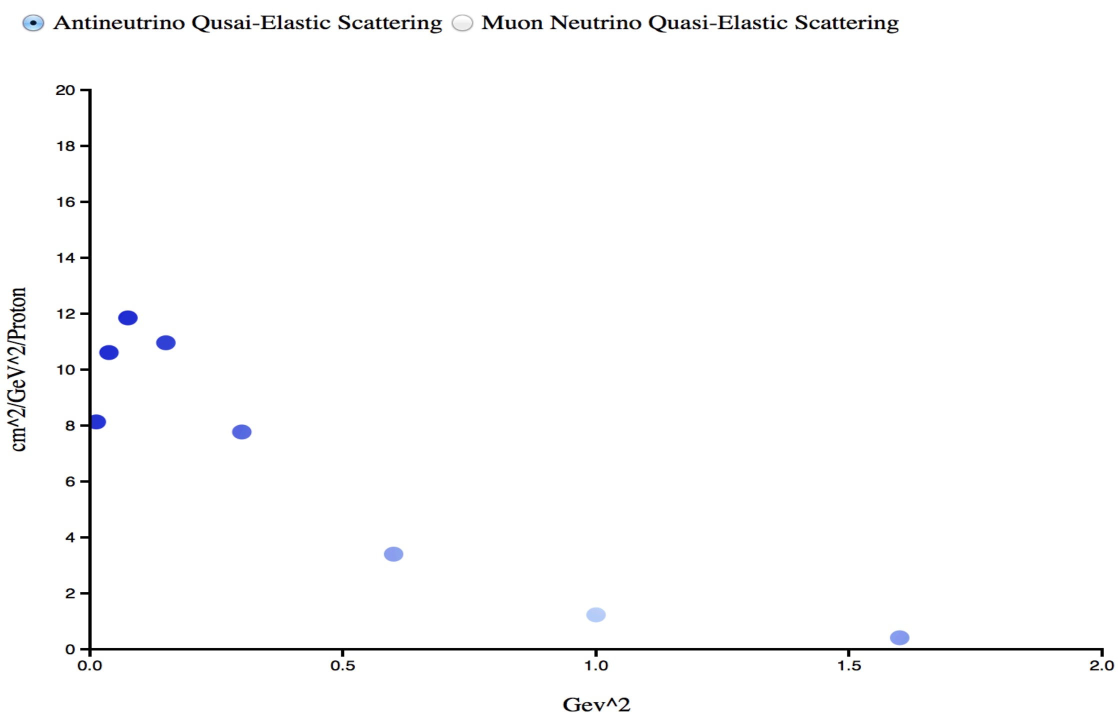
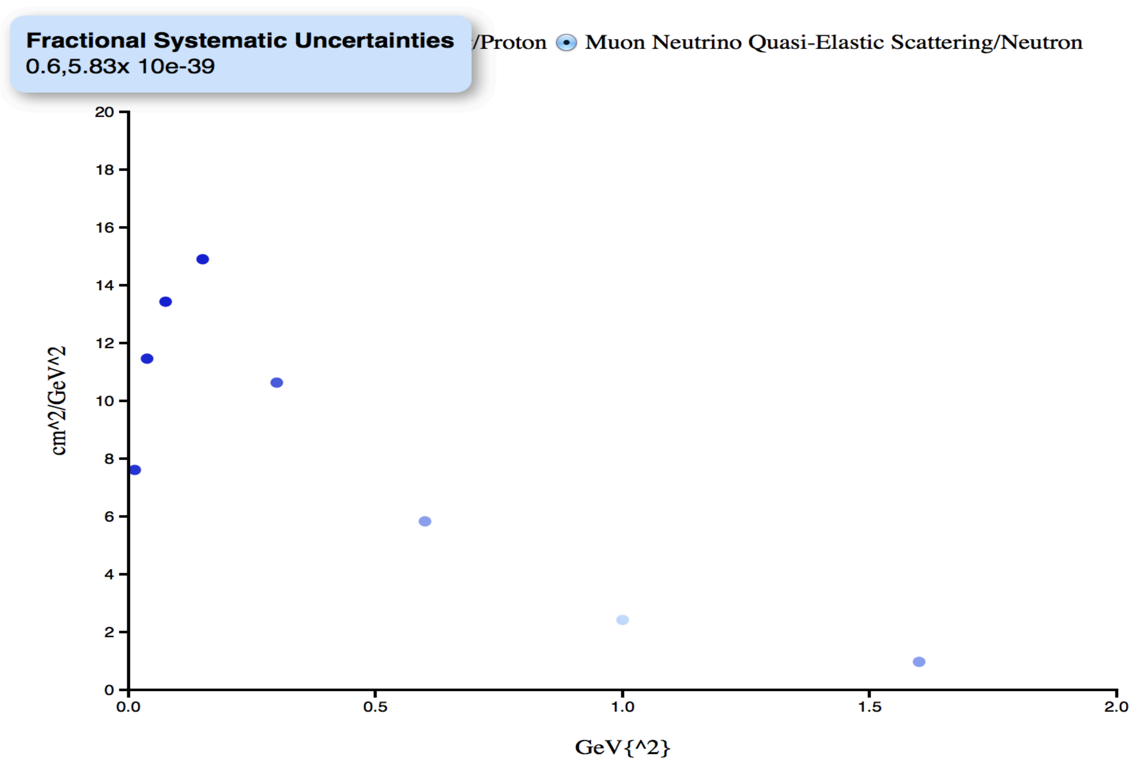


FIG. 5. Graph using MINERvA data post-transition.



Figures 4 and 5, which detail the statistical findings of quasi-elastic nucleic scattering, are cleaner and more efficient than the previous two plots (Figures 1 and 2). The radio buttons make it very evident, which set of data is being previewed without having to clutter the graph with a title page, and there are tooltips (as shown in Figure 4) that appear when a data point is moused-over. The advantage of the tooltip is its ability to carry other necessary information discreetly and link it to the individual point that the information belongs too. This in turn clears up the need for multiple data plots such as these matrices since they are combined into the tooltip data:

FIG. 6. Fractional systematic uncertainties on $d\sigma / dQ^2_{QE}$

Q^2_{QE} (GeV^2)	I	II	III	IV	V	VI	Total
0.0 − 0.025	0.06	0.04	0.02	0.04	0.09	0.03	0.13
0.025 − 0.05	0.06	0.03	0.02	0.03	0.09	0.02	0.12
0.05 − 0.1	0.06	0.03	0.02	0.03	0.09	0.02	0.12
0.1 − 0.2	0.06	0.03	0.03	0.02	0.09	0.02	0.11
0.2 − 0.4	0.05	0.02	0.03	0.03	0.09	0.01	0.11
0.4 − 0.8	0.05	0.03	0.04	0.04	0.09	0.01	0.13
0.8 − 1.2	0.08	0.07	0.07	0.15	0.09	0.02	0.22
1.2 − 2.0	0.12	0.07	0.07	0.16	0.09	0.02	0.24

D3 clearly is capable of interlacing many different types of data into an elegant and confined space. D3 is also capable of adding flair to a graph by using transitions as well as tooltips, which are also functional in carrying information from data point to data point, which is why the code library has been built.

V. Building the Templates for Experimental Data from MINERvA Results

The examples above were created using D3 and the templates for the graphs will be distributed to the MINERvA group through GitHub. GitHub is a code-sharing website based on open source developments with many different code types, including C++, R, and D3; it is here that the templates for the graphs will be saved for future use by the MINERvA group.

In order to utilize the templates, a MINERvA member will merely have to log onto GitHub, open an account and select the template needed. From here the member will only have to copy and paste the code into the code editor of ones choosing. When this is complete, only the *var dataset (*the JavaScript array), which is the key element that holds the coordinates of the points, has to be replaced. When this is complete, it is only a matter of linking the new data results into the homepage for MINERvA or on a personal page for a thesis.

VI. Drawing the Public into What We Do with Interactive Data as Well as Web Design

The use of D3 on the MINERvA homepage will set the group apart from the rest. It will also draw more attention to the page, and hopefully to neutrino science in general. Using such interactive designs, like those made with D3, science can begin to draw the interest of the public again. If we can continue making results, statistics, and events readily available to the public, in a format that they can understand, there is a chance that we can introduce many people to the wonderful work of physics. Working on this project has been a small demonstration of how this can be accomplished. And to think, it is only one small graph that is this exciting. Imagine what we can do with animations of neutrino oscillation, or even more data made simple and understandable. With time, D3 can bring a broader audience into modern physics and explain findings in an easily relatable way. This can only lead to more information retention by the audience, bring in a more diverse audience over time, and ultimately graph the attention of the public, which can increase more funding for physics.

VII. Acknowledgements

I would like to thank Erik Ramberg, Roger Dixon, Tanja Waltrip, and Gabriel Perdue for giving me this amazing opportunity to learn and grow as a future physicist. I would also like to thank my mother, Arlene Lopez, and father, Leroy Lopez, who are diligently looking after my son in order for me to live out this wonderful opportunity. Lastly, I would like to thank my little love, Dominic, for giving me the drive I need to succeed.

References

[1] Fermilab. (October 2010). *Experiment Profile: MINERvA.*

[2] The MINERvA Collaboration. (03 March 2014). *Measurement of Muon Neutrino Quasi-Elastic Scattering on a Hydrocarbon Target at Eν ~ 3.5 GeV.*

[3] The MINERvA Collaboration. (03 March 2014). *Measurement of Muon Antineutrino Quasi-Elastic Scattering on a Hydrocarbon Target at Eν ~ 3.5 GeV.*

[4] www.w3.org/People/Raggett/book4/ch02.html