VISUALIZING THE MEAN AND THE STANDARD DEVIATION USING R/RSTUDIO SHINY PACKAGE

Rachel Hufnagel, Ziyi Chen and Mamunur Rashid
DePauw University
Department of Mathematics
Greencastle, IN 46135
mrashid@depauw.edu

Jyotirmoy Sarkar
Indiana University-Purdue University Indianapolis
Department of Mathematical Sciences
Indianapolis, IN 46202
jsarkar@iupui.edu

Abstract: Many of us have experienced an unpleasant situation in which only the mean and the standard deviation of a data set are reported, but we are expected to know everything about the dataset as if those two values were all we needed to know. We would learn so much more if there were easy ways to create and share graphical representations and interpretations of the entire raw data. In this paper, we not only explain what the mean and the standard deviation tell us about a data set, but also describe how to include additional information on the data.

We utilize the work of Sarkar and Rashid (2016) that introduced a geometric visualization of the sample mean based on the empirical cumulative distribution function of the raw data. They also extended the idea to visualize measures of spread such as the mean deviation, the root mean square deviation and the standard deviation. Our research involves creating interactive applications of these methods using R/RStudio Shiny, an open source package that provides an elegant and powerful web framework for building web applications. We hope, upon publication of these tools, users all over the world will use such interactive visualization methods for learning, teaching, and building more advanced tools.

Keywords: Mean, dot plot, empirical cumulative distribution function plot, standard deviation, shiny package

1. Introduction

The mean is a common measure of center, and the standard deviation (SD) of spread, of a set of values of a quantitative variable. These are basic concepts in all quantitative disciplines, and they appear frequently in everyday life applications. While most users understand the mean reasonably well, the SD remains a challenging concept for many—even after they have learned its definition and mastered its computation.

Sarkar and Rashid (2016) provided a new way to visualize the mean in which they used a vertical line method to locate the mean using an empirical cumulative distribution function. On this foundation, they also built some geometric methods to visualize measures of spread: the primary objective is to visualize the SD; additionally, we will visualize the mean deviation (MD) and the root mean squared deviation (RMSD)—quantities that serve as lower bounds for the SD. To explore their different type of visualization methods, see Sarkar and Rashid (2016, 2017a, 2017b, 2019). Though they provided theoretical framework and the visualizations of the methods, they did not provide how these visualization methods can be done interactively. In this paper, we describe the interactive applications of visualizing the sample mean and the sample standard deviation using R/RStudio Shiny package, an open source package that provides an elegant and powerful web framework for building web applications.

Before we proceed, we recall the following summary statistics for our visualizations:

The mean of a set of *n* numbers $\{x_1, x_2, \dots, x_n\}$ is defined by

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

The deviations from the mean are $d_i = |x_i - \bar{x}|$, for $i = 1, 2, \dots, n$; and the MD of $\{x_1, x_2, \dots, x_n\}$ from the mean is the mean of the deviations $\{d_1, d_2, \dots, d_n\}$; and is defined by

$$\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d_i = \frac{1}{n}\sum_{i=1}^{n} |x_i - \bar{x}| \tag{2}$$

The SD of $\{x_1, x_2, \dots, x_n\}$ is defined by

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{3}$$

Oftentimes, a student is inclined to think of the SD as "the average distance of the numbers from their center." Unfortunately, this description, though perfect for the MD, is erroneous for the SD! In an attempt to rectify the error, a teacher may offer a better explanation of the SD as "the root mean squared deviation (RMSD) from the mean." However, when translated into an algebraic expression, the RMSD becomes

$$\tilde{s} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{4}$$

which is different from the SD in (3), because the denominators under the radical signs differ. Expression (4) is also known as the population SD (see, Sarkar and Rashid (2019)) when the set of values $\{x_1, x_2, \dots, x_n\}$ is considered as an entire population.

The correct interpretation of the SD is somewhat convoluted: It is "the positive root *mean* squared deviation (R*M*SD) from the mean," where only the positive root is admissible, and the first mean involves a division by $(n - 1)$. For an explanation of why it is so, see for example, Martin (2003) and Sarkar and Rashid (2016). Here we simply mention that the knowledge of the complete original data $\{x_1, x_2, \dots, x_n\}$ is equivalent to the knowledge of

the transformed data $\{\bar{x}, (x_1 - \bar{x}), (x_2 - \bar{x}), \dots, (x_{n-1} - \bar{x})\}$, since $(x_n - \bar{x})$ can be recovered from the fact that

$$(x_1 - \bar{x}) + (x_2 - \bar{x}) + \cdots + (x_{n-1} - \bar{x}) + (x_n - \bar{x}) = 0 \quad (5)$$

which is an easy consequence of (1).

Note that s^2 is called the variance and $\tilde{s}^2$ is called the mean squared deviation (MSD). The algebraic definitions (2)-(4) and their verbal interpretations may suffice to teach students how to compute the three measures of spread—the MD, the RMSD and the SD. But these algebraic definitions fall short of giving students deep insight or complete understanding.

In section 2, we briefly describe the R/RStudio Shiny package. In section 3, we provide details how we created our first interactive application to visualize the mean using the Shiny package. In section 4, we provide our second interactive application to visualize the standard deviation using the Shiny package. Section 5 concludes the paper with some remarks. All applications have been developed using the R/RStudio computing environment.

2. The Shiny Package in R

A package in R/RStudio is merely a collection of functions, data, and working code that is in a well-defined format and also allows a user to start with a desired set of inputs. Within the integrated development environment (IDE), R/RStudio, the user is granted access to a library of packages which ultimately act as ground work for building and development. To develop our applications, from a user-interaction perspectives, in our assessment the best package available in the R/RStudio library would be the shiny package (Chang, et al., 2017). The shiny package creates a *reactive* environment for any code that might be added to already contained in the package. So, what does reactive mean? It means that when the code is executed a direct line of communication is opened between the application and the user. Basically, it is an environment once a user acts on it, the software will react within seconds. Thus, the user interaction information can be processed immediately, and results can be shown. The Shiny package is also unique in the sense that it is very flexible and easy to work with. For example, unlike other IDE's where users could run into problems with classes, data structures, and organization, the Shiny package allows you to input your R code directly into the outline.

For both our applications, input sliders were used. Input sliders require a minimum value and a maximum value, and can be thought of as a portion of the number line with a range set by the users. The input slider is linked to a variable that is a key to determining the output—either the mean or the standard deviation. The output can be a range of different widgets. In both our applications, graphs were the output produced as visual aids. The shiny package also allows many other ways to input or 'speak' to the application, however in our research only slider inputs were used. When the user selects a point on the slider input, any weight carried by that variable is depicted in the output section of the application instantaneously.

3. Mean using the R/RStudio Shiny Package

The first application we built allows users to interact with the software by locating the mean and verifying whether the areas around an empirical cumulative distribution function (ECDF) are equal. Also known as a step plot, an ECDF plot with the mean indicated by a vertical line is shown in the figure below. See, Sarkar and Rashid (2016) for details.

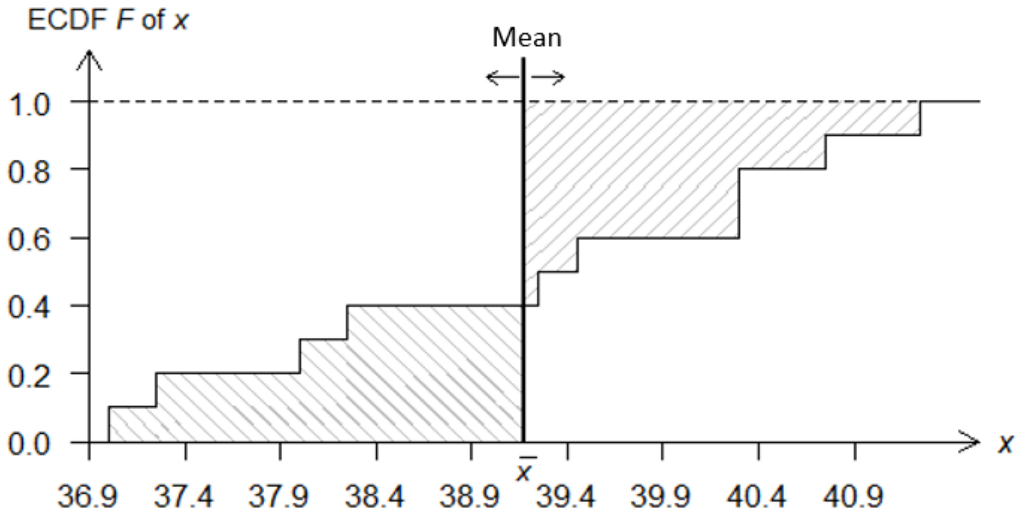

Figure 1. A vertical line placed at the mean equalizes the areas of the shaded regions

No matter what the size of the data set, in this case, we always rely on the area of the ECDF plot to locate the mean. As long as the areas to the left and to the right of the vertical line indicated in Figure 1 are equal, we have found the mean and there is no need of any further computation. For our application, we use a slider input that can be moved appropriately to find the mean. Figure 2 shows unequal areas; that is, the vertical line is not placed appropriately to find the mean. We use two different types of slant lines to the left and to the right of the vertical line indicating that the two areas are not the same. When the slider moves from left to right, the application calculates the areas to the left and to the right of the vertical line and also shows the exact areas at the upper-left corner of the graph.

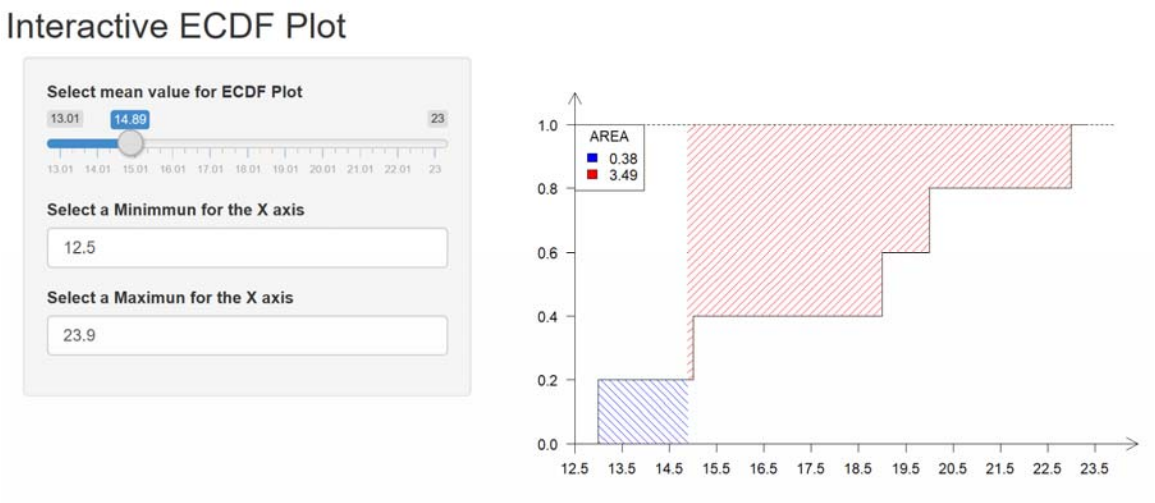

Figure 2. Interactive ECDF plot application with the mean not identified correctly

When the input slider moves from the left to the right, until the vertical line locates the mean appropriately is shown in Figure 3. In this case, we use two different types of filled colors to the left and to the right of the vertical line indicating that the two areas are now the same. The intermediate value theorem (Anton, 1984) says that there is one and only one position for the vertical line to achieve equality of the two areas.

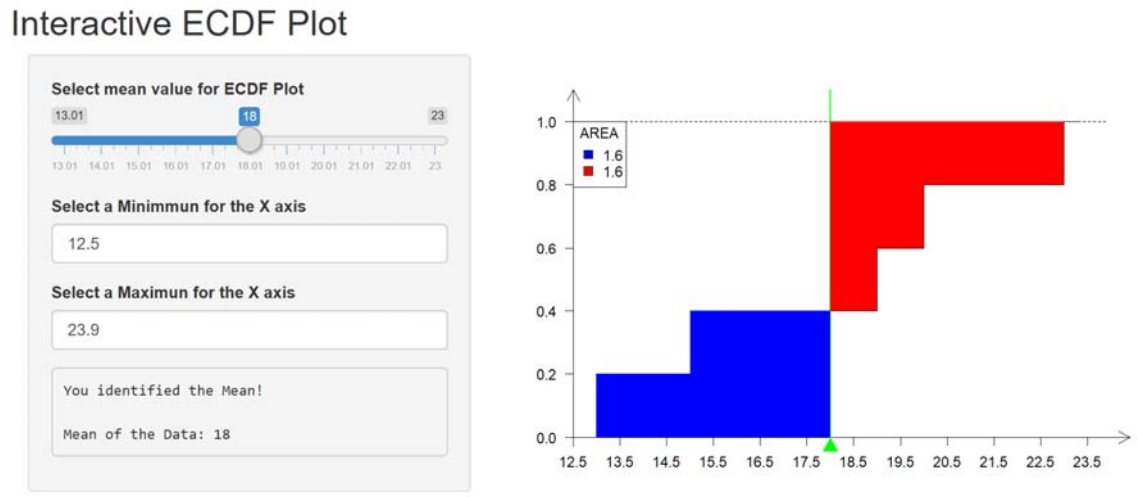

Figure 3. Interactive ECDF plot application with the mean identified correctly

If the user pushes the vertical line too far to the right, the solid colors will once again turn into slant lines as shown in Figure 4 below.

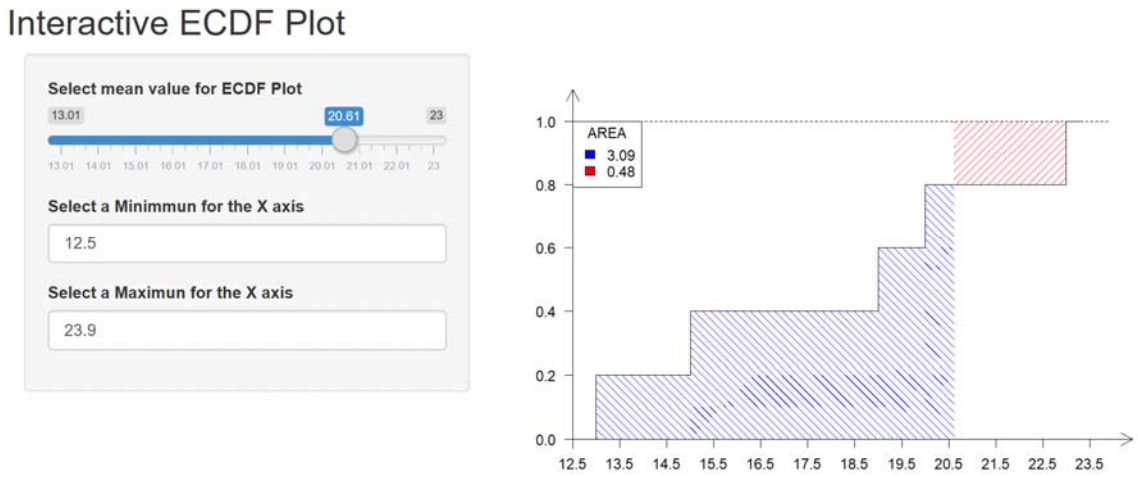

Figure 4. Interactive ECDF plot application with the mean identified incorrectly

Notice that the actual ECDF plot is never changing, only the shading/color and the vertical line denoting the mean are dynamic. As we can see in the above figures, the slider is first communicating with the user to receive an input and then producing the plot as an output, which in our case is the ECDF plot. There are also some numeric inputs below the slider which allow the user to change the minimum value of the vertical line as well as the maximum value. These two additional inputs are in place so that the user can change the viewing window of the plot output (for instance, when comparing two data sets). The default values are set to be the minimum value and the maximum value of the dataset. There is also a text output located under the three widgets which allows the user to see which value the slider input has been dragged to as well as whether the mean has been identified (correctly) or approximated (incorrectly).

4. Standard Deviation using the R Shiny Package

Our next application is designed with similar intentions as the interactive mean application. With any given data set, our hope is that with a few inputs we will be able to locate and see where the standard deviation is. However, the implementation is a bit different. Sarkar and Rashid (2019) developed a static graph shown in Figure 5.

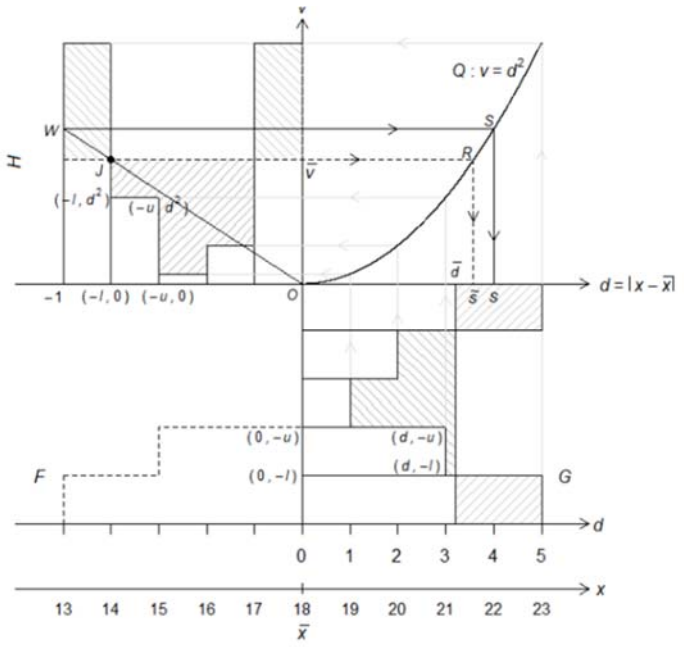

Figure 5. Visualizations of MD, SD and RMSD

As we can see there are many components to this graph but the most important thing to note is that this new static graph has been built with the foundation of an ECDF plot. Before we explain our technological implementation, we first quote from Sarkar and Rashid (2019) the nine steps needed to construct Figure 4:

1) On the ECDF *F* of the raw data, draw the mean vertical line $x = \bar{x}$. Reflect the portion of the ECDF to the left of the vertical line $x = \bar{x}$, about that line, with the reflection falling on the right side.
2) Let *G* be the resultant graph to the right of the vertical line $x = \bar{x}$ consisting of both the reflected part and the part of *F* that was already to the right of the vertical axis. See Remark 1 below for an interpretation of graph *G*.
3) At the top edge of graph *G*, introduce a new horizontal axis $d = |x - \bar{x}|$ representing the (absolute) deviation from the mean. The vertical axis v is the line $d = 0$. Thus, the graph *G* now resides in the fourth quadrant between $v = -1$ and $v = 0$. Draw the graph *Q* of $v = d^2$ in the first quadrant.
4) Find the mean vertical line $d = \bar{d}$ of *G* so that the areas of shaded regions to its two sides are equal. Then $\bar{d}$ is the mean deviation (MD) of the data. In fact, $\bar{d}$ is also the total area of the two shaded regions $A_{\bar{x}}$ and $B_{\bar{x}}$ in Fig. 1.
5) Extend the steps of *G* all the way left to reach the vertical axis $d = 0$, thereby obtaining a collection of rectangles G that are left aligned at $d = 0$, contiguously stacked between $v = -1$ and $v = 0$, and of varying widths.
6) Consider any one member rectangle in *G*. Let the coordinates of its four vertices be denote by $(0, -u), (0, -l), (d, -l), (d, -u)$. To these vertices apply the transformation $(d, v) \rightarrow (v, d^2)$, so that they move to four new vertices having coordinates

$(-u, 0), (-l, 0), (-l, d^2), (-u, d^2)$ respectively: First, draw a vertical line through $(d, -u)$ until it meets Q at (d, d^2); then turning left, draw a horizontal line $v = d^2$. (See the light, solid, directed lines.) These new vertices form a transformed rectangle. When all members of G have been so transformed, call the collection of all such transformed rectangles H. The rectangles in H are bottom aligned at $v = 0$; they are contiguously arranged between $d = -1$ and $d = 0$; and they have varying heights resembling a histogram. Call their graph *H*.

7) Find the mean horizontal line $v = \bar{v}$ of *H* so that the shaded areas of two regions above and below it are equal. Indeed, $\bar{v} = \tilde{s}^2$ is the MSD, and its square root is the RMSD.
8) Join O and *J*= $(-1 + 1/n, \bar{v})$ by a line and extend it to meet the vertical line $d = -1$ at *W*= $(-1, w = \bar{v}.n/(n-1))$. Indeed, $w = \bar{v}.n/(n-1) = s^2$ is the variance, and its positive square root is the SD.
9) The horizontal lines through J and W intersect the quadratic curve *Q* at $R = (\tilde{s}, \tilde{s}^2)$ and $S = (s, s^2)$ respectively. Dropping vertical lines from R and S, we see the RMSD $\tilde{s}$ and the SD s on the horizontal d-axis. (See the dark, dotted/solid, directed lines.)

After incorporating these steps into the shiny package, we can build our second application to visualize the MD, the RMSD and the SD. In this case, we use three input sliders to be used in order: the very bottom one for MD and the left one for MSD for which we will get the RMSD. Figure 6 shows that the vertical lines that are not located in correct positions to find these quantities correctly. Figure 7 depicts the correct values of all three statistics after adjusting the sliders. Only when both sliders are correctly placed, the right side of Figure 7 reports exact calculated values of the MD, MSD, RMSD, and SD.

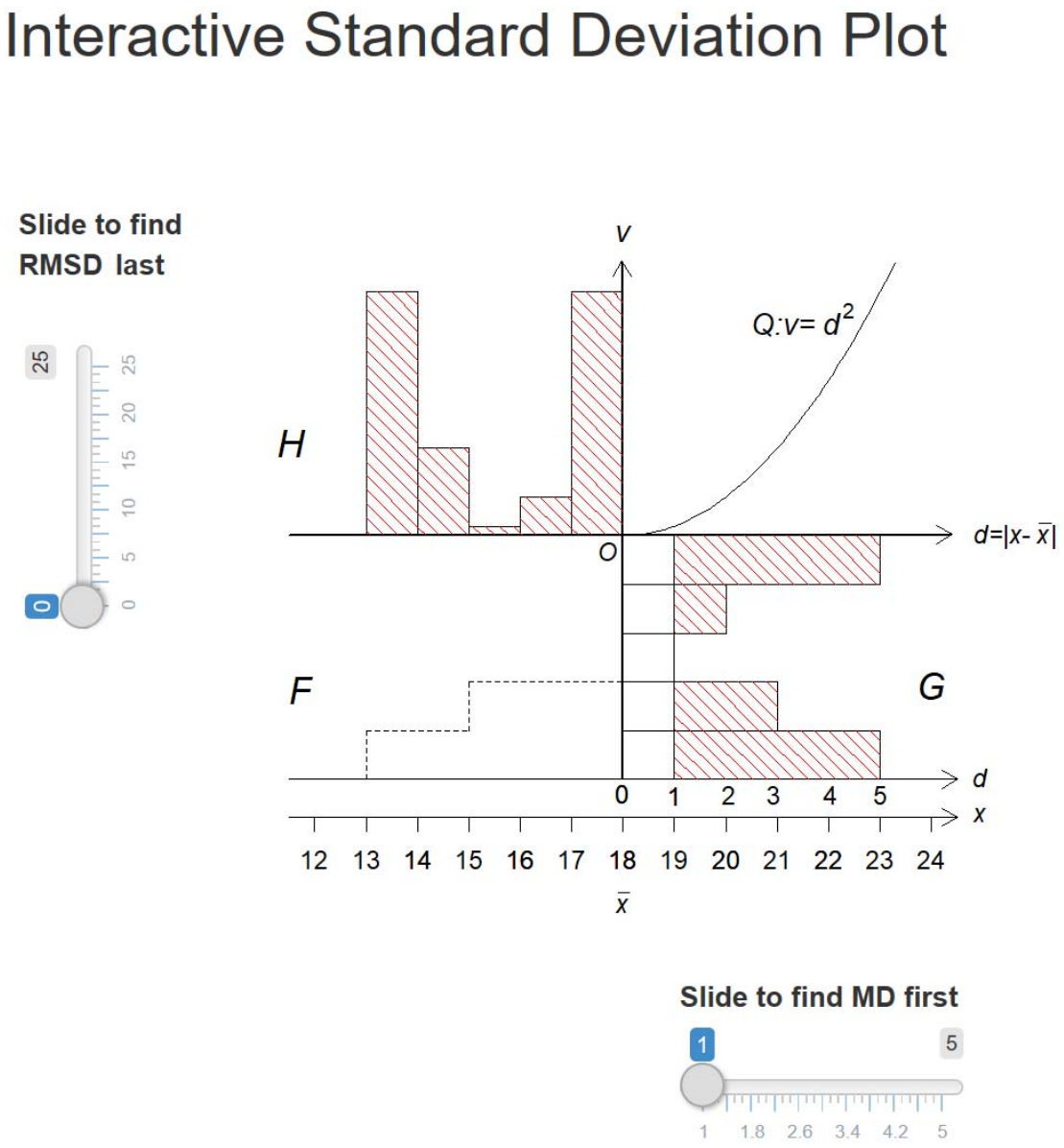

Figure 6. Interactive application plot with the MD, RMSD and SD not identified correctly

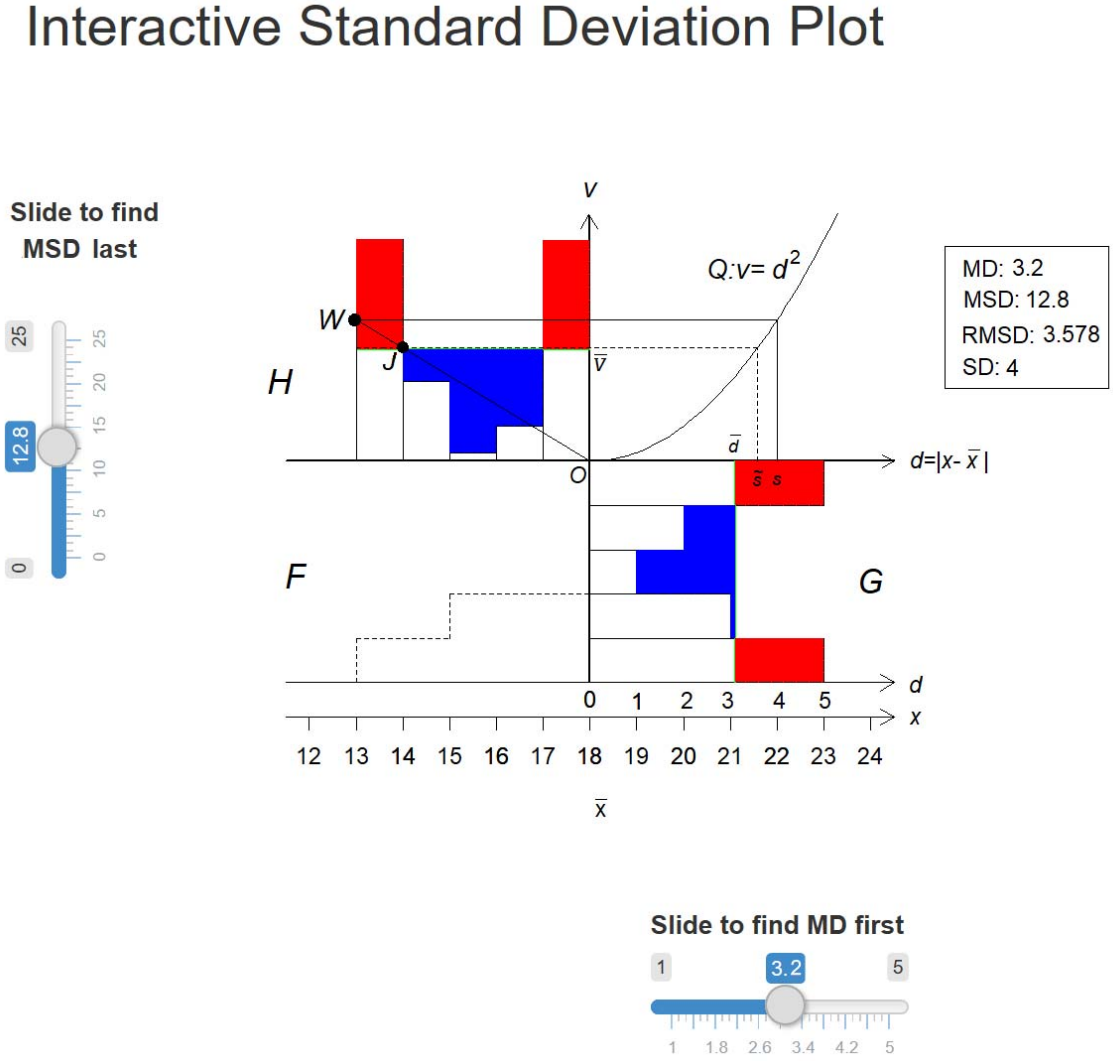

Figure 7. Interactive application plot with the MD, the RMSD and the SD identified correctly

5. Conclusion

Our research involves creating interactive applications of mean and standard deviation using R/RStudio Shiny package. These two concepts have been taught in the classrooms almost uniformly. But what other ways are there to teach such elementary concepts of statistics in the classroom? One answer could be to jump on the train of technology, to incorporate higher level software with lower level learning. This is where the software system, R/RStudio comes into play. R/RStudio offers a wide range of packages that can be helpful in the construction of further applications. We hope, upon publication of these

tools, users all over the world will use such interactive visualization methods for learning, teaching, and building more advanced tools.

Acknowledgments

We thank the VPAA's office and the Science Research Fellows Programs of DePauw University to support 2019 summer student-faculty research funds for conducting the research.

References

[1] Anton, H. (1984). *Calculus with Analytic Geometry*, 2nd ed. New York: Wiley, p. 189.
[2] Chang,W., Cheng, J., Allaire, J., Xie, Y., and McPherson, J. (2017). *Shiny: web application framework for R*, R package version 1.4.0.2.
[3] Sarkar, J. and Rashid, M. (2016). Visualizing Mean, Median, Mean Deviation and Standard Deviation of a Set of Numbers. *The American Statistician*, 70(3), 304-312.
[4] Sarkar, J. and Rashid, M. (2017a). Visualizing the Sample Standard Deviation. *Educational Research Quarterly*, 40(4), 45-60.
[5] Sarkar, J. and Rashid, M. (2017b). Visualizing the Mean and the Spread of a Random Variable. *Journal of Propagations in Probability and Statistics*, 17(2), 59-70.
[6] Sarkar, J. and Rashid, M. (2019). Have you seen the standard deviation?. *Nepalese Journal of Statistics*, 3, 1-10.
[7] Martin, M. (2003). 'It's like...you know': The Use of Analogies and Heuristic in Teaching Introductory Statistical Methods. *Journal of Statistics Education*, 11. Available at https://www.tandfonline.com/doi/full/10.1080/10691898.2003.11910705
[8] R Core Team (2013). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. Available at http://www.R-project.org/.