Adapting Statistical Software to the Apple iPad

Neil W. Polhemus

Statpoint Technologies, Inc., 560 Broadview Ave., Warrenton, VA 20186

Abstract

This paper considers how statistical software can take advantage of what the Apple iPad and other tablet computers offer today's data analysts. Adapting statistical software so that it works well on such devices requires changes that account for both the hardware constraints and the need for easy access to users' data. Challenges to effective data visualization include issues such as screen size, lack of a cursor, and non-support for either Flash or Java. On the other hand, the portability of the platform offers users instant access to their data and analytical routines. As an example, the paper describes the adaptation of Statgraphics *Online* for use on the iPad.

Key Words: Software, iPad, tablet, visualization, Statgraphics, Statlets, web application

1. Background

Much software has been developed during the last several decades with the expectation that it would be used on a computer with a relatively large screen, a keyboard, and a pointing device such as a mouse. Adapting such programs to run on a tablet computer such as an Apple iPad requires changes to accommodate the smaller platform and user interaction through a touch-sensitive screen.

The types of people who would use statistical software on a tablet computer fall into two main groups:

1. Users wishing to enter data on the tablet itself and immediately analyze it.
2. Users wishing to access larger databases that have already been created and reside somewhere else.

It is important to create software that satisfies both types of users.

2. Decision #1: What type of App?

The first decision that needs to be made concerns the type of application to be constructed. Two types would seem to be most promising:

1. A native application designed specifically for a single type of tablet. Such an application could be tailored to the look and feel of that device and could be optimized to achieve maximum performance. However, even though a tablet such as the Apple iPad has a large market share, there are formidable competitors that would be excluded by writing a program specifically for that device.
2. A cloud-based application that could be accessed by any device that has a web browser. Such an approach would give users a consistent interface across

multiple platforms. For individuals wishing to use the iPad to access their data while away from the office, but who like to use their desktop or laptop computer at other times, this approach has definite advantages.
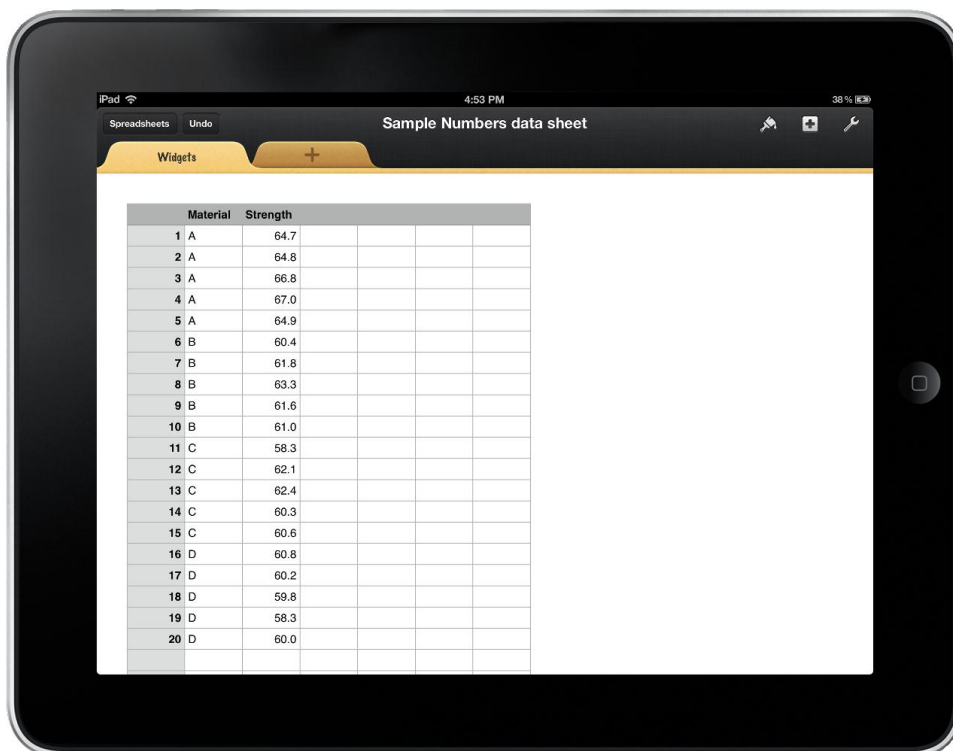
In adapting Statgraphics *Online* to run on the Apple iPad, we decided to take the second approach. This insured that users could pick up any device and after entering a username and password, have access to their data and the same set of analysis routines.

3. Decision #2: How should data be entered?

The second decision is a critical one for any statistical software package: How should the data be entered? To accommodate the two types of users envisioned for the iPad, it was decided that:

1. There should be some capability to enter data directly on the iPad through a familiar spreadsheet interface.
2. There should be the ability to import data from files created in other applications external to the iPad.

Since Apple Computers offers a good, inexpensive spreadsheet app for the iPad called *Numbers*, it was decided to let users who wanted to enter data directly on the iPad do so in that program. The figure below shows the entry of some typical data:
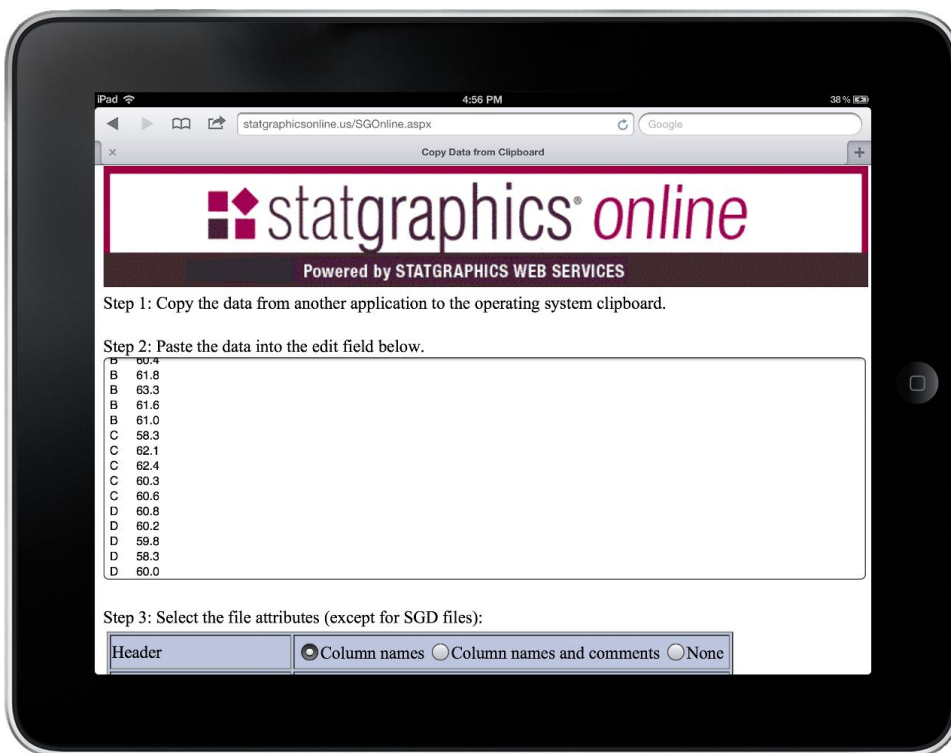

Figure 1: Data entry on the iPad using Apple's *Numbers* application

Note that each column contains a header by which that column may be referenced. The data may then be copied from *Numbers* to the iOS clipboard and pasted into an edit field in the Statgraphics app:

Figure 2: Pasting data from the clipboard into the *Statgraphics* application

The data are then saved in a user file on the web server:
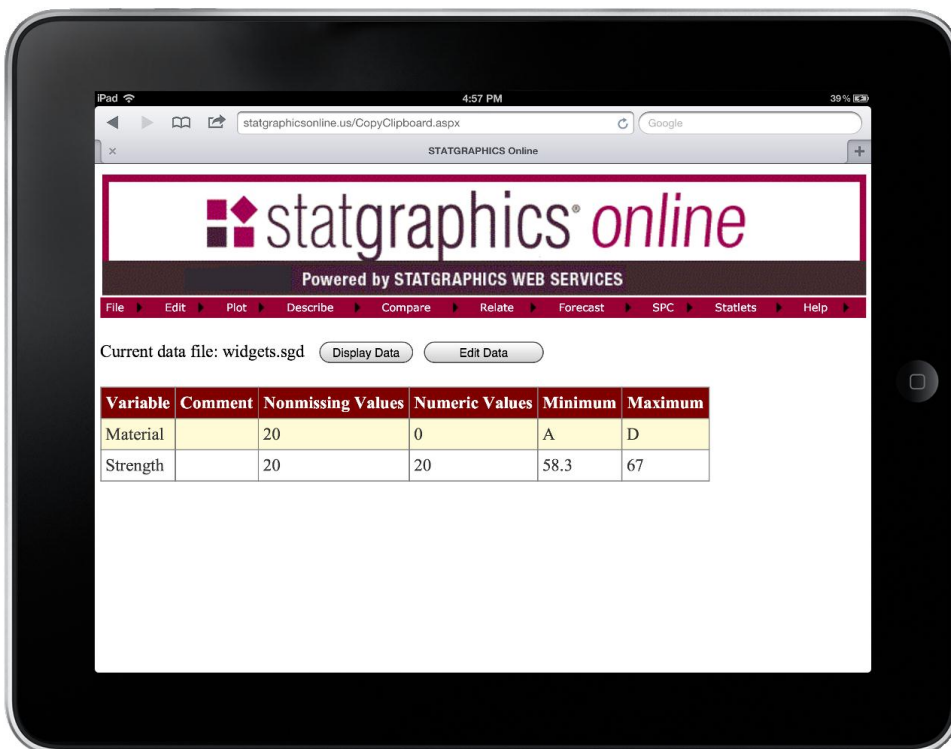
Figure 3: Summary of data file after import into Statgraphics *Online*

For users wishing to read data files created on other computers, the approach is somewhat different:

1. The data file is first uploaded to the web server. Supported files include delimited text files, Excel files, and XML files.
2. The data are then imported into a standard Statgraphics file and saved under the user's name and password.

4. Decision #3: What architecture should be used to create the web application?

To use the statistical application on an iPad, the user first loads the Safari web browser and then navigates to the site where the application is stored. This could be a site on the Internet or on an organization's internal server, possibly behind a firewall. A menu of statistical procedures is offered, each of which begins by displaying a data input page similar to that shown below:
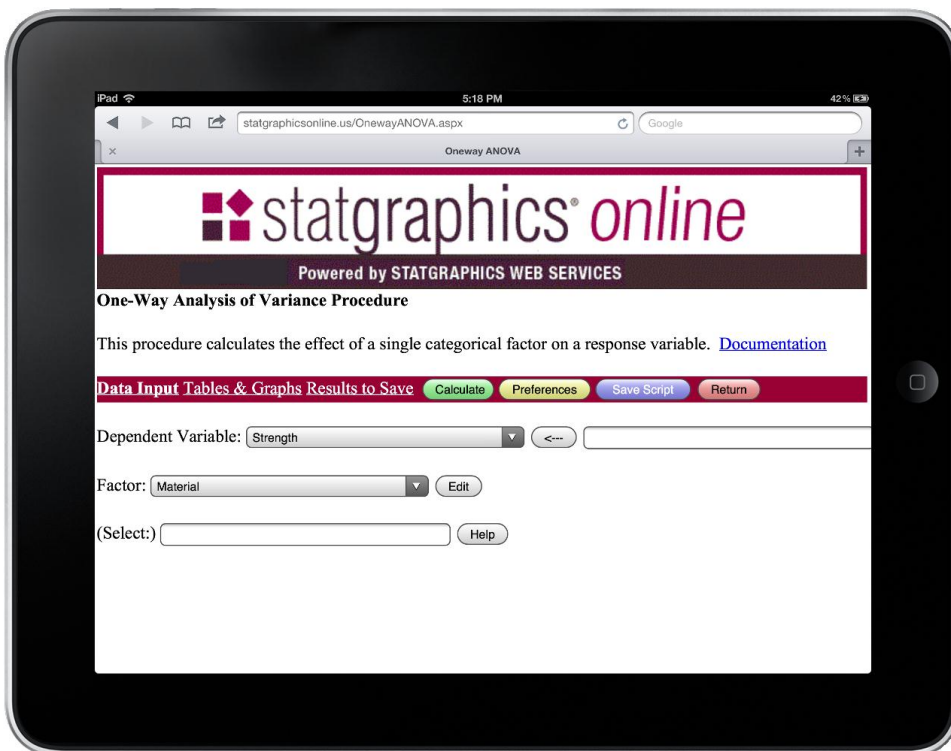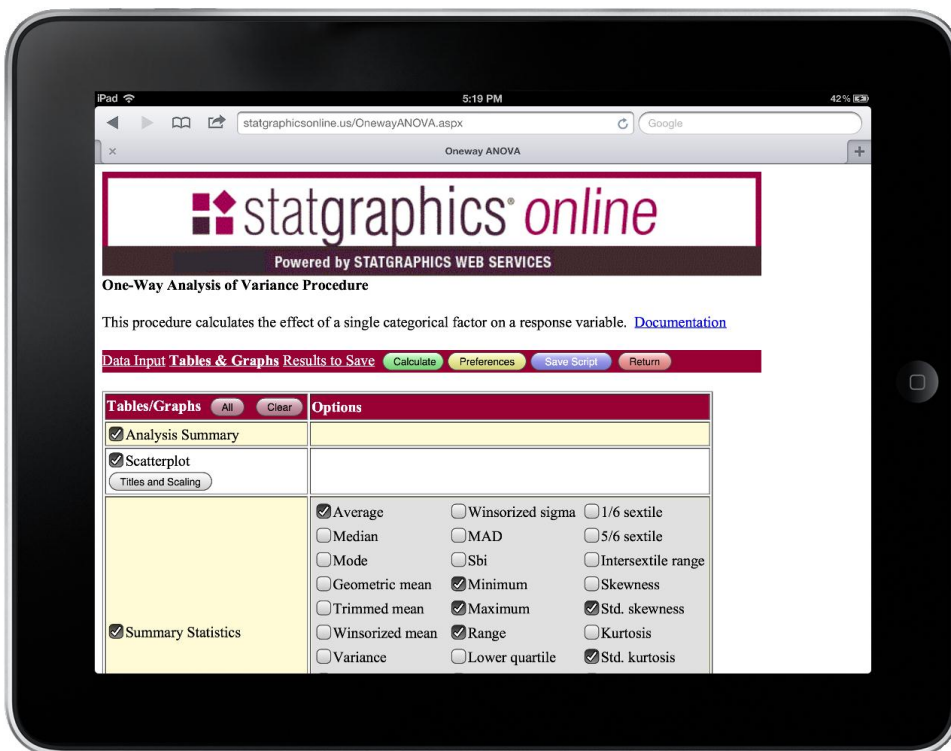
Figure 4: Data input page for the One-Way Analysis of Variance procedure

Users select data columns by name to specify the required input. For example, the above figure requests an analysis of *Strength* by type of *Material*. An edit field is also provided to the right of each variable list so that the user may enter transformations by typing an expression such as *LOG(Strength)*. The *Select* field is used to select a subset of the data rows. For example, material D could be excluded from the analysis by typing *Material<>"D"*.

Additional pages are presented if the user clicks on one of the available links:

1. *Tables & Graphs* – This shows a list of tables and graphs that may be created, together with relevant options.
2. *Results to Save* – This shows a list of numeric output that may be saved in an output data file.

Figure 5: Tables & Graphs options page for the One-Way Analysis of Variance procedure

When the user presses the *Calculate* button, an XML script is created with the statements needed to perform the analysis. The script is then processed by an ASP *.Net* program running on a web server, which executes the statistical code which is written in C#. An ASP page is created in which text output is written in standard HTML and graphics output is stored in GIF files. The page is returned to Safari which displays it in a format similar to that shown below:
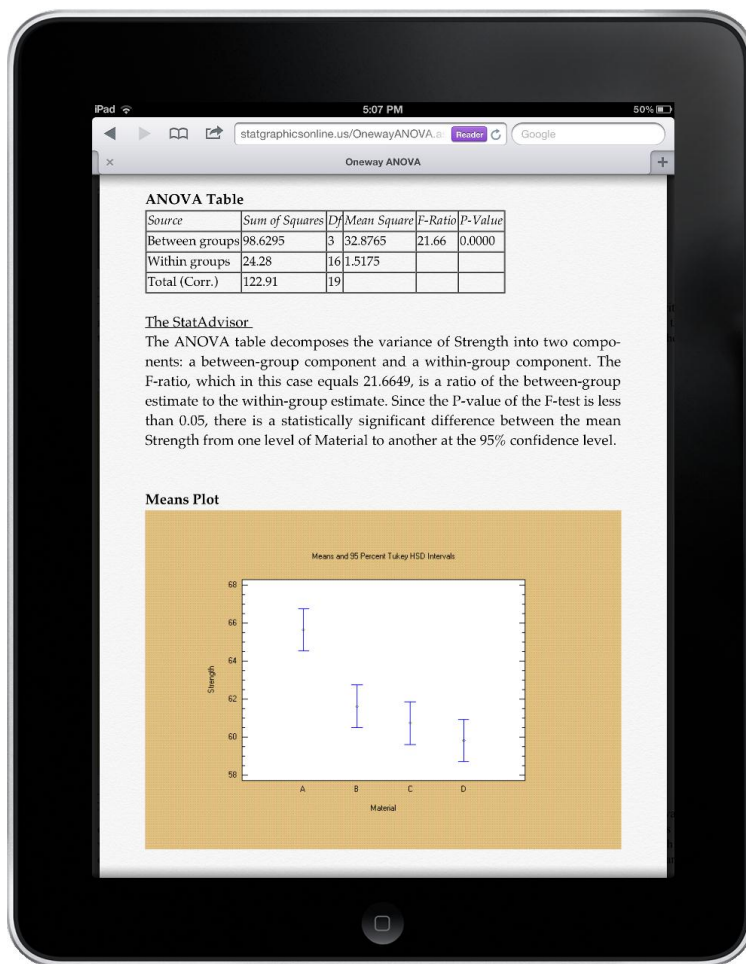
Figure 6: Statistical output as displayed by Safari

5. Statlets: Interactive Graphs with User Interaction

The steps described above, in which an XML script is returned to the server which then generates output and returns it to the browser, works well when the user does not need to interact with the output in real time. They can of course change the options and resubmit the script, but there is no facility for dynamic interaction with the output.

A special type of procedure has been created which allows for such interaction. Called *Statlets*, these procedures display a single graph which the user can then manipulate. Under browsers such as Firefox and Internet Explorer on a desktop or laptop computer, *Statlets* are implemented as Java applets. For example, the figure below shows an interactive map of the United States:
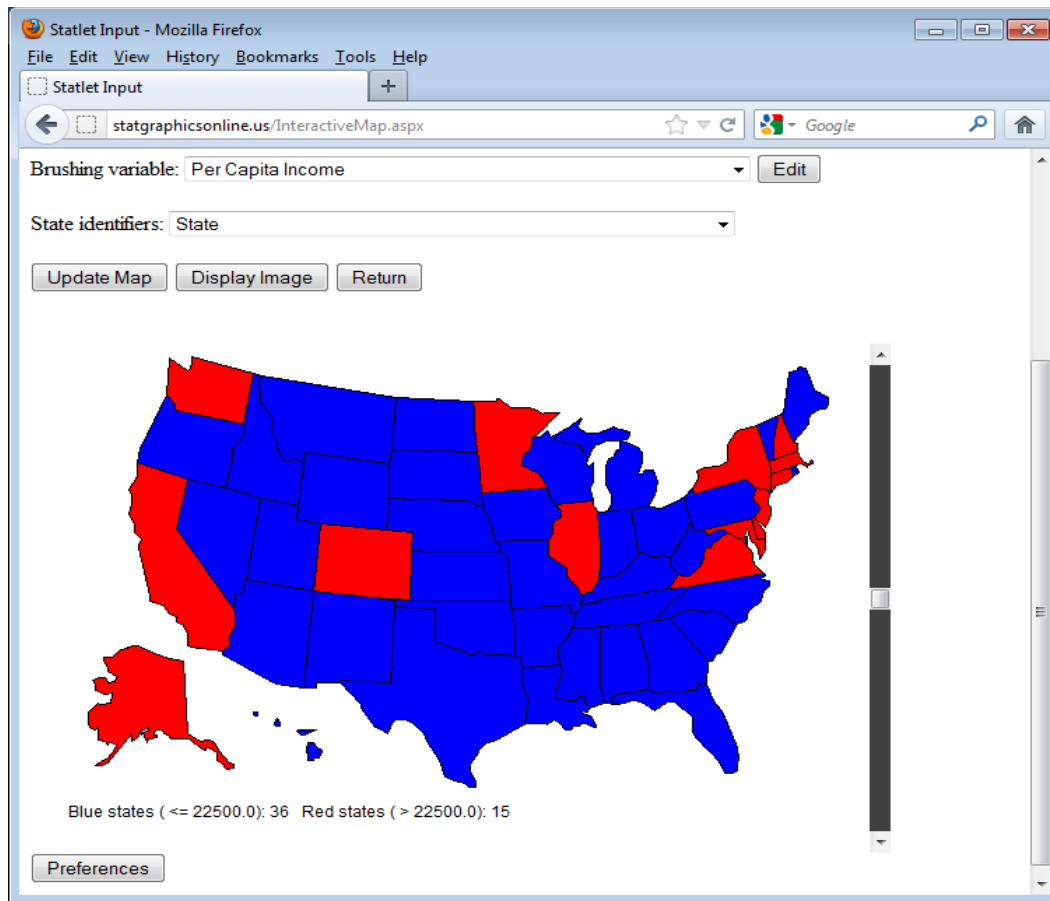
Figure 7: Interactive Statlet displaying per capita income for each state in the United States

The Statlet colors each state according to the value of the per capita income in that state. States with a per capita income of no more than $22,500 are colored blue. States with higher per capita income are colored red. The user may click on the scrollbar with a mouse and move the mouse to change the cut-off between red and blue states. The states change color as the mouse is moved, which gives the user a method for visualizing the distribution of per capita income throughout the country.

Since the Apple iPad does not support Java, a different implementation is needed for such devices. In addition, iPad users do not have a mouse, but instead are used to touching the screen to make things happen. To accommodate the iPad, each *Statlet* was rewritten using HTML5. A canvas is created and the map is drawn on the canvas:
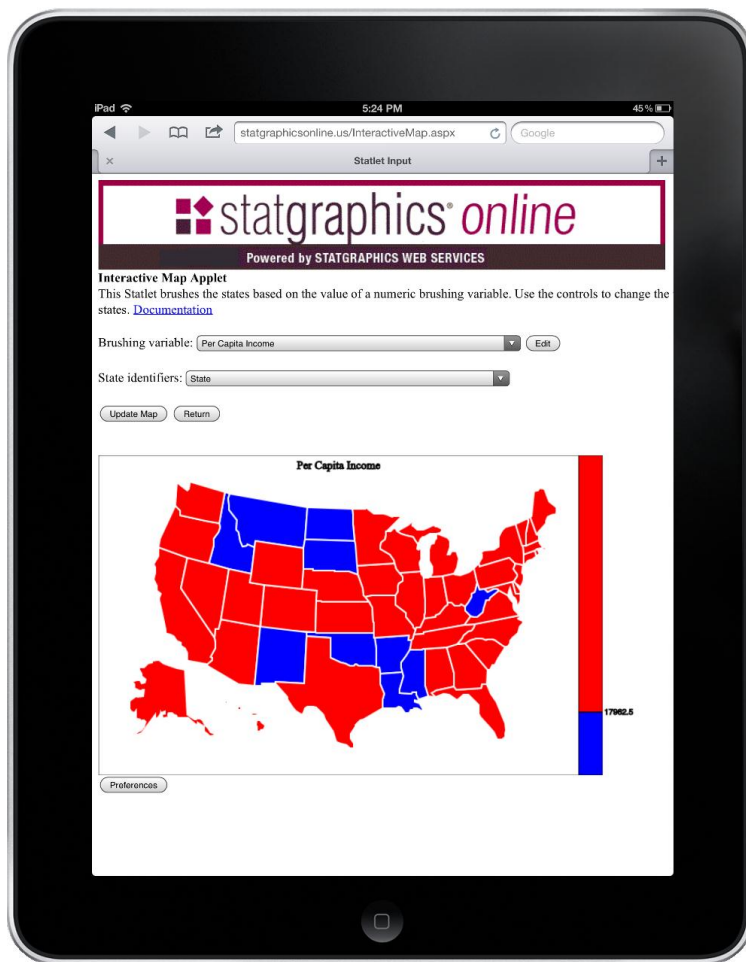
Figure 8: Interactive Statlet rewritten using an HTML5 canvas

To change the cut-off between red and blue states, the user slides a finger along the colored bar at the right margin of the graph.

In rewriting the Statlet to work on the iPad, some critical decisions had to be made about the division of labor between the client and the server.

- *Statistical computations* – It was decided that all statistical computations should be performed on the server. One reason for this decision was computational efficiency, since compiled C# is much faster than interpreted JavaScript. Second, the code could then be validated in a single environment (the server) rather than under various browsers on many output devices. Third, security of the source code could only be guaranteed if the code remained on the server.

- *Graphics metafile* – The initial graphics metafile, containing objects such as the polygons that define each state in the above Statlet, is created on the server. It is then passed to the browser, which draws the initial image on the canvas. Whenever the user moves a finger to change the cut-off between red and blue, the client program receives a message, changes the graphics objects as necessary, and then redraws the image.

- *Graphics rendering* – All graphics rendering and response to user actions is done on the client using JavaScript, since the desired responsiveness makes it unacceptable to request additional action by the server.

6. Conclusion

The adaptation of existing statistical software to run on tablets such as the Apple iPad necessitates changes in both the user interface and in the way output is created. To support a wide range of devices, it was decided to structure the program so that it could run on any device that hosted a web browser. Standard statistical procedures are invoked by sending an XML script to a web service, which returns the output as HTML with imbedded images. For interactive procedures, HTML5 canvases and JavaScript replace Java applets.

References

A demonstration version of Statgraphics *Online* may be accessed at *www.statgraphicsonline.us*. You may contact the author by sending an email to *neil@statgraphics.com*.