Paper SAS216-2014

Leveraging SAS® Visualization Technologies to Increase the Global Competency of the U.S. Workforce

Jim Bauer, SAS Institute Inc., Cary, NC

ABSTRACT

U.S. educators face a critical new imperative: to prepare all students for work and civic roles in a globalized environment in which success increasingly requires the ability to compete, connect, and cooperate on an international scale. The Asia Society and the Longview Foundation are collaborating on a project to show both the need for and supply of globally competent graduates. This presentation shows you how SAS assisted these organizations with a solution that leverages SAS® visualization technologies to produce a heat map application. The application surfaces data from more than 300 indicators and surfaces over a quarter million data points in a highly iterative heat map. The application features a drillable map that shows data at the state level as well as at the county level for all 50 states. This endeavor involves new SAS® 9.4 technology to both combine the data and to create the interface. You see how SAS procedures, such as PROC JSON, which came out in SAS 9.4, were used to prepare the data for the web application. The user interface demonstrates how SAS/GRAPH® output can be combined with popular JavaScript frameworks like Dojo and Twitter Bootstrap to create an HTML5 application that works on desktop, mobile, and tablet devices.

INTRODUCTION

The U.S. heat map project detailed in this paper was a joint effort between SAS, the Longview Foundation, and the Asia Society. The Longview Foundation and the Asia Society were looking for a way to use data that they had acquired from various sources. The data was stored in 34 separate Microsoft Excel spreadsheets, which, when combined, contained 312 indicators. Given the number of indicators, we decided to group them into four main categories: economics, education, demographics, and key indicators. The key indicators were hand-picked indicators that were seen as being broad indicators of higher importance. The key indicators were further classified into subcategories. This hierarchical classification system was chosen to provide the user with an easier way to navigate the variables available for analysis.

Another dimension in this project was that the data that was acquired included data from both the national and state levels. There were indicator values for each state and the District of Columbia at the national level. At the state level, the data was provided on a county-by-county basis. State-level data could not be obtained for some indicators.

Because the scope of this project was limited to the visualizing mapping endeavor, we decided to supply the Longview Foundation and the Asia Society with the output from SAS/GRAPH procedures. This output was wrapped in a custom viewer to produce a basic, but powerful, application. This application could be deployed on their servers.

APPLICATION OVERVIEW

The application featured a hierarchical selection box on the left and a map on the right (as shown in Figure 1):

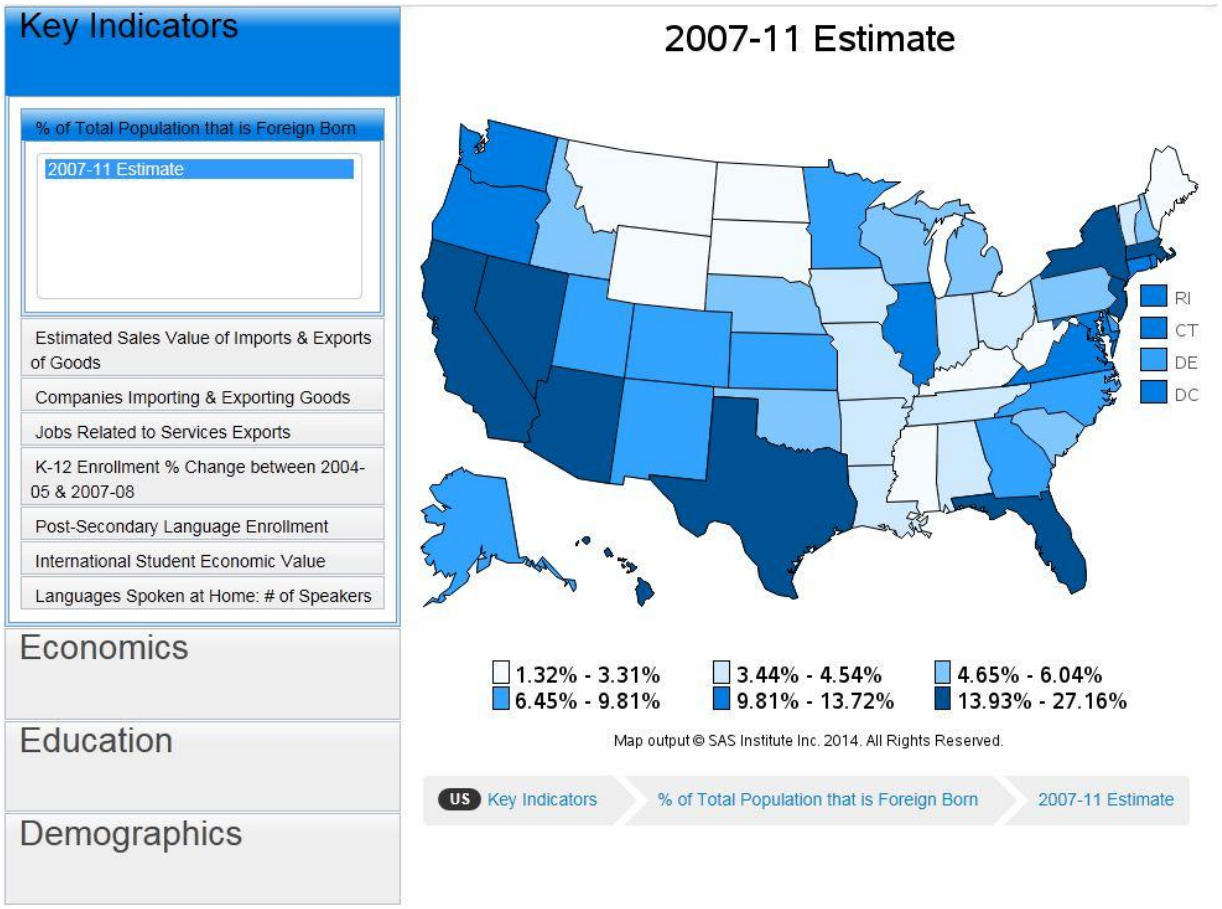

Figure 1. Heat Map Application

In the application, the user selects a category. Then, the user sees subcategories. After the user selects a subcategory, the user sees the available indicators. After the user selects an indicator, the map on the right updates to display a color that indicates the intensity of the value for a region (state, in this case) for the indicator that is selected. The breadcrumb below the map shows the indicator, category, and subcategory that are displayed. The map, legend, and boxes representing the smaller states were created using SAS/GRAPH. The map is clickable. For example, clicking on North Carolina displays a new map (as shown in Figure 2):

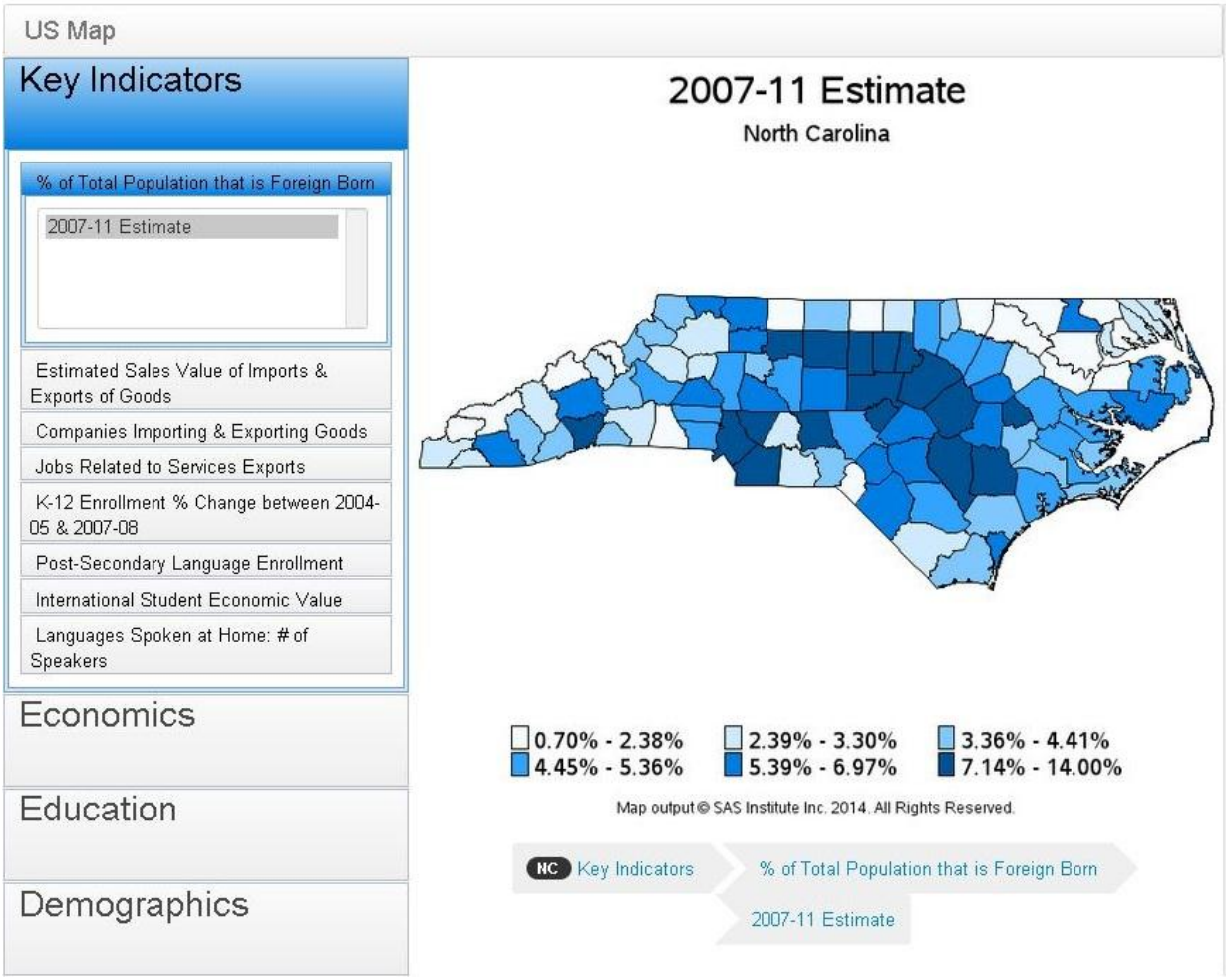

Figure 2 2. Heat Map Application—State View

READING THE DATA

SAS provides a number of interactive and batch facilities to import data from external sources such as Excel spreadsheets. The data provided for this project was in 34 separate spreadsheets. Each spreadsheet had 51 separate tabs (a tab for national data and a tab for each state). Because of the sheer volume of spreadsheets and the layout of the data, we decided to use a SAS programming approach to read the data into SAS data sets. PROC IMPORT was used and it solved most of the problem.

Here is the PROC IMPORT code:

```
PROC IMPORT OUT=&raw_ds
     DATAFILE="&xls"
     DBMS=EXCELCS REPLACE;
     SHEET=%str("'&xls_sheet'");
     SCANTEXT=YES;
     USEDATE=YES;
     SCANTIME=YES;
RUN;
```

PROC IMPORT greatly simplifies the effort to get data from Excel spreadsheets into a SAS data set. The procedure just needs a path to the Excel file, the tab on the spreadsheet to read, and the location where you want the SAS data set stored. To read multiple tabs per spreadsheet and multiple spreadsheets, the SAS Macro Language was used. Macros enabled the procedure to be called multiple times to read the data. Once the data was read, multiple steps

cleansed the data. The TRANSPOSE procedure swapped rows for columns. As a result, each state and indicator combination was not an individual observation in the table (in other words, each indicator did not have a column). This allowed a WHERE clause to be applied to the table to produce a table in a format compatible with the GMAP procedure.

USING PROC GMAP FOR MAP CREATION

There are lots of different ways to display the data from the Longview Foundation and the Asia Society. For example, the data could be displayed as a bar chart with each state on the x axis and the magnitude on the y axis. We decided to use maps because that visualization approach seemed to offer the user a more immediate understanding of how a particular indicator was impacted based on its geographical location. Once the mapping approach was chosen, the stakeholders added requirements to the maps. The requirements included the ability to hover over the states to display the values and the ability to click on a state to display a county-level map for that state. The maps needed legends so that the user could discern the intensity of an indicator based on the map coloring. PROC GMAP met all of these requirements.

The following is an example of the PROC GMAP code used to generate one of the maps:

```
proc gproject project=robinson nodateline data=map3 out=map_us;
   id county;
proc gmap map=map_us data=mapdata all;
   id county;
   choro response /
   coutline=black
      levels=6
       legend=legend1
   html=htmlvar
   legend=legend1
   html=htmlvar
   des=""
   name="&state.&&j";
run;
```

The code produces an HTML file and an image file. An additional HTML file contains an image map of the clickable areas. The mapdata data set is the input data set that contains all of the values for a given indicator. The values are contained in the response variable.

Because of how this application was going to be deployed, all of the maps needed to be generated in advance. The previous code generates just one individual map. However, for this application, we needed the code to generate 12,150 individual maps in advance. The SAS Macro Language was used to loop though PROC GMAP for all 50 states and the District of Columbia as well as through all 312 indicators. For the state-level maps, every county was looped through as well as each indicator.

The following is an example of the PROC GMAP code used to generate one of the maps:

```
%macro genmaps;
    %do s=1 %to 51;
          %do i=1 %to 312;
                data mapdata;
                   set alldata(where=(state_id=&s and indicator_id=&i));
                run;
                proc gproject project=robinson nodateline data=map3 out=map3;
                   id county;
                proc gmap map=map3 data=mapdata all;
                   id county;
                   choro response /
                   coutline=black
                      levels=6
                          legend=legend1
                   html=htmlvar
                   legend=legend1
                   html=htmlvar
                   des=""
                   name="&state.&&j";
```

```
                run;
            %end;
    %end;
%mend genmaps;
```

By combining the SAS Macro Language with PROC GMAP, we were able to generate a large quantity of maps. And, each map had customizations that supported the needs of the application.

CREATING JSON METADATA

Most of the interactive functionality of this application is the result of using JavaScript embedded in HTML pages. The hierarchical selection box uses JSON data to display all of the indicator choices. The JSON data structure is an efficient way of storing a large amount of data while minimizing the size of the file. The application started development before the release of SAS 9.4 when PROC JSON became available.

To create the JSON data file, the SAS DATA step was used to write the JSON structure:

```
data one;
```

Here is the source code:

```
DATA _null_;
    file "\\kmcbi2\US_heatmap\src\data.json";
    set indicators end=lastrec;
    array state_array {52} $ 2 state1-state52;
    by category_id subcat_id id;
    if _N_ eq 1 then do;
        put '[';
    end;
    if first.category_id then do;
           put '{"category":"' category '",';
           put '"subcategories": [';
    end;
    if first.subcat_id=1 then do;
           put '{"subcategory":"' subcategory '",';
           put '"indicators": [';
    end;
    put '{"id":"' id '",';
    text=tranwrd(display_label,'"','\"');
    put '"display_label":"' text '",';
    put '"no_data_states": [';
    do i=1 to 52;
        if state_array[i]^="" and (state_array[i+1]="" or i=52) then
            put '{"state":"' state_array[i] +(-1) '"}';
           else if state_array[i]^="" then
                  put '{"state":"' state_array[i] +(-1) '"},';
    end;
    put ']}';
    if last.subcat_id then do;
        put ']}';
    end;
    else do;
        *put ',';
    end;
    if last.category_id then do;
        put ']}';
    end;
    else do;
        *put ',';
    end;
    if lastrec eq 1 then do;
        put ']';
    end;
```

```
    else do;
        put ',';
    end;
run;
```

This was a tedious process. After SAS 9.4 was released, PROC JSON simplified the code:

```
proc json out='\\kmcbi2\US_heatmap\src\MapControlOutput.json'
    pretty;
    export indicators2
    /  nosastags;
run;
```

PROC JSON greatly streamlines the process of converting a SAS data set to a JSON data structure.

CONCLUSION

The power of SAS/GRAPH can be used to custom-build applications to provide the user with a rich visualization experience. Procedures of the SAS language, including PROC IMPORT and PROC JSON, can be used to combine disparate data from Excel spreadsheets into a format that can be useful and analyzed effectively. This approach can be used for other data that could benefit from a heat map presentation.

REFERENCES

Carpenter, Art. 2004. ***CARPENTER'S COMPLETE GUIDE TO THE SAS MACRO LANGUAGE***, *Second Edition*. Cary, NC: SAS Institute Inc. Available at https://support.sas.com/pubscat/bookdetails.jsp?catid=1&pc=59224.

ACKNOWLEDGMENTS

The author would like to thank Robert Allison for his input on this project. His assistance enabled the project to take advantage of all PROC GMAP could offer. Dick Vohlers also contributed immensely to the effort to read data and format it.

RECOMMENDED READING

- *Base SAS® Procedures Guide*

CONTACT INFORMATION

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

Jim Bauer
SAS Institute Inc.
100 SAS Campus Drive
Cary, NC 27513
Jim. Bauer@sas.com