Demographic Data Monitoring System: Technology Used to Track Survey Quality

André Harper[1], Brian Dumbacher[2]
[1]U.S. Census Bureau, 4600 Silver Hill Road, Washington, DC 20233
[2]U.S. Census Bureau, 4600 Silver Hill Road, Washington, DC 20233

Abstract

During the 2012 calendar year, the U.S. Census Bureau underwent a realignment of its regional offices for operational efficiency; reducing their number from twelve to six. In order to monitor the quality of demographic survey data during this transitional period, an automated system was produced to display graphical and tabular data based on statistical models. The system, named the Demographic Data Monitoring System, was implemented within five months using rapid application development to ensure its availability by January, 2012. A variety of software options were evaluated, keeping in mind the aggressive timeline and evolving requirements. Senior management, including the U.S. Census Bureau director and deputy director, used the system to ensure the realignment did not unexpectedly impact demographic survey data.

Key Words: system development, rapid application development, data monitoring, statistical modeling

1. Introduction

For operational efficiency, the U.S. Census Bureau realigned its regional offices in the field. This reduced their number from twelve to six and involved changes in the geographic coverage and management structure of the remaining regional offices. The transition to the new management structure occurred in seven waves that started in January 2012 and ended in November 2012. In August 2011, in order to measure the impact of the regional office realignment on the quality of survey estimates, then director of the U.S. Census Bureau, Dr. Robert Groves, charged the Demographic Statistical Methods Division to develop a proactive monitoring system to detect daily data changes in real-time. The system needed to be in place by January 2012 to monitor data for the following surveys: the American Community Survey, the Consumer Expenditure Quarterly Interview Survey, the Current Population Survey, the National Crime Victimization Survey, the National Health Interview Survey, and the Survey of Income and Program Participation.

In five months, the Demographic Statistical Methods Division defined, planned, and implemented the Demographic Data Monitoring System (DDMS). The automated DDMS was comprised of server-side software and a web-based graphical interface that disseminated the output of the encompassed statistical models on a daily basis in the form of graphs and tables. It required close coordination between management, IT specialists, and statisticians in order to succeed.

Any views expressed are those of the authors and not necessarily those of the U.S. Census Bureau.

2. Demographic Data Monitoring Team

2.1 Tiered Team Structure

The team responsible for managing, designing, and implementing the DDMS was broken down into three tiers, as illustrated in Figure 1. Each parent team included the leads from its child team(s). The team leads were responsible for communicating statuses to the upper-level teams, as well as relaying information from senior-level meetings to their respective groups. This was effective in facilitating bilateral communication in every aspect of the tiered structure.

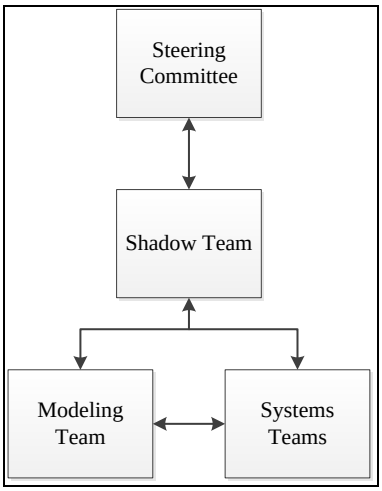


Figure 1: Demographic Data Monitoring Team Structure

The Steering Committee consisted of senior U.S. Census Bureau managers who provided the vision and guidance for the project. Below them was the Shadow Team, which was responsible for communicating project-related concerns and statuses to the Steering Committee. Additionally, the Shadow Team consisted of mid-level managers and sub-team leaders who provided oversight and support to the Modeling and Systems Teams.

The system development was accomplished through the joint efforts of the Modeling and Systems Teams. The Modeling Team, comprised of statisticians, developed statistical models to examine the effect of the new management structure on key variables for each survey. The Systems Team was comprised of IT specialists who secured data, researched software solutions, and developed an automated system to encompass the statistical models.

2.2 Sub-Team Collaboration

The Modeling and Systems Teams collaborated on multiple aspects of the project. Data dissemination, model integration, and system validation were among the processes that required joint efforts from both teams.

In order for the Systems Team to begin working on the display for the system, it was necessary to know what output to expect from the models. The teams met to discuss the output so the portion of the Systems Team disseminating the data could proceed with their development.

Both teams were involved in deciding how the data from the system would be disseminated. Team members researched data visualization text and attended a seminar by an expert in the field, Nathan Yau. Yau (2011) emphasized the importance of using interactivity, avoiding three-dimensional charts, and using a zero-centered axis. The Systems Team considered these guidelines when designing the DDMS.

In lieu of a formal software requirement document, which was not produced due to time constraints, the Systems Team engaged in brainstorming sessions and joint meetings with the Modeling Team. The developers had to remain flexible as the informal, oral objectives evolved and changed regularly. The outputs of the joint sessions were sketches of the graphs and charts needed for the final system, along with methods for how to incorporate the statistical models into the system. The Modeling Team had determined the most effective way to monitor the effect of the regional office realignment was to use 95% confidence interval bands centered at zero. Therefore, as the System Team finalized their software selections, they had to ensure it was capable of producing these desired graphs.

Integrating the statistical models into the automated system required modelers and IT staffs to work together closely. The Systems Team programmers initially reviewed the model code and worked with their counterparts on the Modeling Team to resolve any concerns. Multiple iterations of tests were performed to ensure the model performed as intended within the subsystem. Some issues were found due to the restrictions from running SAS batch processes on Linux machines. For example, when the Systems Team incorporated the models into the system, some code was incompatible with the Linux environment and required adjustments. The cause of this incompatibility was due to the Modeling Team writing their statistical models predominantly on PC-based SAS applications.

3. Software Selection

3.1 Front-End Software Evaluation

At the onset of the project, the Systems Team rapidly reviewed multiple software options over the span of a month for displaying the model output. The solution would need to be automated and quickly implementable, able to keep Census data secure, and capable of displaying the necessary graphical and tabular data. The software that the Systems Team evaluated consisted of Google APIs, Microsoft SharePoint, SAS Software Suite, and ORACLE APEX.

3.1.1 Google APIs

The Google APIs were dynamic enough to meet the display requirements. However, due to a limited timeline, the sheer number of available APIs to review, and the associated learning curve of the unfamiliar technology, the Systems Team decided early on not to pursue Google APIs as a solution. Learning curve aside, there were other concerns. Code for Google APIs was managed and maintained by Google. Therefore, the data would be transmitted outside of the Census firewall, which was a security risk. Additionally, there were potential limitations with its ability to automate the system as desired.

3.1.2 Microsoft SharePoint

The Systems Team did not have experience in Microsoft SharePoint. However, SharePoint's learning curve was likely less than Google API's. Microsoft SharePoint is a collaboration tool that is accessible via a web interface. It is optimized to integrate with

other Microsoft software like Microsoft Excel, which could be used for producing graphs. Additionally, Microsoft SharePoint offered a dashboard component that was a preferred feature.

However, Microsoft SharePoint was in its beginning stages at the U.S. Census Bureau at the time the Systems Team was researching software. Before a demonstration of the software capabilities was available, the project would have been two months from the January 1, 2012 deadline. Therefore, there would not have been enough time invested in Microsoft SharePoint or a sophisticated enough environment to deploy the application by its release date.

3.1.3 SAS Software Suite

Due to the heavy SAS usage within the U.S. Census Bureau, the Systems Team researched several SAS solutions. Of those, SAS Graph and JMP were readily available to all staffs. SAS Graph had a dashboard feature that was a desired component of the final software solution. Additionally, SAS Graph could produce graphs relatively easily and the expertise was readily available to the Systems Team. However, SAS Graph itself did not provide a solution for quickly distributing information to multiple recipients. SAS JMP could produce a variety of graphs. However, it was PC-based and could not be automated.

Therefore, the Systems Team looked into SAS products capable of using web-interfaces for displaying data. Of the SAS suite of software products, the Systems Team examined SAS Enterprise Business Intelligence (EBI), SAS Business Intelligence (BI), SAS Stored Process Web Application, and SAS IntrNet. Though SAS IntrNet was easier to install than other SAS web solutions, it was dated software. Therefore, the Systems Team directed its efforts towards the other solutions offered by SAS.

SAS Stored Process Web Application had a dashboard feature and could have potentially integrated within the division's existing structure. However, it called for the installation of over a dozen additional components on the division's servers with assistance from a SAS trained contractor. Therefore, with an aggressive schedule in mind, the Systems Team was unable to pursue this option further.

SAS BI and SAS EBI are similar products, with the latter being the more sophisticated and larger product. SAS EBI had a dashboard feature and could have satisfied the requirements of producing an automated, secure system for displaying graphs and tables. However, it would have been difficult to install and was incompatible with the existing server configuration.

SAS BI, the smaller of the two, was also capable of fulfilling the system's needs. It shared the same advantages of SAS EBI. Unfortunately, like other SAS products that were researched, the installation process was involved and, in the case of SAS EBI and SAS BI, required multiple contractors. Additionally, the recommended configuration required four servers, which drastically increased the potential complications associated with the setup procedure. Overall, in order to use SAS software for data visualization, the Systems Team would have needed to disrupt its current server environment, install a webserver, and/or seek out contractors.

3.1.4 ORACLE APEX

The last software option that the Systems Team researched was a product from ORACLE. ORACLE Application Express (APEX) is a component of the ORACLE relational database software that is designed to allow rapid application development through a web-based graphical user interface. It could produce polished web sites and had security features that allowed for controlling access to sites. It allowed for dashboard capability and had a relatively small learning curve. Though a working knowledge of SQL was necessary to use the basic functionality of the software, it included wizards to make creating web pages easier.

As an ORACLE product, ORACLE APEX seamlessly integrated with ORACLE databases. It had a built-in webserver, which minimized installation and configuration. In addition, because the relational database software had already been purchased, ORACLE APEX was available at no additional cost. Multiple members of the Systems Team were familiar with ORACLE and the division had two dedicated ORACLE database administrators. The Systems Team was able to quickly generate several webpages within a few days to give an example of the software's capabilities for a new ORACLE APEX user. As a result, the Systems Team was able to confirm the software met the graphical needs of the project.

As an additional advantage, ORACLE APEX used Adobe Flash to render graphics. Therefore, it was capable of producing sophisticated effects, including mouse-over effects and moving elements. For instance, on line graphs, the date and data values could be displayed when a user's cursor moved over them.

Though it was relatively easy to produce results with ORACLE APEX, no one in the division was familiar with the software and it was used sparsely within the U.S. Census Bureau. Therefore, should a problem arise in the system development, the Systems Team would be solely responsible for finding a resolution. Additionally, the software required opening a communication port that was outside of the recommended security guidelines for the U.S. Census Bureau servers.

Without advanced knowledge of the software, ORACLE APEX can be used only for display purposes. It would not be trivial to evaluate data once within the database. Therefore, a database design reflecting this limitation was required. Additionally, any form of customization beyond the preset wizards involved a proficiency in XML, SQL, PL/SQL, JavaScript, and/or HTML. In addition to these technologies, the developer would need to know exactly where the code would have to be placed in ORACLE APEX's multi-tiered, advanced web-interface.

With the advantages of this tool outweighing its risks, the Systems Team's final decision was to use ORACLE APEX. With it, the Systems Team could quickly develop an automated system capable of displaying the necessary graphical and tabular data. The information in the system would also be securely housed and restricted to staffs requiring access.

3.1.5 SAS/Excel Contingency

Due to the risk associated with unfamiliarity with ORACLE APEX, the Systems Team also developed a contingency option, which used SAS to create Microsoft Excel-based graphs depicting data quality for the surveys. SAS Base output the beta information to ranges within an Excel document. Due to the effort involved, this was only generated for

national-level data. Though it was designed as a contingency plan, it evolved into a distribution tool for allowing survey sponsors to see the status of the project.

3.2 Hardware and Back-End Software

The DDMS and its components spanned two IBM HS21 Blade Servers. Both servers were configured with the Red Hat Enterprise Linux Server release 5.8 operating system. One of the servers was a dedicated ORACLE 11g server, while the other contained all the remaining software components of the system. The back-end consisted of software and processes not directly related to displaying information to the end-user. Below is a brief description and rationale for each.

3.2.1 Structured Query Language

Structured Query Language (SQL) is a query language designed to manipulate data of enterprise level databases. The syntax for SQL varies based on the database distribution. The system used ORACLE compatible SQL to clear data from tables and data updating. SQL*Loader, an ORACLE utility designed for bulk loading of ASCII data, was used for daily insertions.

3.2.2 SAS

SAS is the U.S. Census Bureau standard software package for statistical data analysis. Therefore, the Systems Team used SAS data sets as a means to transport data to and from the Modeling Team for testing their statistical models. Additionally, SAS was used to collect the geographical data, such as tract and block, from the Demographic Statistical Methods Division's Sample Design database that was necessary to run the models successfully.

3.2.3 Perl

Perl 5 is a free, portable, scripting language readily available on U.S. Census servers. It has quick and efficient file handing capability as well as robust regular expression functionality. Perl was a suitable choice due to its availability, flexibility, and speed.

Perl was used for web scrapping web-based, summarized, field cost data for all regional offices. The collected HTML was parsed using regular expression matching and then summarized based on regional office level. The output PL/SQL code aggregated cost data based on the current and previous days within a month.

3.2.4 Kornshell

Most code throughout the system is embedded in or called from within a Kornshell (KSH) script. KSH is one of many shell scripting languages available on UNIX/Linux platforms. The Systems Team chose KSH over other shell languages, such as BASH or C-shell, because it is the standard used within the Demographic Statistical Methods Division.

KSH scripts allowed us to automate terminal commands, including calls to applications such as ORACLE and SAS. The main script that was executed daily by the batch query system (BQS) spawned several processes per survey. BQS is the standard batch tool used within the Demographic Statistical Methods Division and allows for easy viewing and manipulation of queue processing. An additional query mechanism, cron, was used for automated tasks run by non-system accounts. The cron utility is readily available on most Linux distributions and allows for scheduling task executions automatically based on time.

3.3 Server Configuration

In accordance with the Demographic Statistical Methods Division's secure directory policy, a directory structure locked down by Linux groups was generated for the DDMS to ensure only those with the need to know would have access to the raw, unedited Computer Assisted Personal Interviewing (CAPI) data from the field and system software. Subversion, the Demographic Statistical Methods Division's software configuration management tool, was used to store version history once the DDMS was in production. Additionally, any software changes required approval from the division's change control board.

4. Development

4.1 Rapid Application Development

Rapid application development was used in order to achieve the objective of producing a fully functioning system approximately three months after finalizing software selection. The teams went through multiple cycles of development and verification to ensure the fluid objectives were being satisfied. For example, as aforementioned, the Systems Team and Modeling Team met to discuss the graphical output. Following those meetings, the Systems Team was able to produce mockup webpages using Oracle APEX to validate the request was clear. This iterative process repeated until the display matched the desired output. Similar steps were done when selecting data input, validating data sets, and verifying system outputs.

Additionally, the Systems Team produced the necessary documents to comply with the Demographic Statistical Methods Division's IT development lifecycle including system designs and technical designs for both the front-end and back-end components of the system. The Systems Team was also able to design their database entity relationship diagraph, which was provided to the system database administrators in order to create the Oracle environment. Additionally, the database administrators enabled and configured the Oracle APEX product.

4.2 Database Design

An initial four-table design was used to store the beta estimates and associated information. A variable estimate table was the central table and contained the data output from the daily runs including the beta estimate, lower and upper confidence interval bounds, and date. In anticipation of a post-release enhancement, two additional attributes for mean values were included.

The variable estimate table had three related tables. One contained the details related to the regional offices, such as the city where the regional office is located. Secondly, the survey table contained information about the surveys in scope, including name and survey acronym. The final table contained data specific to the variables being modeled, including their description and associated survey.

4.3 Data Capture

The DDMS required multiple data inputs in order to operate effectively. This included raw field data, geographical data, and ZIP code data. The Technology Management Office serves as a hub for field data within the U.S. Census Bureau, collecting unedited field data and distributing it using their custom, Linux-based delivery commands. The

DDMS used these commands to automate the secure transfer of raw, unedited CAPI data collected by field representatives.

The Modeling Team informed the Systems Team what key variables and covariates were required for their survey-specific models. In turn, the Systems Team worked with the Technology Management Office to transmit variable request files ranging in length from 200 to over 1,300 variables. The range in variable request length was due to the number of person-level information and arrays necessary for each model. Variables common to all surveys included information about the regional office realignment and the field representative who handled the case. The Technology Management Office used the supplied variable lists to generate ASCII-based survey data for the Systems Team, who validated the data and converted it into SAS data sets.

There were variations in the format of data retrieved, which were handled within survey-specific subprograms. For example, in the case of the Survey of Income and Program Participation, the software used to collect field data was a Microsoft-based product that differed from the Linux-based software used by the other surveys. This resulted in a different format and structure of the received ASCII files that required additional coding to interpret.

Geographical data and sample design information were retrieved as well. For most surveys, this data was available in the Demographic Statistical Methods Division's Sample Design database. For the American Community Survey, the American Community Survey Office periodically provided geographical information in the form of SAS data sets. The Systems Team also made available tract-level demographic and economic data, while the Modeling Team obtained publicly available US Internal Revenue Service ZIP code-level tax data. The Systems Team appended raw field data, geographical data, case date, and ZIP code data to existing SAS data sets and then delivered them to the Modeling Team in the form of a cumulative data set.

4.4 System Data Flow

In Figure 2, a high-level system flow of the DDMS, which ran nightly to offer real-time survey quality results, is depicted.

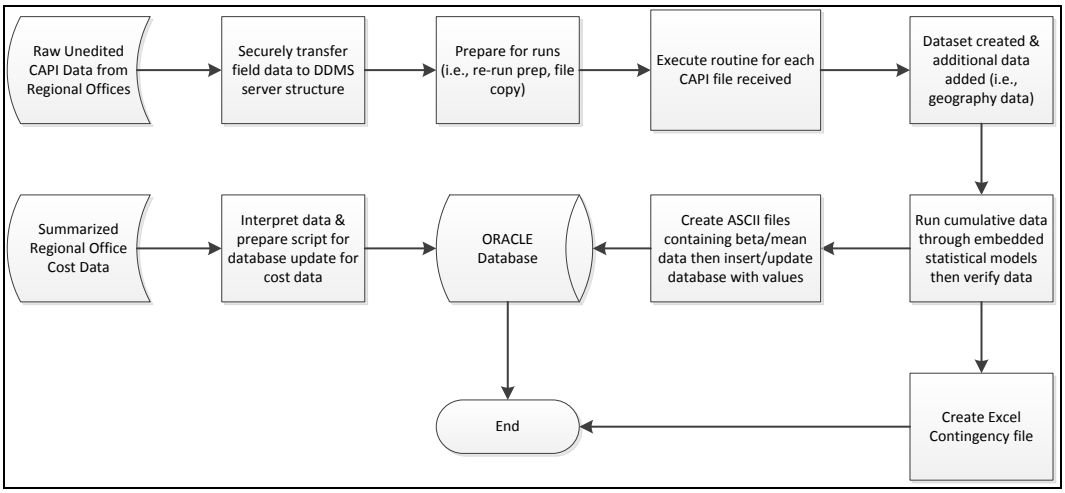


Figure 2: DDMS Data Flow

The back-end was broken into survey-specific subcomponents. The single parent process, which ran once daily, executed multiple child processes. The DDMS began with the parent process transferring raw, unedited CAPI survey data containing information input by the field interviews from each regional office. This ASCII data was transferred securely to division's servers where they were stored in survey-specific directories. The Systems Team received emails confirming that the files had arrived successfully.

The cumulative data sets were backed up for each survey in case the incoming date's data would need to be rerun for a particular survey. Reruns occurred for a variety of reasons, such as an embedded statistical model requiring an adjustment or issues connecting to the database. The cumulative data sets were updated later in the process following a successful conversion of the current day's file.

Lastly, the parent process spawned child processes for each ASCII file received. Therefore, each CAPI file received was run individually through survey-specific subsystems that interpreted the raw, unedited CAPI data.

Using SAS layouts, the ASCII field data was converted into individual SAS data sets that were placed into survey-specific directories for secure data storage. The layouts were created using a custom shell script that generated SAS compatible names and lengths for over thirty-five hundred received variables. A similar script then took those variables that were part of an array and produced compatible, human-readable names for the Modeling Team.

Geographic data was then appended to the newly created data sets. Additionally, the case's date was appended to each record based on the received ASCII filename, which was named uniquely using specific survey and date codes. Therefore, the CAPI filenames were difficult to decipher without an associated key to translate the survey and date codes.

At this point, the data for the current run date for each survey-specific file was added to its respective cumulative data set. This was needed for the statistical models as well as portions of the system that depended on the run date. The complete cumulative file for each survey was then run through the survey-specific embedded model.

Due to the high visibility of the project and the unedited nature of the received CAPI data, the Systems Team embedded basic data validation and quality control checks in the system before dissemination. The checks made sure the data received from the field had enough observations under the new management structure for model fitting and fell into acceptable value ranges called legal values. For instance, if the system encountered an age value greater than 120, an illegal value flag would be set and the system would suppress that survey's daily output and notify the Systems Team to take action in accordance with the established communication plan.

Because the models were fitted using cumulative data, it was expected that the t-statistics would not change much from one day to the next. To monitor this, the system included a t-statistic flag that was set to one if the absolute difference of that day's t-statistic and the previous day's t-statistic was greater than 0.5. If this t-statistic consistency check failed, the system was not halted. Instead, the Systems Team followed the communication plan and notified the Modeling Team.

Having successfully passed validation and model fitting, the next step in the process was to update the display system with the model output, which was a SAS data set that included survey code, regional office code, variable code, beta estimate, lower and upper confidence interval bounds, t-statistic, and date. In the experience of the Systems Team, Oracle is more compatible with ASCII files than SAS data sets. Therefore, the data sets were output into SQL*Loader control files. As the files were created, an additional Boolean was added to specify whether the beta estimate was within bounds. Stand-alone processes retrieved, evaluated, and summarized regional office cost data then prepared a script to update the database, which was run within the DDMS daily processing. The system flow concluded with the ORACLE SQL*Loader component parsing the SQL*Loader control file into the database.

5. Demographic Data Monitoring System Site

The DDMS was first released as a prototype since all necessary procedures involving multiple Census divisions were impossible to accomplish in the allotted time frame. The DDMS prototype, which became available on January 1, 2012, had over 320 beta estimate graphs and a table containing beta estimate data.

Figure 3: DDMS Homepage

It was designed to allow for quick maneuverability between pages that were divided by survey and national/regional office level. On the home page, as seen in Figure 3, all six surveys had a separate icon that led to survey-specific pages when clicked. Those survey-specific pages also had links; these allowed users to select a specific variable of interest and then view national-level or regional-level beta graphs. Only the six consolidating regional offices that were within project scope were graphed.

5.1 Beta Estimate Graphs

For each key variable, the system created a plot of beta versus time with confidence intervals centered at zero. There were multiple beta graphs for each key variable including those at the national level and for each of the six regional offices under the new management structure. Each key variable had a full-page, national-level graph. From here, a link led to a single, survey-specific page containing each variable's national-level graphs. A page with a similar design existed for the regional office graphs that contained all the regional office graphs for a particular variable. See Figure 4.

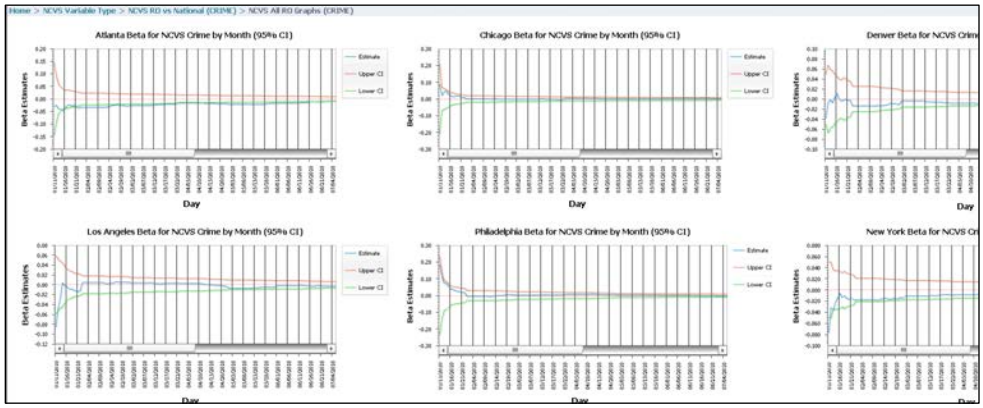


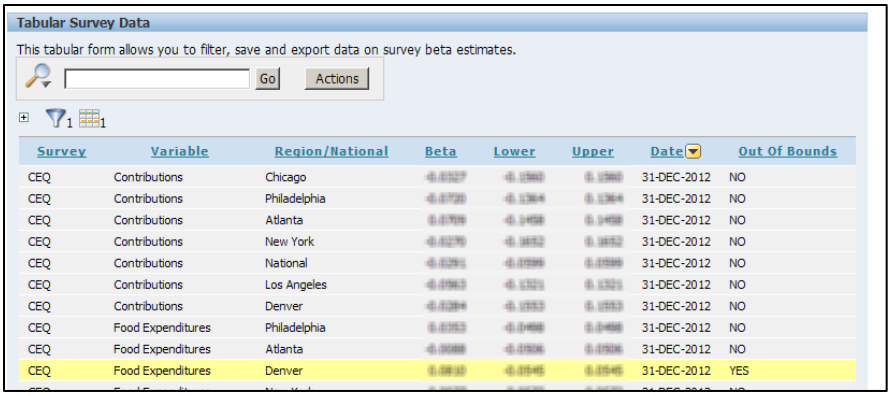
Figure 4: Regional Office Beta Estimate Graphs

The beta graphs used three SQL statements to access the ORACLE table containing the beta information. These separate queries defined the upper confidence bound, lower confidence bound, and beta estimate. When mousing over a particular day's plot for any of these three lines, that day's value was displayed in a rectangular label on top of the line. The graphs pulled from the table containing the model outputs in order to plot the accumulating data, which spanned calendar year 2012.

5.2 Tabular Data

The detailed information related to the estimate was available in a configurable table on the home page of the DDMS. The table included headings for: survey acronym, variable name, the regional office or national, the upper and lower confidence bounds, beta estimate, whether the beta estimate was out of bound, and the date.

Table columns were sortable and filterable. For example, it was possible to view the Consumer Expenditure Quarterly Interview Survey's data sorted by variable name, as seen in Figure 5. Additionally, the APEX table offered the ability to filter data further using date ranges or limiting output to the national level or a specific regional office.



Survey	Variable	Region/National	Beta	Lower	Upper	Date	Out Of Bounds
CEQ	Contributions	Chicago				31-DEC-2012	NO
CEQ	Contributions	Philadelphia				31-DEC-2012	NO
CEQ	Contributions	Atlanta				31-DEC-2012	NO
CEQ	Contributions	New York				31-DEC-2012	NO
CEQ	Contributions	National				31-DEC-2012	NO
CEQ	Contributions	Los Angeles				31-DEC-2012	NO
CEQ	Contributions	Denver				31-DEC-2012	NO
CEQ	Food Expenditures	Philadelphia				31-DEC-2012	NO
CEQ	Food Expenditures	Atlanta				31-DEC-2012	NO
CEQ	Food Expenditures	Denver				31-DEC-2012	YES

Figure 5: Filtered Tabular Survey Data

6. Postproduction Enhancements

Following the prototype stage and after the system was in a production environment, the Systems Team was requested to add a summary for all surveys on the front page of the site. An additional request was made to add vertical lines on the beta graphs to signify the beginning of a new wave in the regional office realignment. Also, a means graph for each key variable being monitored was requested. The last enhancement required incorporating the means graphs with a cumulative cost graph for each survey.

Within the ORACLE APEX software, SQL, PL/SQL, JavaScript, and XML modifications were necessary to accomplish the postproduction enhancements. PL/SQL and SQL queried the necessary information from the database tables and loaded the information into session variables. XML controlled the manner in which the graphs were rendered and contained stored session variables. It also set animation styles and the details that were displayed when a user interacted with the graphs. JavaScript code interpreted session variables and reloaded dynamic pages.

6.1 Summary Statistics

The daily reports to the U.S. Census director, deputy director, and upper management included summary statistics for the number of cases that were out of bounds. The Systems Team was requested to display this information in an easily discernible fashion. Initially, a tabular layout was designed for this information; however, because the System Team's knowledge of the ORACLE APEX software grew, the option to produce an animated bar graph became available. As seen in the middle of Figure 3, after logging into the system, two bar graphs appeared; one showed summary information at a national level, and the other showed data at a regional office level.

When the homepage loaded, the bars on the summary statistic graph animated vertically from the x-axis until they reached their final height based on their value. Each survey had four bars of summary data that showed the number of observations out of bounds for that day, the total number of observations for that day, the total number of observations out of bounds today, and the total number of observations out of bounds since monitoring began. Totals for each of these summary breakdowns were beside their descriptions in the legend. Survey-specific values within the summary statistics graph were displayed on top of the bars. Only those surveys that received data most recently appeared in the summary

statistics graphs. For example, the Current Population Survey was displayed only during the approximately ten-day period when data for the survey was collected in the field.

Mousing over one of the bars showed the survey acronym and number of observations for the particular bar selected in a rectangular pop-up. Mousing over the legend highlighted all applicable items on the graph. For example, moving one's mouse over "Total Obs" in the legend highlighted all bars corresponding to the total number of observations.

6.2 Means Graphs

The statistical means were incorporated as the second enhancement to the system in order to compare a variable month-by-month. Each variable had a corresponding line graph with a separate line and unique color for each month. Therefore, as seen in Figure 6, the DDMS depicted multiple months on a single graph. The x-axis on the graph represented date, and the y-axis represented mean, so each point displayed the mean value for the cases interviewed up to that date. The key variables for most surveys were 0/1 indicators, so the means displayed on these line graphs could be interpreted as cumulative proportions. The exception is the Consumer Expenditure Quarterly Interview Survey, whose key variables were all continuous. For example, the food expenditures variable was the average amount spent on food during a weeklong period.

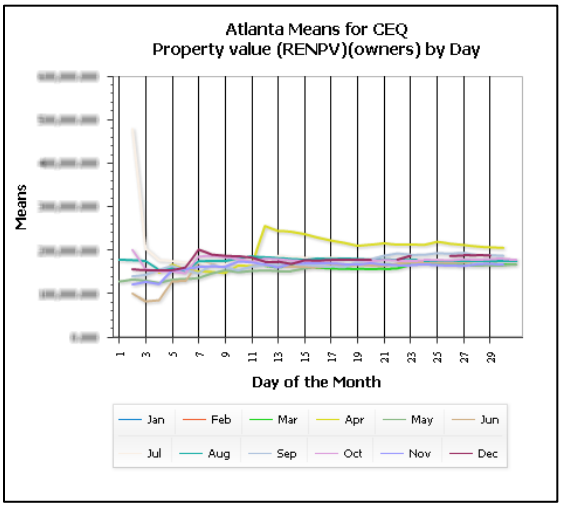


Figure 6: Means Graph

The legend consisted of months January through December for the 2012 calendar year. Mousing over a month highlighted the corresponding month's line in the graph. Mousing over the line within the graph highlighted the line and displayed the value of the mean for the day selected.

Most surveys produced means on a daily basis, in a similar manner as the beta estimates. However, the means graphs for the Current Population Survey were updated in the DDMS only at the beginning of the following month. For example, July data was available after the first August run. This was due to the potential employment information that could be ascertained from the means data.

6.3 Wave Markers

In order to differentiate between the waves and better interpret the graphical output from the models, the Systems Team fulfilled a request from the director to add vertical lines on

the day each wave began. The yellow wave markers intersected the beta estimate lines and the upper and lower confidence bounds. An example of this can be seen in the national-level beta graph shown in Figure 7. Beneath the days listed on the x-axis, the waves were listed by number.

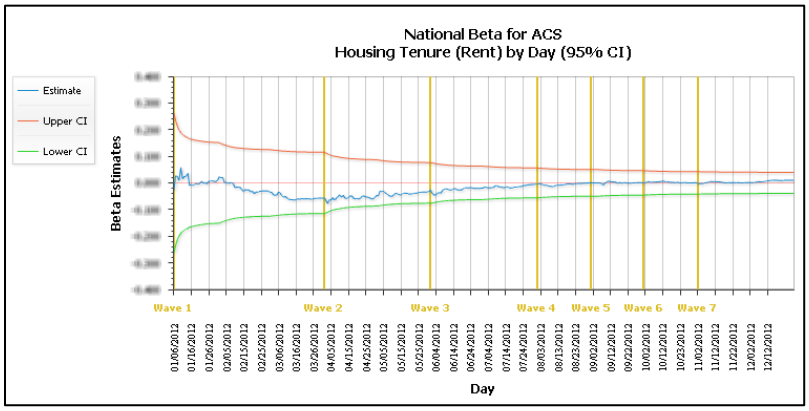


Figure 7: National Beta Estimate Graph

6.4 Cost-Means Graphs

The final addition to the DDMS was the cost-means graphs. These graphs examined the relationship between the statistical means and the total expenses associated with conducting a specific survey. These graphs showed monthly results. However, unlike the means graph, these graphs showed a single mean line per graph.

The cost-mean graph displayed the cost data as a bar graph behind the mean line graphs. Mousing over the means lines behaved identically to the means graph described previously. Interacting with the bar lines would display the day and cumulative cost for the survey up to and including the date found on the x-axis as shown in Figure 8 below.

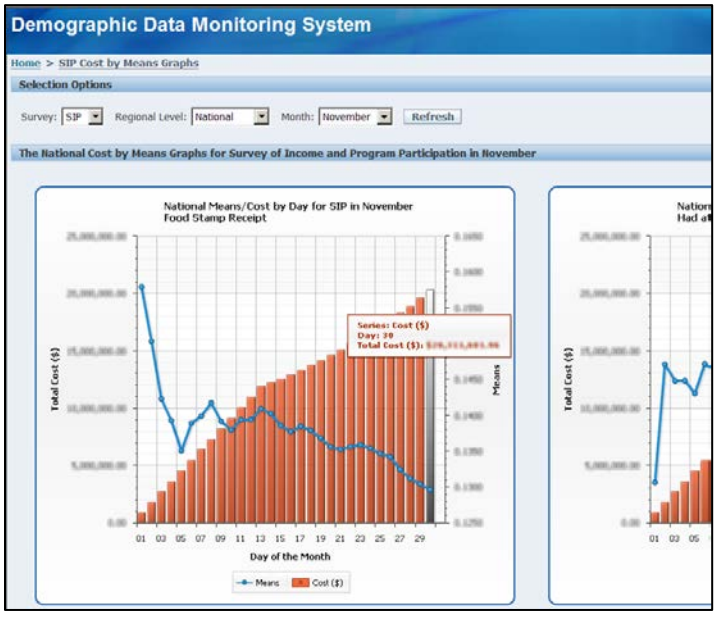


Figure 8: Cost-Means Graph

Unlike the other graphs, the cost-means graphs were dynamically produced. By the time this request was fulfilled, the Systems Team had acquired the knowledge for the APEX tool necessary to produce more sophisticated graphs and pages. The cost-means single webpage was driven by session variables that were set using database querying.

Graphs, labels, and navigation elements were determined by these session variables. The session variables were set when the drop-down menus on the page changed or based on the navigation path taken by users. Prior to navigating to the cost-mean page, session variables were set to ensure the appropriate survey page was displayed.

7. Future Uses

The DDMS succeeded in fulfilling its purpose of monitoring the effects of the regional office realignment on demographic survey data, but it could serve as a template for similar systems in the future. The DDMS could be adapted to monitor the effects of transitional periods in general, or more specifically, changes to sample design and data collection. Consequently, the Demographic Statistical Methods Division is planning to implement a similar system during the upcoming decennial redesign of its demographic surveys in order to be more proactive in addressing survey sponsor concerns (Rottach, Sundukchi, & Toribio, 2013). This redesign involves selecting sample in new areas, training new field representatives, and switching from sample selection every ten years to sample selection every year from a continually updated Master Address File (Nguyen & Gerstein, 2011). A similar team dynamic may be created where statisticians and IT specialists work cohesively in order to produce a high quality monitoring system.

Acknowledgements

Thank you to the Demographic Data Monitoring Project Team members and those who supported our efforts. The primary author, who served as sub-team leader, would like to share a special thank you with the Systems Team members for their exceptional work and dedication in producing the DDMS.

References

Nguyen, T. & Gerstein, A. (2011). Sample Design Research in the 2010 Sample Redesign. *Proceedings of the Section on Survey Research Methods, American Statistical Association* (pp. 4359- 4372). Miami, FL.

Rottach, R. Sundukchi, M. & Toribio, N. (August, 2013). *Challenges Faced in the Daily Modeling of Survey Responses*. Unpublished paper presented at The Joint Statistical Meetings, Montréal, Québec Canada.

Toribio, N. & Rottach, R. (August, 2013). *Monitoring Key Estimates and Costs from the National Health Interview Survey throughout the Realignment of Census Bureau Regional Offices.* Unpublished paper presented at The Joint Statistical Meetings, Montréal, Québec Canada.

Yau, N. (2011). *Visualize This: The Flowing Data Guide to Design, Visualization, and Statistics*. Indianapolis: Wiley Publishing.