Paper 1493-2014

Experiences in Using Academic Data for SAS® BI Dashboard Development

Evangeline Collado, University of Central Florida; Michelle Parente, University of Central Florida

ABSTRACT

Business Intelligence (BI) dashboards serve as an invaluable, high-level, visual reference tool for decision-making processes in many business industries. A request was made to our department to develop some BI dashboards that could be incorporated in an academic setting. These dashboards would aim to serve various undergraduate executive and administrative staff at the university. While most business data may lend itself to work very well and easily in the development of dashboards, academic data is typically modeled differently and, therefore, faces unique challenges. In this paper, the authors detail and share the design and development process of creating dashboards for decision making in an academic environment utilizing SAS® BI Dashboard 4.3 and other SAS® Enterprise Business Intelligence 9.2 tools. The authors also provide lessons learned as well as recommendations for future implementations of BI dashboards utilizing academic data.

WHO WE ARE

The mission of Enterprise Decision Support (EDS), a division of Institutional Knowledge Management (IKM), is to provide data integration services and actionable information solutions to support executive and operational decision-making and planning at the University of Central Florida (UCF) through the delivery of business intelligence applications and other knowledge management tools. The EDS team delivers data and information in various formats tailored to user need and technical aptitude, and is responsible for system support, administration, and security of the university data warehouse and functional reporting data marts.

INTRODUCTION

BI dashboards are useful tools when there is a need to display myriad information, such as key performance indicators (KPI), metrics, tables, and charts in a single screen. The EDS team has delivered information to our constituency through the use of SAS® Web Report Studio reports that are delivered through the SAS® Information Delivery Portal, SAS® Stored Processes that are executed via the portal and return data to an MS Excel spreadsheet or Adobe PDF document using the Output Delivery System (ODS), and static content available through a link in the portal. We had not used the SAS BI Dashboard tool extensively in the past but various departments across campus have recently come to us asking if we could develop something for them to highlight specific information and to track metrics over time so we decided that a dashboard could effectively fit their need. This paper will discuss two such requests and the challenges we faced using academic data to deliver useful actionable information for effective decision making at this major metropolitan university.

PROJECT 1 - RETENTION REPORT DASHBOARD

BACKGROUND

One important metric for higher education is student retention and progression. Measures are calculated for each cohort based on the number of students who enroll each subsequent fall term (retention), earn a degree (graduation), and do not return to the university (attrition). For many years EDS has been providing retention and progression rates to our user community in tabular reports (Figure 1). The table highlights the counts of students from the initial cohorts that have been retained at the university, graduated from the university, or have not returned to the university for enrollment and/or completion of the degree program.

A rate is calculated based on the count of students initially entering the cohort. Although effective to show the actual counts and percentages there was no measure to quickly indicate significant rises or falls in our rates over time. It was decided to track the difference in the percentage from one cohort year to the next for each year of progression and highlight this difference in some fashion to serve as an alert.

University of Central Florida
Undergraduate Retention and Progression Report
Summer-Fall Full-Time FTIC Cohort
University Total - Male and Female - All Ethnic Groups

Chrt Yr	Chrt Cnt	One Year			Two Years			Three Years			Four Years			Five Years			Six Years			Seven Years			Eight Years		
		Still Enroll	Grad	Attrit Tot	Still Enroll	Grad	Attrit Tot	Still Enroll	Grad	Attrit Tot	Still Enroll	Grad	Attrit Tot	Still Enroll	Grad	Attrit Tot	Still Enroll	Grad	Attrit Tot	Still Enroll	Grad	Attrit Tot	Still Enroll	Grad	Attrit Tot
2004-2005	5722	4,747	0	975	4,236	9	1,478	3,894	131	1,697	1,937	1,972	1,813	589	3,287	1,846	247	3,617	1,858	115	3,738	1,869	88	3,787	1,847
		83.0%	0.0%	17.0%	74.0%	0.2%	25.8%	68.1%	2.3%	29.7%	33.9%	34.5%	31.7%	10.3%	57.4%	32.3%	4.3%	63.2%	32.5%	2.0%	65.3%	32.7%	1.5%	66.2%	32.3%
2005-2006	6025	4,941	0	1,084	4,429	11	1,585	4,175	109	1,741	2,111	2,018	1,896	622	3,428	1,975	252	3,784	1,989	145	3,912	1,968			
		82.0%	0.0%	18.0%	73.5%	0.2%	26.3%	69.3%	1.8%	28.9%	35.0%	33.5%	31.5%	10.3%	56.9%	32.8%	4.2%	62.8%	33.0%	2.4%	64.9%	32.7%			
2006-2007	6379	5,353	0	1,026	4,856	12	1,511	4,576	112	1,691	2,249	2,253	1,877	647	3,788	1,944	264	4,152	1,963						
		83.9%	0.0%	16.1%	76.1%	0.2%	23.7%	71.7%	1.8%	26.5%	35.3%	35.3%	29.4%	10.1%	59.4%	30.5%	4.1%	65.1%	30.8%						
2007-2008	6345	5,427	0	918	4,971	11	1,363	4,628	143	1,574	2,356	2,276	1,713	699	3,878	1,768									
		85.5%	0.0%	14.5%	78.3%	0.2%	21.5%	72.9%	2.3%	24.8%	37.1%	35.9%	27.0%	11.0%	61.1%	27.9%									
2008-2009	6134	5,343	0	791	4,889	18	1,227	4,517	185	1,432	2,132	2,461	1,541												
		87.1%	0.0%	12.9%	79.7%	0.3%	20.0%	73.6%	3.0%	23.3%	34.8%	40.1%	25.1%												
2009-2010	6237	5,409	0	828	4,998	20	1,219	4,645	197	1,395															
		86.7%	0.0%	13.3%	80.1%	0.3%	19.5%	74.5%	3.2%	22.4%															
2010-2011	6011	5,246	1	764	4,773	23	1,215																		
		87.3%	0.0%	12.7%	79.4%	0.4%	20.2%																		
2011-2012	6139	5,387	0	752																					
		87.8%	0.0%	12.2%																					

Figure 1. Retention and Progression Report

CHALLENGES AND OUTCOMES

Data Sources

The retention database is modeled to track a student for ten years and capture various information such as degree awarded, if the student graduated in the past three semesters, and program information, if the student is enrolled the current fall term. There is one row per student per cohort year and each row contains a series of columns for each retention year. We categorize students by their type such as First Time in College (FTIC), Undergraduate Transfer from a Florida College System school, or any other type of Undergraduate Transfer student. We also track what term a student enters a cohort and his/her full-time or part-time enrollment status. Due to the nature of this data source it was not easy to create indicator data objects in the dashboard tool. We would need to subset the data for the cohort of interest and then calculate the percent difference using the SAS function DIF(), which “computes the differences between data values and one or more lagged (shifted) values for time series data”. Since this function is not available in a SAS® Information Map or SQL query indicator data object within SAS BI Dashboard it was decided the appropriate data source for this indicator would be a stored process that would run and create the data set containing the aggregate measures for counts, percentages, and percentage differences when the dashboard is loaded.

Indicator Data and Indicator Development

The stored process program uses a series of %DO loops within a %MACRO to run through each of the most recent ten cohort years, and each of their ten retention years, and calculate counts, percentages/rates, and the difference in percentages. To make the data set that is created when the stored process executes available for indicator data code was added to generate and publish a package.

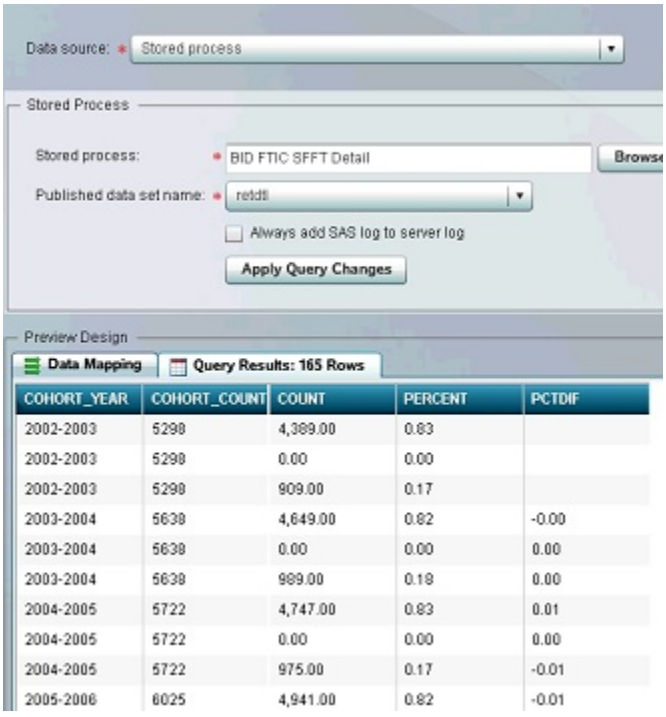
```
/* Create package for data set */
%let _archive_path=%sysfunc(pathname(WORK));
data _null_;
   rc = 0;
   pid = 0;
   desc = 'Retention Detail';
   nameValue = '';
   call package_begin(pid,desc,nameValue,rc);
   call insert_dataset(pid,"WORK","retdtl","Recent_Retention_Detail",'',rc);
   length fullpath $4096;
   call package_publish(pid, "TO_ARCHIVE", rc, "archive_path, archive_name,
   archive_fullpath","&_archive_path", "retdtl", fullpath);
   call symput('_ARCHIVE_FULLPATH', fullpath);
   call package_end(pid, rc);
run;
```

The code was placed inside the macro block of code and the stored process was tested. This presented a challenge that took a long time to overcome. The stored process executed perfectly without any error when tested by itself. The log indicated the physical path for the data set and when located and reviewed it contained the data as expected. However, the dashboard application produced an error (Figure 2) when the stored process was selected as the indicator type.

Figure 2. Dashboard Error

After several months, and with the aid of various SAS Technical Support personnel, we discovered that the package creation code had to be outside of the macro after the %MEND statement. None of us initially thought that would be an issue so we pursued other solutions. As soon as we made that change the data set was immediately available for selection.

The next challenge we faced had to do with the display of the values within the indicators. The data set stored values as shown in Figure 3. Although some formats are available with the dashboard tool the needed formats were either not available or did not apply when selected. For example, a value could be formatted as a percent but there was no way to specify how many decimal places to display.



COHORT_YEAR	COHORT_COUNT	COUNT	PERCENT	PCTDIF
2002-2003	5298	4,389.00	0.83	
2002-2003	5298	0.00	0.00	
2002-2003	5298	909.00	0.17	
2003-2004	5638	4,649.00	0.82	-0.00
2003-2004	5638	0.00	0.00	0.00
2003-2004	5638	989.00	0.18	0.00
2004-2005	5722	4,747.00	0.83	0.01
2004-2005	5722	0.00	0.00	0.00
2004-2005	5722	975.00	0.17	-0.01
2005-2006	6025	4,941.00	0.82	-0.01

Figure 3. Data Set Values

To overcome this challenge custom user-defined picture formats were created, new variables formatted as such were created, and then the new variable was used as the label for the original value.

```
/* Create picture formats for numeric variables */
proc format;
   picture pctfmt (round) low-<0 ='009.9%' (prefix='-' mult=1000)
                          0-high='009.9%' (mult=1000);
   picture numfmt low-high = '00,009';
run;
```

Thus, after using these new formats our spark table indicator was able to display the rates and differences in the desired format (Figure 4). We now have a table with counts and rates as in the old report but we are able to show at-

a-glance whether the difference from one cohort year to the next for a particular progression year has remained unchanged, decreased, or increased significantly to our specifications (+ or - 0.5%).

Cohort Year	Cohort Count	Progression Year	Progresssion Type	Progression Count	Progression Rate	Prog Rate Difference
2002-2003	5,298	First Year	Retention	4,389	82.8%	. ▲
2002-2003	5,298	First Year	Graduation	0	0.0%	. ▲
2002-2003	5,298	First Year	Attrition	909	17.2%	. ▲
2003-2004	5,638	First Year	Retention	4,649	82.5%	-0.4% =
2003-2004	5,638	First Year	Graduation	0	0.0%	0.0% =
2003-2004	5,638	First Year	Attrition	989	17.5%	0.4% =
2004-2005	5,722	First Year	Retention	4,747	83.0%	0.5% ▲
2004-2005	5,722	First Year	Graduation	0	0.0%	0.0% =
2004-2005	5,722	First Year	Attrition	975	17.0%	-0.5% ▼
2005-2006	6,025	First Year	Retention	4,941	82.0%	-1.0% ▼
2005-2006	6,025	First Year	Graduation	0	0.0%	0.0% =

Figure 4. Spark Table Indicator

Dashboard Design and Development

One of the great features of a dashboard is the ability to have multiple indicators in one report that encompasses one screen when viewed. Having a table with visual indicators for the change in percentage rate over time was not enough. The SAS BI Dashboard software contains a large number of indicator types to select from. However, many were not applicable to this particular data set. We decided to add a bar chart that would highlight this information clearly by having the color of the bar signify the difference in the percentage rate. Unfortunately, there was no way to include a legend in the indicator to showcase this fact. By adding prompts configured as client-side filters, and setting up the appropriate interactions, each of the indicators would change as the values were selected from the drop-downs (Figures 5 and 6).

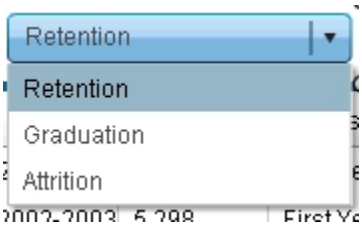


Figure 5. Progression Type Filter

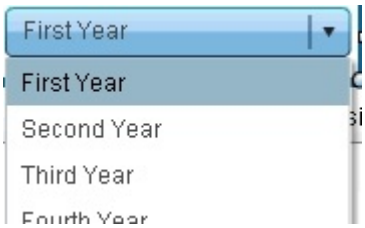


Figure 6. Progression Year Filter

The Retention Dashboard (Figure 7) was created as a proof of concept to explore the addition of adding functionality to our existing retention reports. Additional indicators will be added in the future and linked reports are a possibility as they offer the opportunity to show more detail when a value is clicked in an indicator. We can explore how our data fits into the myriad indicator types available in the software to see which offer the best visualizations. We want to conform to best practices yet offer our user community an easy to read and understand tool that provides information they can act upon to achieve the goals of the university. For retention data, the goal is to improve student retention and graduation rates. By understanding how these rates change over time we can create solutions to improve these measures.

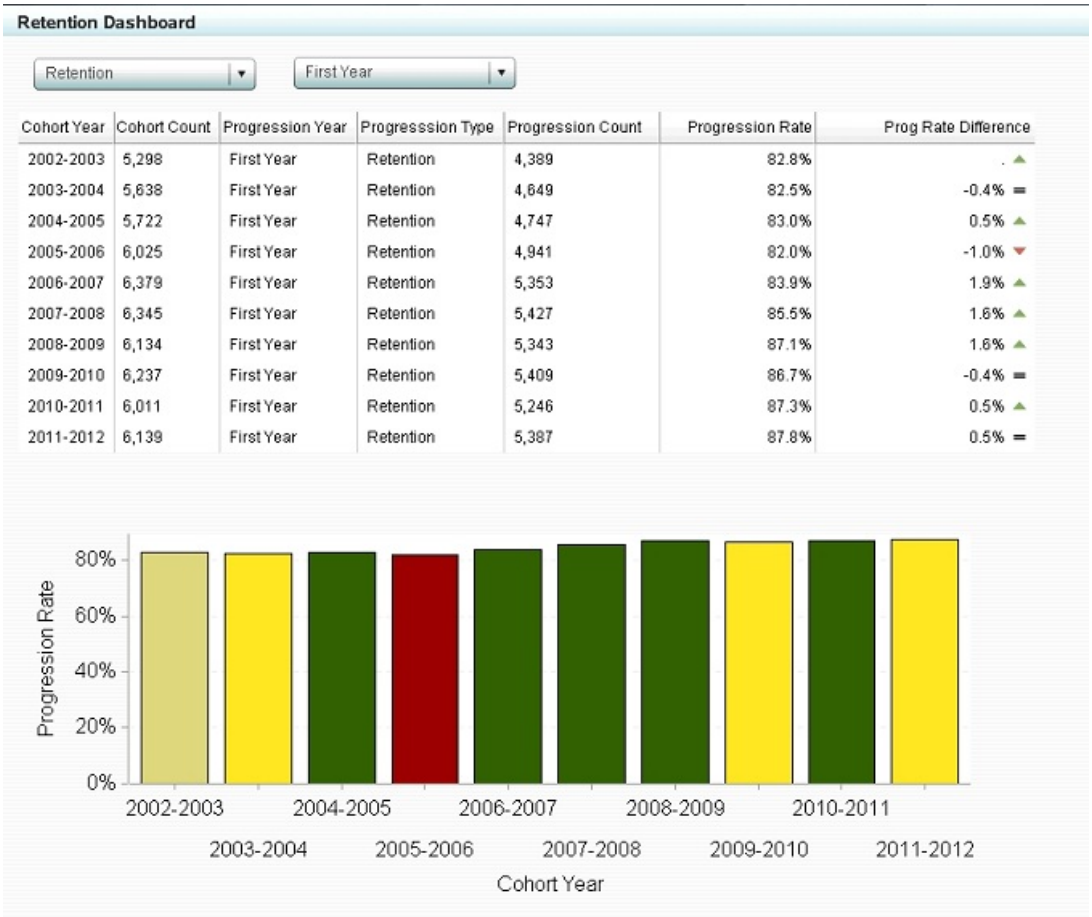


Figure 7. Retention Dashboard

PROJECT 2 - UNDERGRADUATE RESEARCH DASHBOARD

BACKGROUND

For the past three years the Office of Undergraduate Research (OUR) has been collecting data on students who participate in undergraduate research. With the amount of data growing, they wanted to be able to query and create reports. The reports would allow administrators to review student data and track the population of undergraduates engaged in research at the University of Central Florida. The reports would also aid executives and administrators in obtaining a sense of the distribution of undergraduate researchers in order to make decisions on involvement goals and assessments. We decided the needs of this project would be a good candidate for implementing a dashboard. Up until this point we had not created a fully functioning and complete dashboard. We began developing a proof of concept for OUR utilizing the SAS BI Dashboard.

CHALLENGES AND OUTCOMES

Data Sources

An initial dataset was provided by OUR at the start of the project. Since the data was manually collected and entered into an MS Excel spreadsheet over the span of three years by different people, an intense amount of data cleansing was needed in order to prepare the data and remedy data inconsistencies and data error entries. The dataset contained student information such as the year and term they were involved in research, the program and college in which they were enrolled, the program of research with which they were affiliated, and the mentor who guided them in their research. In order to use information from one data source and obtain additional demographic information about the students, student data was obtained through our data warehouse and joined to the initial dataset by an Emplid (a student identifier). Our data warehouse contains the official state vetted data and various data relating to the university. This process remedied most of the inconsistencies. Several other techniques including various functions

and SAS program's developed in SAS® Enterprise Guide were used to further create a consistent final table for the SAS BI dashboard that contained reasonable data.

Indicator Data and Indicator Development

After the creation of our final table for the dashboard, we ran into our first challenge with creating our indicator data objects. Most of our student data with the exception of Grade Point Averages (GPAs), DateTime and other calculated variables are character data types even though the text itself is numeric. Two such examples are variables such as Emplid and Birth Year. The character data types made aggregating the data difficult in the indicator data build since many of the aggregation functions in the indicator data work area require a numeric data type.

Another challenge we faced, as discovered in the Retention Report, was that the data structure and nature of our student data does not lend itself to work very well in the dashboard application used to build indicator data objects. Student data is multifaceted. Performing aggregations without first filtering the attribute of interest is difficult due to the number of different attributes that a student has. There are multiple rows for a student depending on what is being filtered. For example, part of the dataset contains variables for Year, Semester (spring, summer, fall), and Academic Levels (freshman, sophomore, junior, senior). If the Semester column is included in the count for Academic Levels by Year, then a freshman may be counted multiple times depending on how many semesters they were a freshman.

To address the challenges and create a usable indicator data for an indicator, we thought of three possible solutions. First, convert certain variables such as Birth Year to a numeric type (if possible). Second, create numeric columns to serve as a "counter" for summing or counting the observation rows of unique students. Lastly, filter, breakout, and aggregate the final table in SAS Enterprise Guide prior to using it to build the indicator data objects.

The solutions outlined would require the need to write customized expressions and perform querying procedures not available in the indicator data building workspace. This meant we had to create data tables as aggregated as possible prior to creating our indicator data objects.

To begin, we identified three key areas for the dashboard: student demographics, enrollment demographics, and program performance (Figure 8). Next, in SAS Enterprise Guide, we broke out and created subsets of the final table according to the areas identified. For the different components in each key area, we generated an aggregated table.

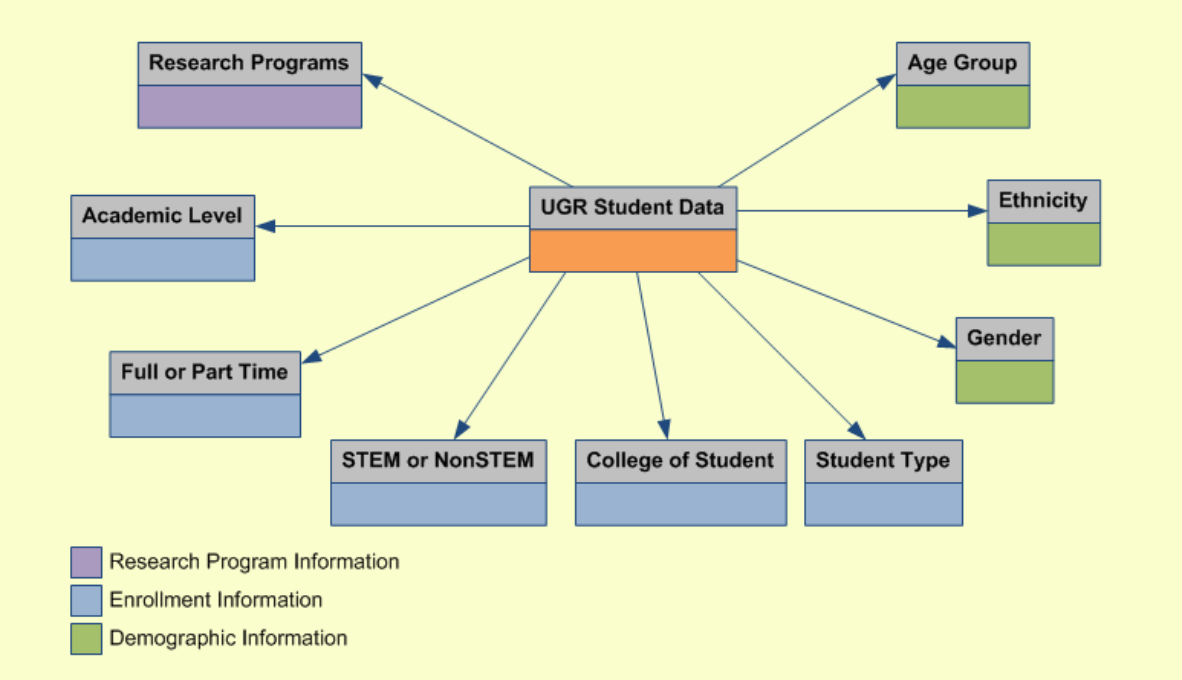


Figure 8. UGR Student Data Table Breakout

In creating the subset tables, we were able to generate usable indicator data objects for the indicators. For example, to create an indicator displaying Age Groups by Year, we filtered what we needed from the main table such as Year, Emplid, and Birth Year. Then, a new advanced column with the following code below was used as an expression in calculating the age.

```
INPUT((substr(left(t1.Year),1,4)),4.)- INPUT(t1.BIRTH_YEAR,4.)
```

From here, we could either group the Age before outputting the table or convert the column to a numeric type and perform a count function in the indicator data building workspace.

As with the Retention Report, we encountered some front-end formatting issues with the data displaying properly in the indicators. For example, in our percentage breakdown of gender and ethnicities, although the percentages were displaying properly in our indicator data, it was not reflecting in the indicators themselves (Figure 9). The decimal representation of the percentage would display instead of the proper percentage format. With a tip from a SAS representative at SAS Global Forum 2013 who told us this was a known problem in the dashboard tool, we were able to remedy the issue and properly display the correct percentage format (Figure 10).

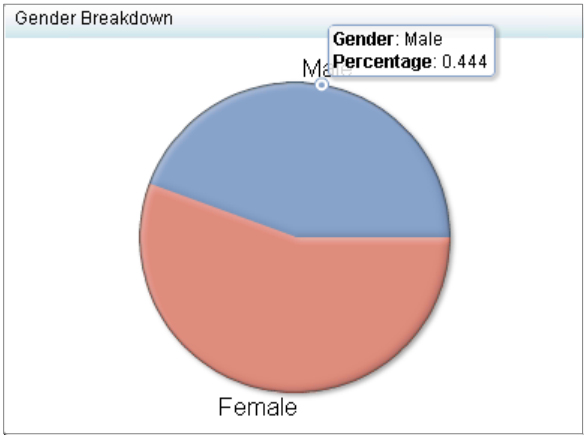


Figure 9. Incorrect Percentage Format

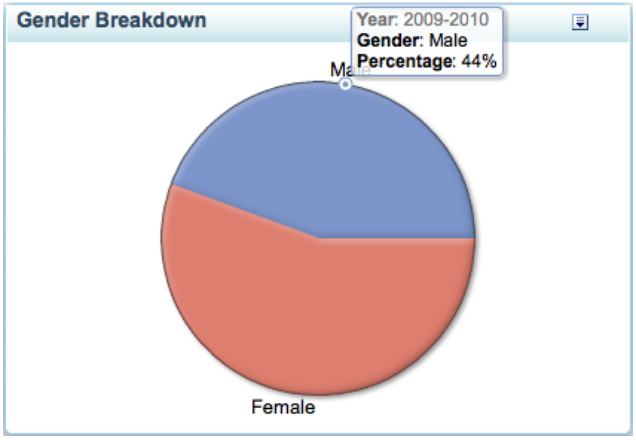


Figure 10. Corrected Percentage Format

The solution was to convert the percentage value to a character data type. Then, point the label for the variable containing the percentage in the indicator data to the newly created character variable representation of the percentage.

In the example of our gender and ethnicity pie charts, we created an advanced calculated column and used the following code in the expression builder. The code takes the numeric decimal and converts it to a character value to break out the % sign. Then after converting the value to a numeric type and multiplying 100 to get the percent value, the value is converted back to a character data type and the % sign is added back in.

```
PUT((INPUT(PUT(t1.Percentage,4.2),best4.2)*100),4.)||'%'
```

This would duplicate the numeric percentage in a character format (Figure 11). Then we modified the Data Mapping to point the label for the Percentages to our new character data type column Pct_char (Figure 12). The resulting indicator then properly displayed the percentages as expected.

Data Mapping | Query Results: 6 Rows

Year	Gender_Code	Gender	NumofStudents	Percentage	pct_char
2009-2010	F	Female	1003	56%	56%
2009-2010	M	Male	801	44%	44%
2010-2011	F	Female	935	54%	54%
2010-2011	M	Male	802	46%	46%
2011-2012	F	Female	851	53%	53%
2011-2012	M	Male	759	47%	47%

Figure 11. Query Results Preview Indicator Data Build with New Column Added

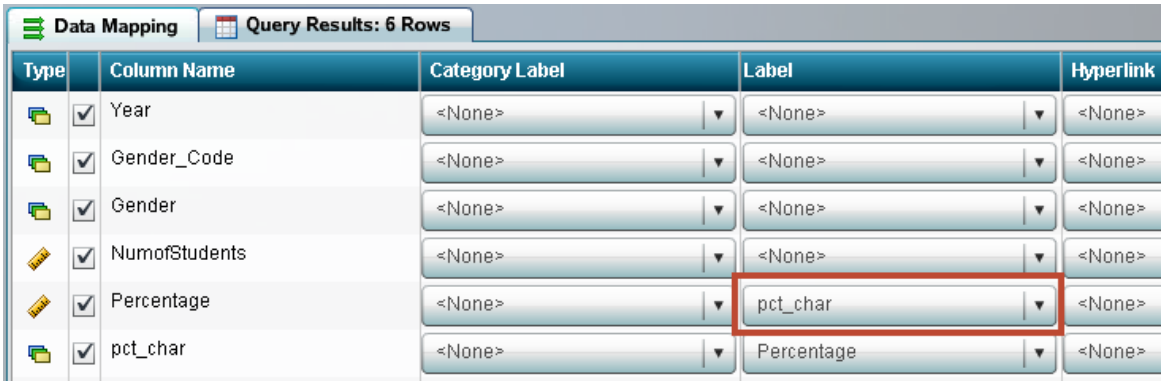


Figure 12. Highlight of Label Change in Numeric Column to New Character Column

We performed a similar procedure in a spark table indicator except we had to find a way to address percentage values which had null values included. To address the null values we wrapped the same code used above in a case statement before generating the resulting table:

```
CASE
  WHEN t1.EXCEL_Pct > 0
  THEN PUT((INPUT(PUT(t1.EXCEL_Pct,6.3),BESTD6.3)*100),6.2)||'%'
  ELSE '.'
END
```

The code would perform the same function as described in the scenario above, but would also only perform the function if there was a percentage value in the cell. Figure 13 displays the spark table prior to the creation of the character percentage column and Figure 14 displays the spark table after pointing the label for the percentage column to the new character percentage column.

Yearly Count By Research Programs*

Year	EXCEL	IDS	YES	HIM	McNair	RAMP	PURE	STEAM	STEM	ICUBED	LEARN	ORC	SURE
2009-2010	0.0351	0.386	0.0057	0.2093	0.0107	0.0372	0.0068					0.263	0.0462
2010-2011	0.0328	0.3762	0.0186	0.1845	0.0268	0.0578	0.0089	0.0064	0.0014	0.0004		0.2537	0.0325
2011-2012	0.0279	0.3076	0.018	0.2102	0.0232	0.1022	0.0066	0.0004	0.0162	0.0092	0.0206	0.2345	0.0235

** Students who participate in more than one program will be counted in each program respectively.*

Figure 13. Spark Table Indicator with Incorrect Percentage Displayed

Yearly Count By Research Programs*

Year	EXCEL	IDS	YES	HIM	McNair	RAMP	PURE	STEAM	STEM	ICUBED	LEARN	ORC	SURE
2009-2010	3.51%	38.60%	0.57%	20.93%	1.07%	3.72%	0.68%					26.30%	4.62%
2010-2011	3.28%	37.62%	1.86%	18.45%	2.68%	5.78%	0.89%	0.64%	0.14%	0.04%		25.37%	3.25%
2011-2012	2.79%	30.76%	1.80%	21.02%	2.32%	10.22%	0.66%	0.04%	1.62%	0.92%	2.06%	23.45%	2.35%

** Students who participate in more than one program will be counted in each program respectively.*

Figure 14. Spark Table Indicator with Correct Percentage Displayed

Dashboard Design and Development

After the development of the indicators, we visualized how we wanted to place the indicator components on the dashboard by creating a flowchart (Figure 15). The flowchart detailed how each area would be laid out. This helped organize the development process. As with the table breakout, we decided each data area would have a dashboard.

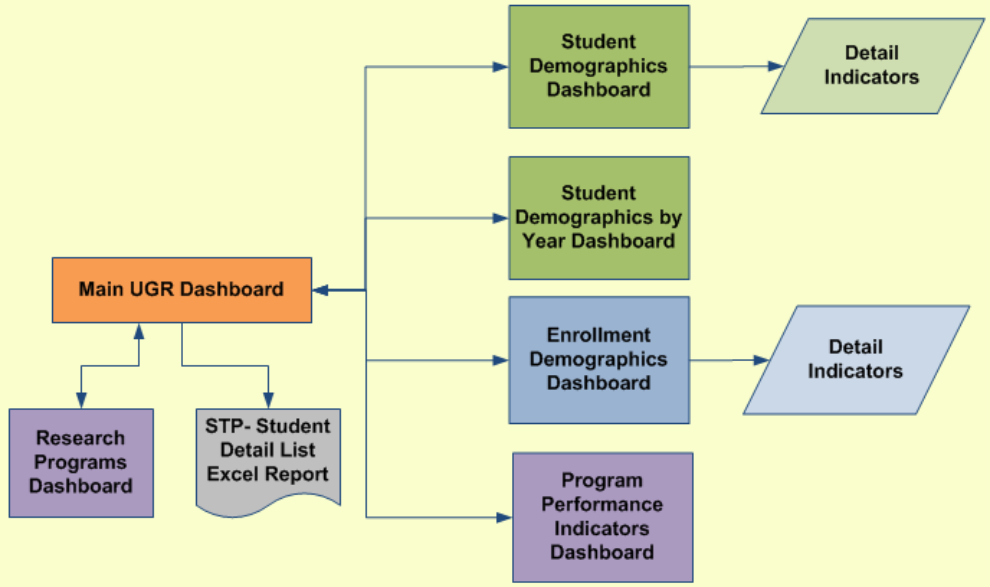


Figure 15. OUR Dashboard Flowchart

Despite some challenges, we were able to develop a proof of concept for OUR utilizing the SAS BI Dashboard. We hope they will see these informative dashboards as a means of meeting their goals and needs. Figure 16 shows some of the dashboards created for the proof of concept we delivered will deliver to OUR.

Figure 16. Examples of Dashboards Used in the Proof of Concept Delivered to OUR

CONCLUSION

Although there were some challenges in our experiences with dashboard development, the two projects addressed in this paper have shown to be a tremendous learning opportunity in seeing the possibilities for Education in utilizing the SAS BI Dashboard tool. Some of the hurdles we faced in the Retention Report Dashboard development included having to resolve the code placement issue with using a stored process indicator data type to make the aggregated data available for indicator creation and having to create additional variables in the data set to properly format the data when displayed in the indicators.

The OUR proof of concept project was one of the first projects where we were able to develop a complete dashboard using tables as a data source. The bumps we faced in the project mainly pertained to the nature of our Student Data not being structured in a way that worked easily in the dashboard application. This lead to the creation of customized fully aggregated tables being built prior to the creation of indicator data objects. Furthermore, workarounds for formatting issues were solved through creating a dummy character variable from which a label was created for the numeric variable used. After the issues were resolved, we were able to successfully create a functional dashboard as a proof of concept to OUR.

Through our experiences in dashboard development, we would recommend building a dashboard backwards. After visualizing what type of indicators are necessary, it would be easier to know if changes are needed with the dataset prior to building the indicator data and indicators. In this way it would not be necessary to continually go back and forth between building indicators and modifying indicator data to fit the needs of the indicator as we discovered the challenges we had in the projects.

RECOMMENDED READING

- *SAS® BI Dashboard 4.3 User's Guide*
- *Base SAS® 9.2 Procedures Guide*
- *SAS® 9.2 Language Reference*

CONTACT INFORMATION

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

Name: Evangeline (Angel) Collado
Work Phone: (407) 823-4968
E-mail: Evangeline.Collado@ucf.edu

Name: Michelle Parente
Work Phone: (407) 823-4764
E-mail: Michelle.Parente@ucf.edu

Organization: University of Central Florida, Enterprise Decision Support
Address: 12424 Research Parkway, Suite 215
City, State ZIP: Orlando, FL 32826-3269
Fax: (407) 823-4769
Web: www.ikm.ucf.edu