Paper SAS2021-2018

There Is No Reason to Hide! Discovering the Benefits of Using Hidden Columns in SAS® Visual Analytics

Michael Drutar, SAS Institute Inc., Cary, NC

ABSTRACT

SAS® Visual Analytics 8.2 includes many features that report designers can leverage to create dynamic, interactive reports. One of the most powerful new features is the hidden data role that is available for many reporting objects. While initially the concept of hidden columns or data might be a bit foreign, just a little instruction can empower report designers to leverage it and create their most dynamic SAS Visual Analytics reports. This paper demonstrates the various methods in which one can use hidden data in report design, and the benefits each of these methods provides to report consumers. Examples include robust mapping between multiple data sources, creating improved color-mapped display rules, and, most importantly, dynamic linking to external web resources based on the values in one or more hidden columns in a list table. Once armed with the knowledge of how to use the hidden data role in SAS Visual Analytics, report designers will be ready to create their most interactive, dynamic, and useful reports yet.

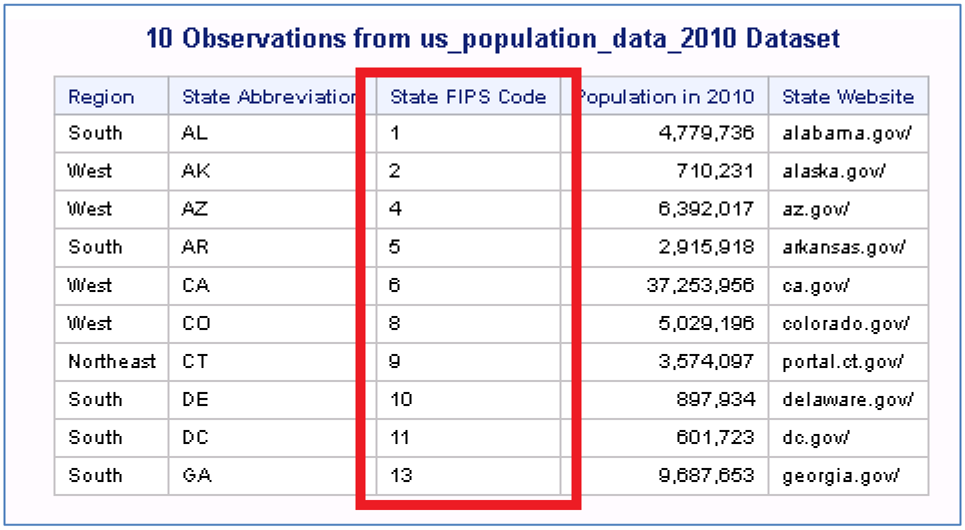
INTRODUCTION

To demonstrate the advantages that hidden columns provide, in this paper I will develop a report using SAS Visual Analytics 8.2. I will design it in a way that highlights the various needs for hidden columns. The new hidden data role will then be used to satisfy each of these needs.

ROBUST MAPPING BETWEEN DATA SOURCES:

When developing SAS Visual Analytics reports, it is quite often the case that the source data resides in multiple data sets. SAS Visual Analytics 6.3 introduced the ability to map two different data sources together and visualize them in one report. However, until SAS Visual Analytics 8.2, when two different data sources are mapped, their key data items have to be used in the objects. This can be problematic, since the data items that are needed to link the data sources might be internal code that should not be surfaced to the report. A common example of this is creating U.S. State geographic maps and needing to use the State FIPS Codes to link multiple data sources together.

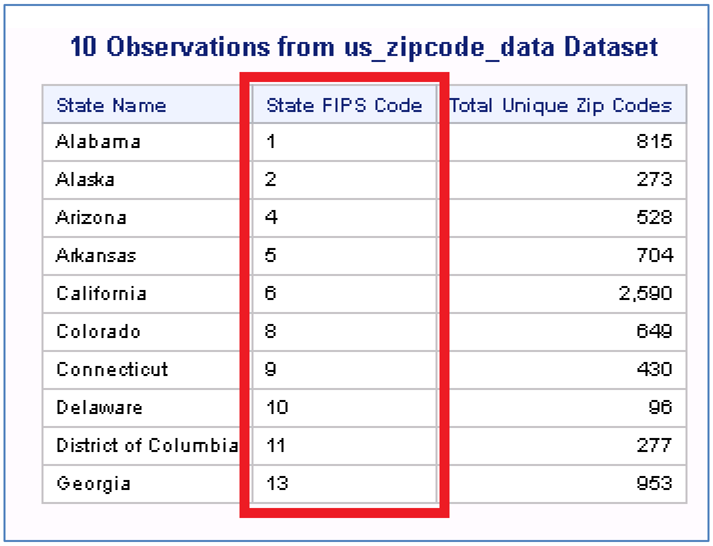
Consider the two data sets shown in Figure 1 and Figure 2 (10 observations are printed for each):



10 Observations from us_population_data_2010 Dataset

Region	State Abbreviation	State FIPS Code	Population in 2010	State Website
South	AL	1	4,779,736	alabama.gov/
West	AK	2	710,231	alaska.gov/
West	AZ	4	6,392,017	az.gov/
South	AR	5	2,915,918	arkansas.gov/
West	CA	6	37,253,956	ca.gov/
West	CO	8	5,029,196	colorado.gov/
Northeast	CT	9	3,574,097	portal.ct.gov/
South	DE	10	897,934	delaware.gov/
South	DC	11	601,723	dc.gov/
South	GA	13	9,687,653	georgia.gov/

Figure 1. 10 Observations from the "us_population_data_2010" Data Set

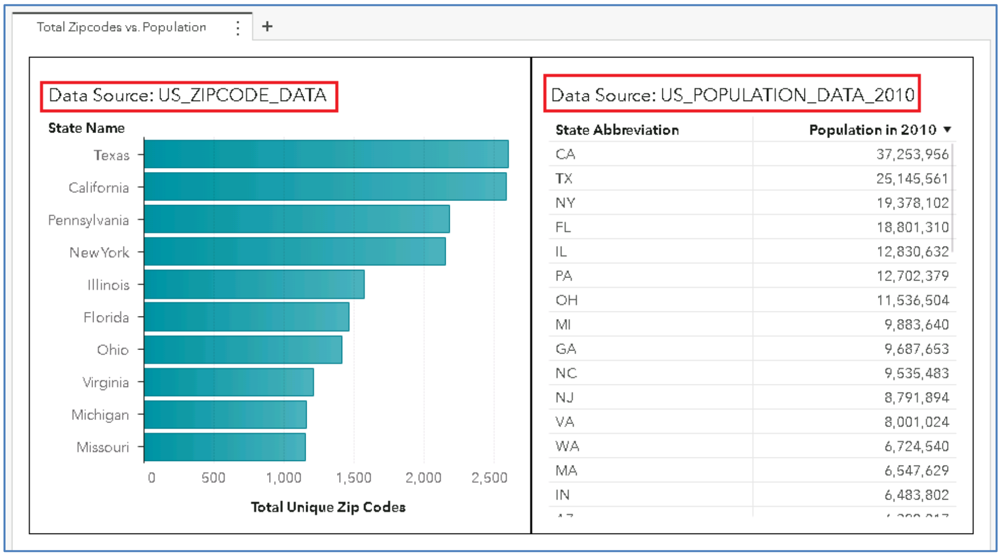


10 Observations from us_zipcode_data Dataset

State Name	State FIPS Code	Total Unique Zip Codes
Alabama	1	815
Alaska	2	273
Arizona	4	528
Arkansas	5	704
California	6	2,590
Colorado	8	649
Connecticut	9	430
Delaware	10	96
District of Columbia	11	277
Georgia	13	953

Figure 2. 10 Observations from the "us_zipcode_data" Data Set

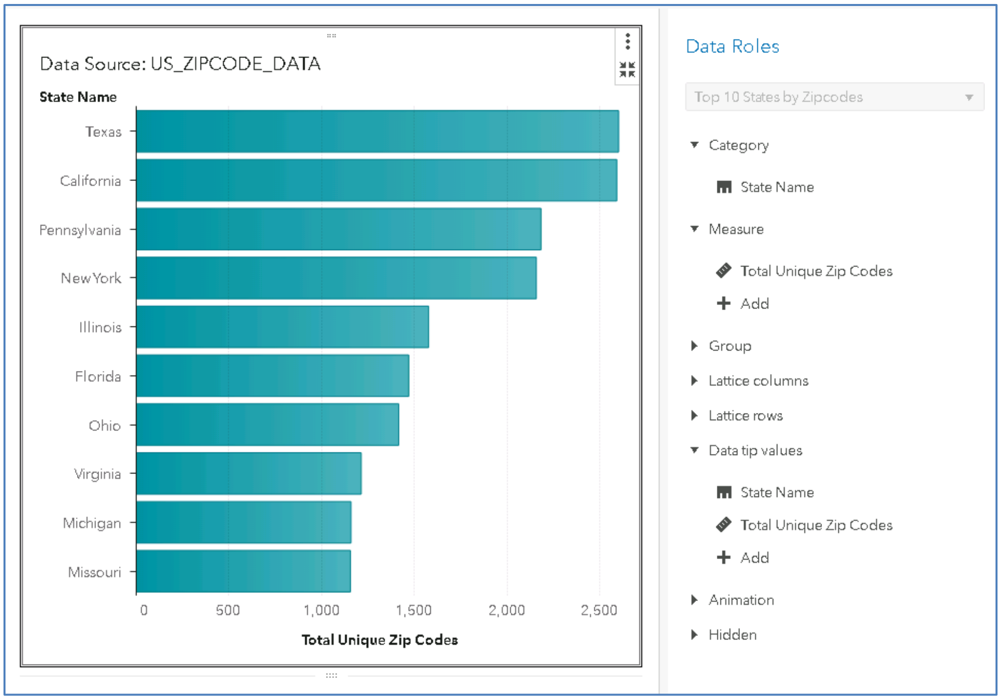
Notice how the common variable in the two data sets is the FIPS code. This is the only key column that can be used to link the two data sets together. With these two data sets in mind, consider the SAS Visual Analytics report in Figure 3 that has two objects, each using a different data source:



State Abbreviation	Population in 2010 ▼
CA	37,253,956
TX	25,145,561
NY	19,378,102
FL	18,801,310
IL	12,830,632
PA	12,702,379
OH	11,536,504
MI	9,883,640
GA	9,687,653
NC	9,535,483
NJ	8,791,894
VA	8,001,024
WA	6,724,540
MA	6,547,629
IN	6,483,802

Figure 3. SAS Visual Analytics Report Built from Two Different Data Sources

The data roles for the bar chart are shown in Figure 4.



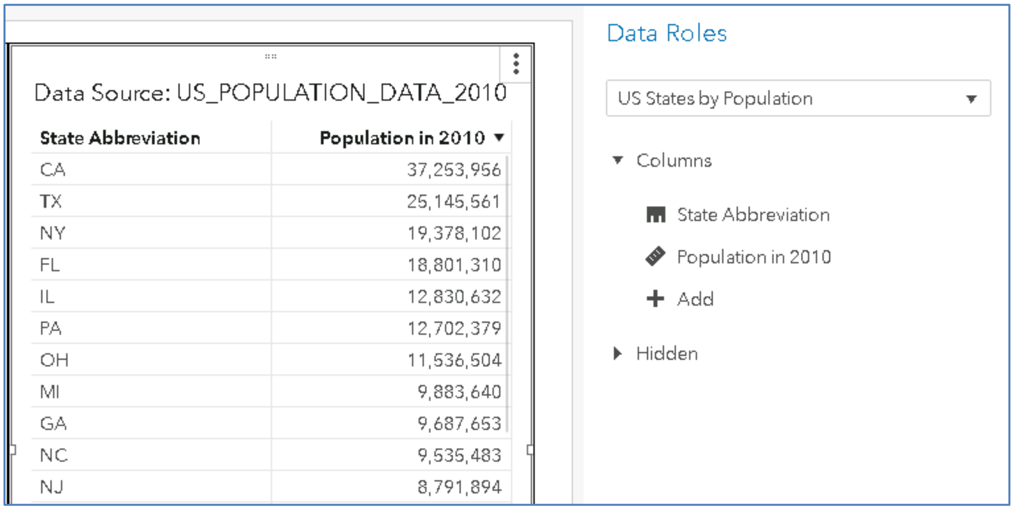
Data Roles

Top 10 States by Zipcodes

- Category
  - State Name
- Measure
  - Total Unique Zip Codes
  - Add
- Group
- Lattice columns
- Lattice rows
- Data tip values
  - State Name
  - Total Unique Zip Codes
  - Add
- Animation
- Hidden

Figure 4. Bar Chart Data Roles

The data roles for the list table are shown in Figure 5.



Data Source: US_POPULATION_DATA_2010

State Abbreviation	Population in 2010 ▼
CA	37,253,956
TX	25,145,561
NY	19,378,102
FL	18,801,310
IL	12,830,632
PA	12,702,379
OH	11,536,504
MI	9,883,640
GA	9,687,653
NC	9,535,483
NJ	8,791,894

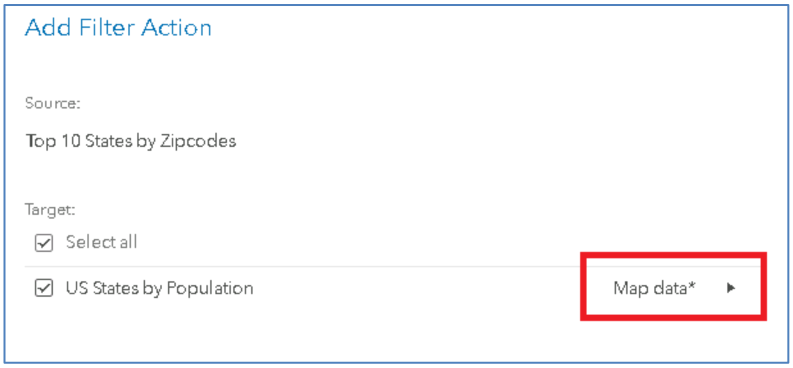
Data Roles

US States by Population

- Columns
  - State Abbreviation
  - Population in 2010
  - Add
- Hidden

Figure 5. List Table Data Roles

The report designer would like to have an action between the two objects, so that when a state is selected in the bar chart the list table highlights that state's population data. Hence, a linked selection action is created from the map to the list table, and the **Map data** option is selected, as shown in Figure 6.



Add Filter Action

Source:

Top 10 States by Zipcodes

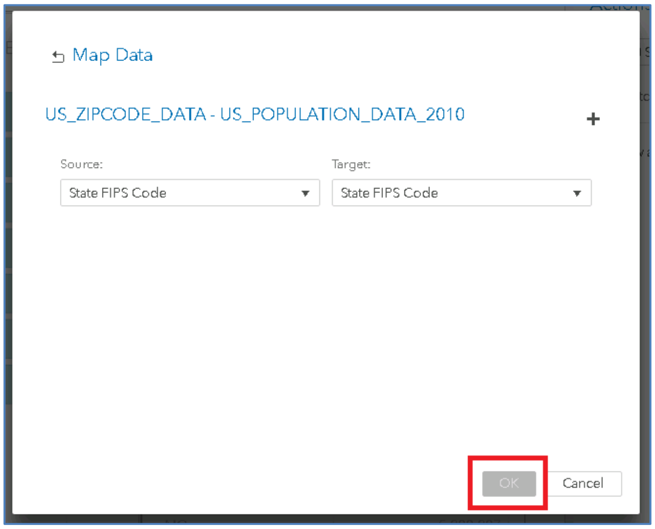
Target:

☑ Select all

☑ US States by Population

Figure 6. Creating an Action for Report Objects That Use Different Data Sources

Since the only common data item between the two data sets is STATE_FIPS_CODE, the report designer needs to use it in the mapping. However, even though STATE_FIPS_CODE is available in the drop-down list for both objects, when they are selected the **OK** button is grayed out (see Figure 7).



Map Data

US_ZIPCODE_DATA - US_POPULATION_DATA_2010

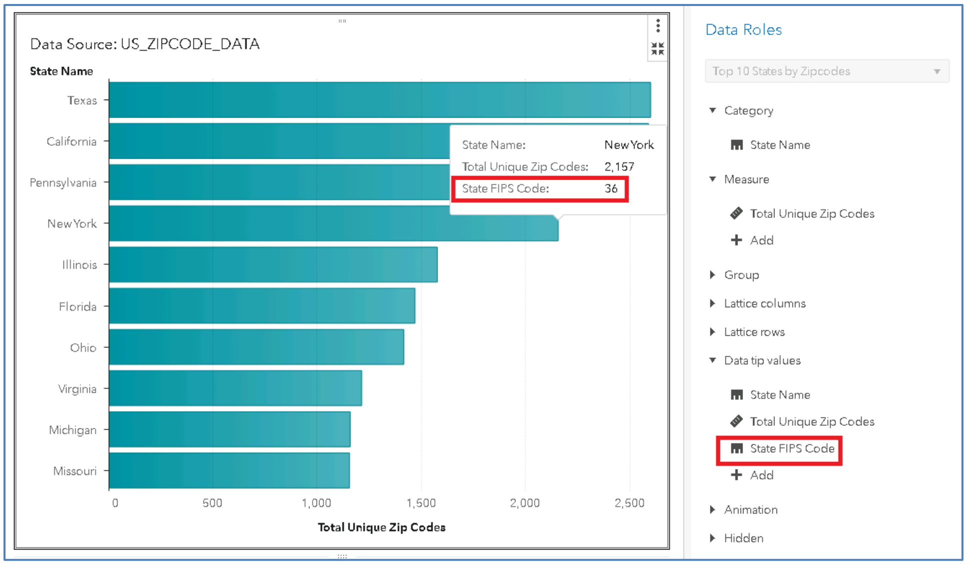
Source: State FIPS Code

Target: State FIPS Code

Figure 7. New Action Wizard Is Unable to Map Data Sources Using Common Variable

The **OK** button is grayed out because the STATE_FIPS_CODE is not used in any data roles for any of the objects. In previous versions of SAS Visual Analytics, the only way these two data sources could be mapped was if the report designer re-architects the report so that the common data item (STATE_FIPS_CODE, in this case) is included and therefore visible in the objects. This is an inelegant solution to mapping the data sets.

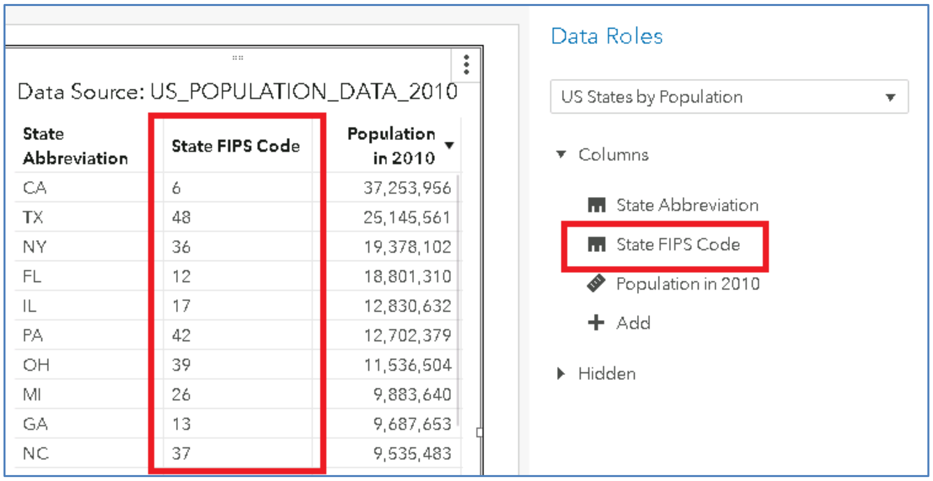
For a moment, imagine that we are working in a previous version of SAS Visual Analytics and hidden columns are not available and the report designer has to include the STATE_FIPS_CODE data item in both the bar chart and list table. For the bar chart, the STATE_FIPS_CODE can be “hidden” in the data tip values role. While this “hides” the STATE_FIPS_CODE data item, when the user moves the pointer over a bar, the STATE_FIPS_CODE is still visible, as shown in Figure 8, which could be confusing to the report consumer.



State Name: New York
Total Unique Zip Codes: 2,157
State FIPS Code: 36

Figure 8. Attempting to "Hide" STATE_FIPS_CODE in Data Tip Values Role Prior to the 8.2 Release

For the list table, there are no roles in which the STATE_FIPS_CODE can be hidden. So, the only option that remains is to include the STATE_FIPS_CODE in the columns role, which will display the STATE_FIPS_CODE to the user. When rendered, this produces a confusing list table, as shown in Figure 9.

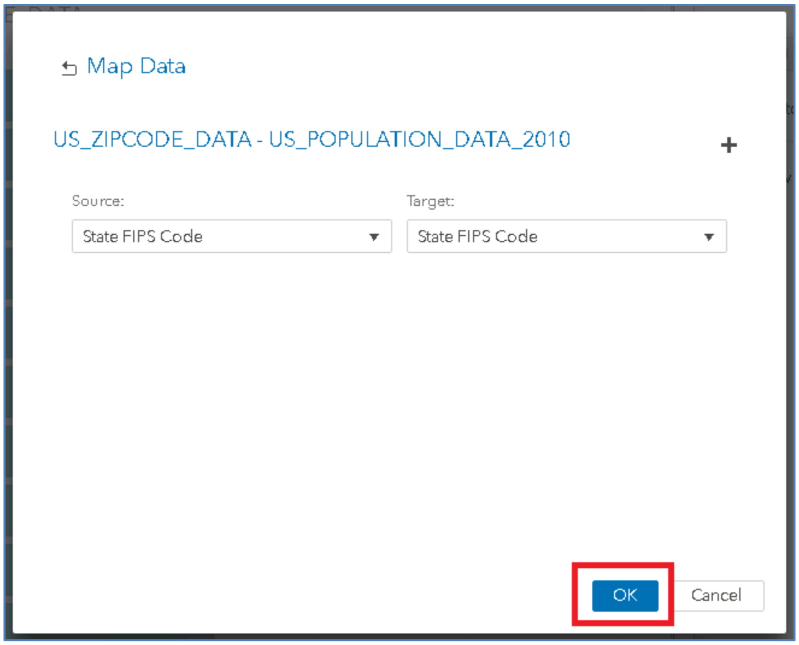


Data Source: US_POPULATION_DATA_2010

State Abbreviation	State FIPS Code	Population in 2010
CA	6	37,253,956
TX	48	25,145,561
NY	36	19,378,102
FL	12	18,801,310
IL	17	12,830,632
PA	42	12,702,379
OH	39	11,536,504
MI	26	9,883,640
GA	13	9,687,653
NC	37	9,535,483

Figure 9. Adding STATE_FIPS_CODE to List Table Object's Roles to Enable Data Source Mapping

As disappointing as the solution shown in Figure 9 is, when the Map Data wizard is opened, the mapping connection is now allowed (see Figure 10).



Map Data

US_ZIPCODE_DATA - US_POPULATION_DATA_2010

Source: State FIPS Code

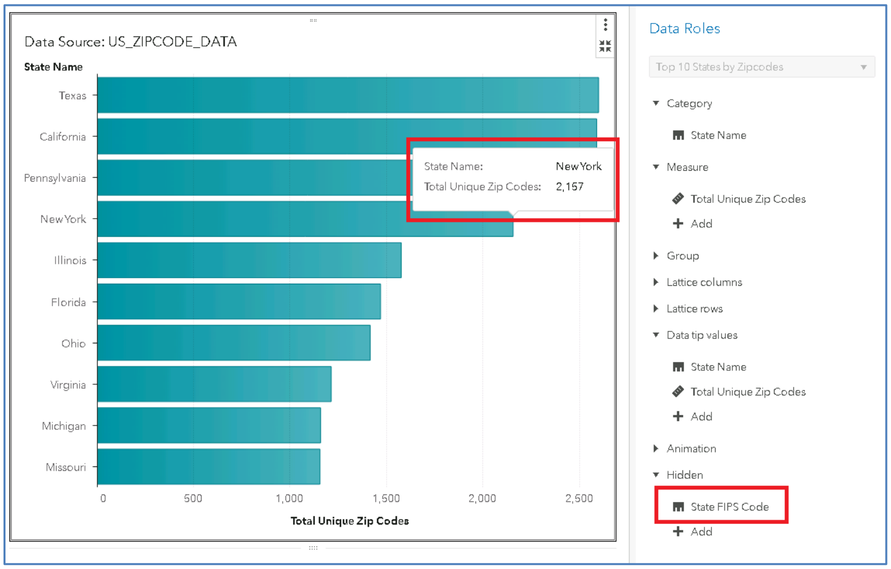
Target: State FIPS Code

Figure 10. Completing Data Mapping between Objects Now That STATE_FIPS_CODE Is Used in Both Objects

Even though data mapping has been successfully added, the cost of confusion to the user by displaying the STATE_FIPS_CODE is significant. It would be great if the report designer could still use the STATE_FIPS_CODE for mapping, while also *hiding* it from the report consumer.

HIDDEN DATA ROLES SOLUTION

Fortunately, the hidden data role in SAS Visual Analytics 8.2 does exactly that. It allows the report objects to include data items for report functionality without displaying them to the user interface (UI). Using this new hidden data role in our working example, we move our key data item STATE_FIPS_CODE to the hidden data role on both objects, as shown in Figure 11 and Figure 12.

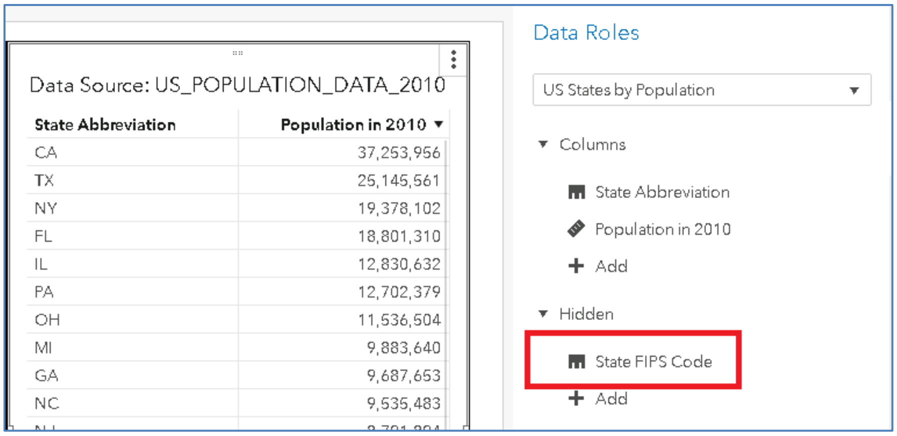


Data Roles

Top 10 States by Zipcodes

- Category
  - State Name
- Measure
  - Total Unique Zip Codes
  - Add
- Group
- Lattice columns
- Lattice rows
- Data tip values
  - State Name
  - Total Unique Zip Codes
  - Add
- Animation
- Hidden
  - State FIPS Code
  - Add

Figure 11. Applying STATE_FIPS_CODE Variable to the Bar Chart's Hidden Data Role



Data Source: US_POPULATION_DATA_2010

State Abbreviation	Population in 2010 ▼
CA	37,253,956
TX	25,145,561
NY	19,378,102
FL	18,801,310
IL	12,830,632
PA	12,702,379
OH	11,536,504
MI	9,883,640
GA	9,687,653
NC	9,535,483

Data Roles

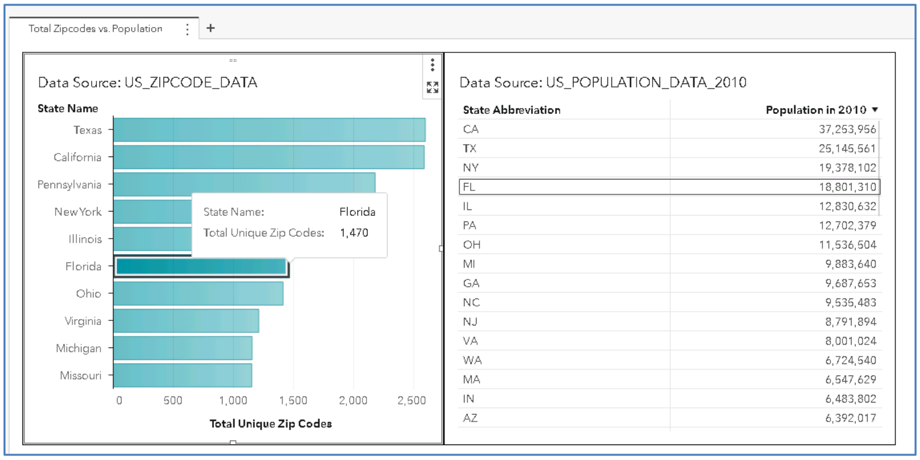
US States by Population ▼

▼ Columns
- State Abbreviation
- Population in 2010
- ＋ Add

▼ Hidden
- State FIPS Code
- ＋ Add

Figure 12. Applying STATE_FIPS_CODE Variable to the List Table's Hidden Data Role

By leveraging the hidden data role, we can have the data mapping for the brushing interaction *without* having to display the STATE_FIPS_CODE in either object. This robust data mapping has enabled the desired action while also suppressing it from the UI, allowing the two objects to interact using a hidden common data item (see Figure 13).



Data Source: US_POPULATION_DATA_2010

State Abbreviation	Population in 2010 ▼
CA	37,253,956
TX	25,145,561
NY	19,378,102
FL	18,801,310
IL	12,830,632
PA	12,702,379
OH	11,536,504
MI	9,883,640
GA	9,687,653
NC	9,535,483
NJ	8,791,894
VA	8,001,024
WA	6,724,540
MA	6,547,629
IN	6,483,802
AZ	6,392,017

Figure 13. Successfully Creating an Interaction between Two Data Sources Using a STATE_FIPS_CODE in the Hidden Columns

CREATING IMPROVED COLOR-MAPPED DISPLAY RULES

Another advantage the hidden data role provides is the ability to use the data role for creating more robust display rules. Continuing with the same report from the data mapping example in the previous section, the data set US_POPULATION_DATA_2010 includes the variable "Region." Perhaps the report designer would like to color-code the states based on the region they are in.

Using the Display Rules pane for the list table, we select **New Rule**. However, the variable "Region" is not available to create the display rule definition because it is not assigned to the report object (see Figure 14).

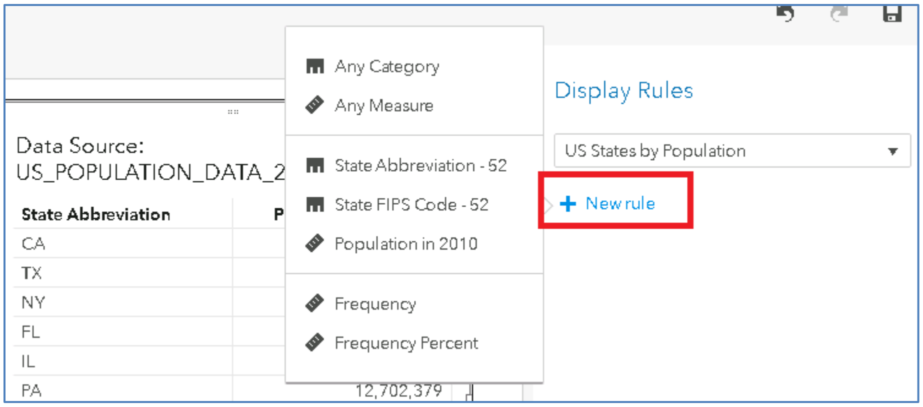
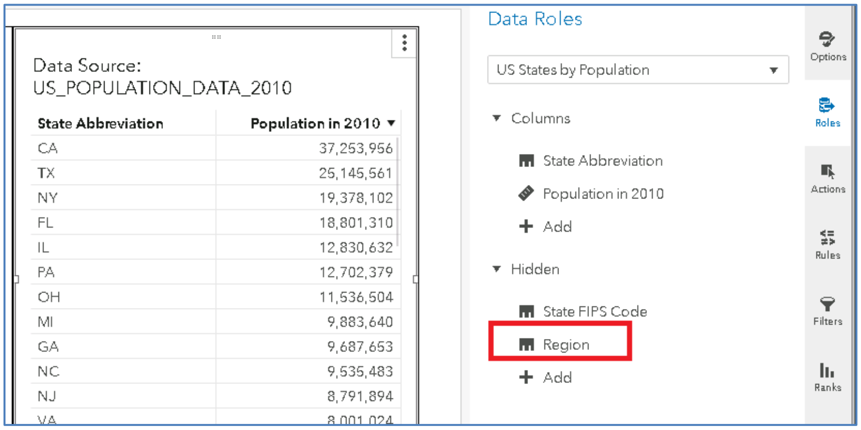


Figure 14. Unable to Create a Display Rule for the List Table Using the Variable Region

As with the STATE_FIPS_CODE, the report designer does not want to add the “Region” variable to the object’s display but still wants to use the column to color-code the table’s observations by region. To solve this problem, the report designer can add the “Region” data item to the Hidden data role, as shown in Figure 15.


Data Source:
US_POPULATION_DATA_2010

State Abbreviation	Population in 2010
CA	37,253,956
TX	25,145,561
NY	19,378,102
FL	18,801,310
IL	12,830,632
PA	12,702,379
OH	11,536,504
MI	9,883,640
GA	9,687,653
NC	9,535,483
NJ	8,791,894

Figure 15. Adding REGION as a Second Hidden Column to the List Table

Once Region has been added to the Hidden data role, the Region variable appears as an option in the Rules wizard and can be selected (see Figure 16).

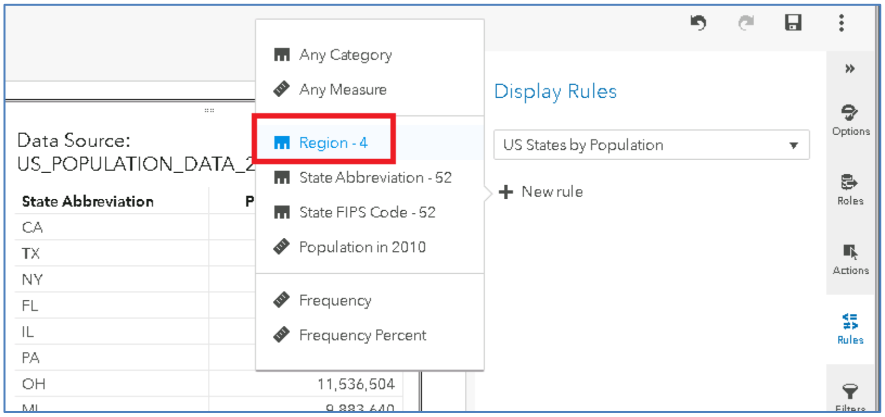
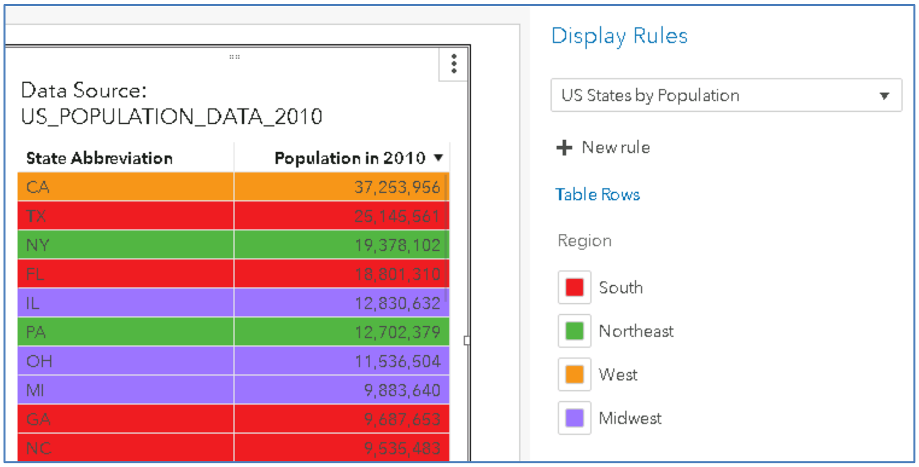


Figure 16. Creating a Display Rule Based on the Hidden Column REGION

As a result, a color-mapped display rule based on the Region data item (which is now assigned to the Hidden data role) can be created (see Figure 17).



Data Source:
US_POPULATION_DATA_2010

State Abbreviation	Population in 2010 ▼
CA	37,253,956
TX	25,145,561
NY	19,378,102
FL	18,801,310
IL	12,830,632
PA	12,702,379
OH	11,536,504
MI	9,883,640
GA	9,687,653
NC	9,535,483

Display Rules

US States by Population ▼

+ New rule

Table Rows

Region

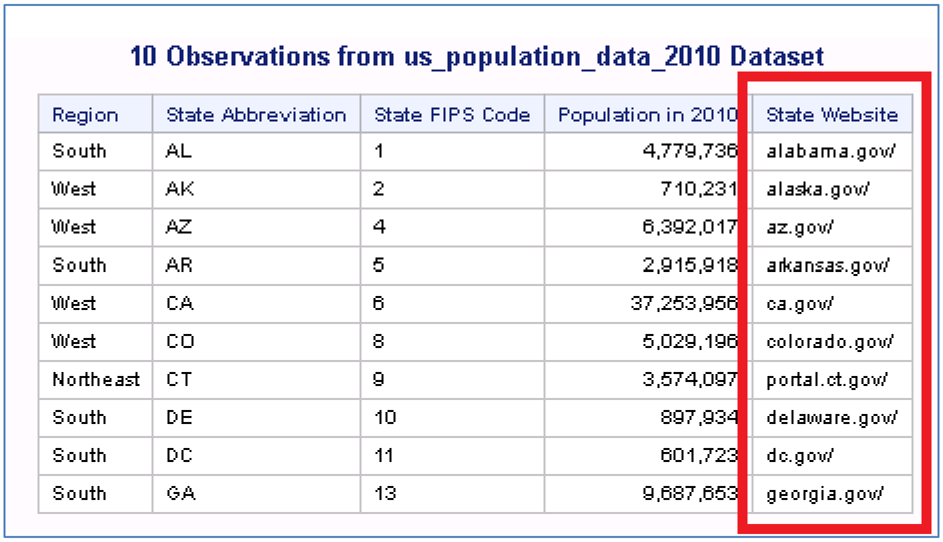
- 🟥 South
- 🟩 Northeast
- 🟧 West
- 🟪 Midwest

Figure 17. Viewing the Display Rule Based on the Hidden Data Role REGION

DYNAMIC LINKING

Another useful action available within SAS Visual Analytics is the ability link a report object to an external website. While the rest of the actions within SAS Visual Analytics apply to objects within the report (creating filters, creating page links, and so on), external links allow report designers to integrate their reports into web applications outside of SAS Visual Analytics.

Using this feature, the report designer can add parameters to the base link that is supplied in the action. Continuing with our example from above, a new action can be applied to the report's color-coded list table. As shown at the beginning of this paper, one of the columns within the US_POPULATION_DATA_2010 data set is "State Website." The report designer would like to create a URL link action using this column so that when the report consumer clicks on an individual row in the list table (such as "CA"), he or she is taken to California's state website: `http://ca.gov`. See Figure 18.

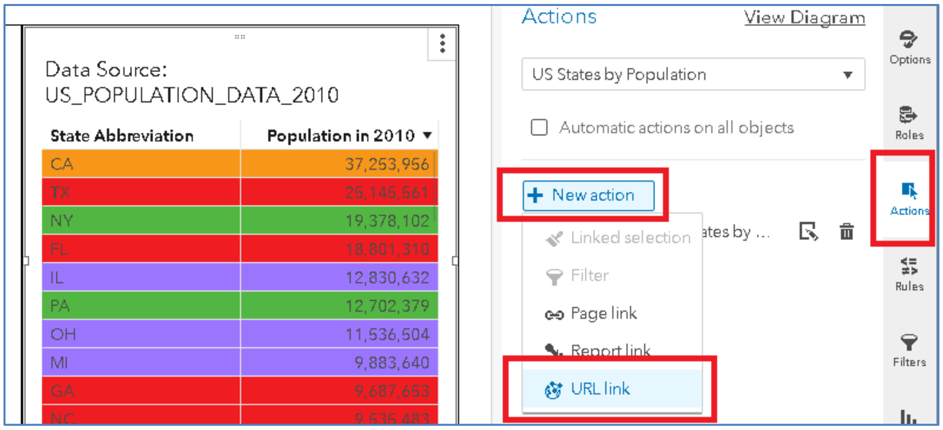


10 Observations from us_population_data_2010 Dataset

Region	State Abbreviation	State FIPS Code	Population in 2010	State Website
South	AL	1	4,779,736	alabama.gov/
West	AK	2	710,231	alaska.gov/
West	AZ	4	6,392,017	az.gov/
South	AR	5	2,915,918	arkansas.gov/
West	CA	6	37,253,956	ca.gov/
West	CO	8	5,029,196	colorado.gov/
Northeast	CT	9	3,574,097	portal.ct.gov/
South	DE	10	897,934	delaware.gov/
South	DC	11	601,723	dc.gov/
South	GA	13	9,687,653	georgia.gov/

Figure 18. 10 Observations from the "us_population_data_2010" Data Set

With the list table selected, the report designer selects "New action" from the Actions pane, then selects **URL link**. See Figure 19.



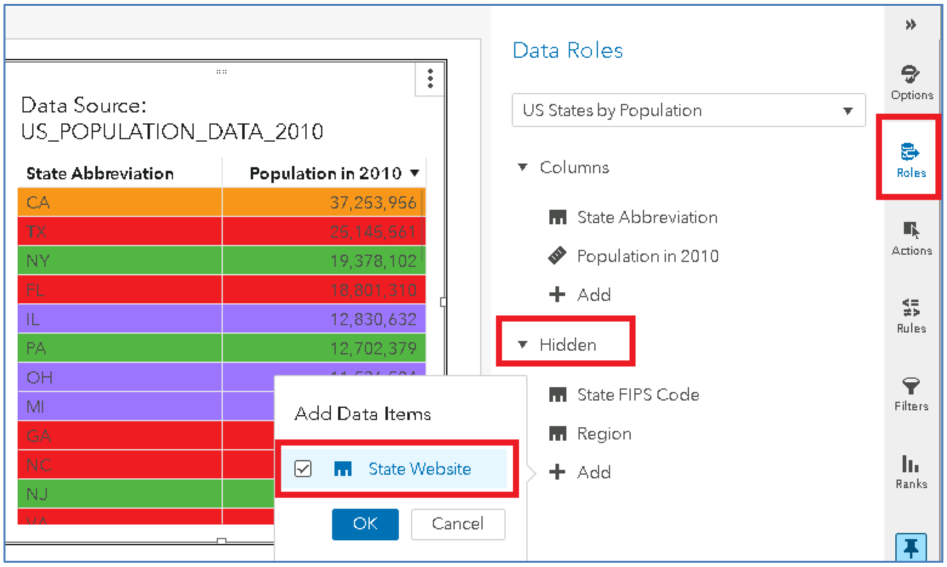
Data Source:
US_POPULATION_DATA_2010

State Abbreviation	Population in 2010 ▼
CA	37,253,956
TX	25,145,561
NY	19,378,102
FL	18,801,310
IL	12,830,632
PA	12,702,379
OH	11,536,504
MI	9,883,640
GA	9,687,653
NC	9,535,483

Figure 19. Creating a URL Link Action on the List Table

The report designer names the new link "Selected State's Government Website." For the URL link, the report designer inputs the suggested default value of `http://`. In the Parameters section, the report designer has the power to create a truly dynamic link from the SAS Visual Analytics list table. A parameter in a URL link is sourced from a column in the object's data source. When the report consumer selects the object, SAS Visual Analytics appends the value of the columns for each parameter to the end of the base URL.

However, in our working example, when the report designer attempts to add the desired column "State Website" as the parameter's source, it is not available. Once again, this is because the column "State Website" is currently not being used by the list table. To achieve the desired effect of linking to the state government's website, the report designer can assign "State Website" to the hidden data role, as shown in Figure 20.

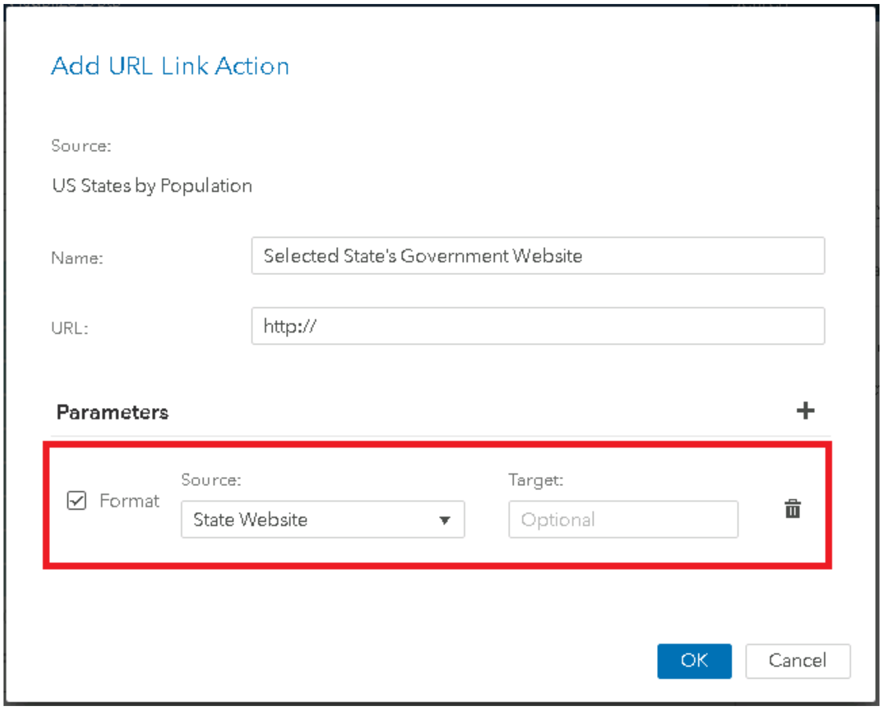


Data Source:
US_POPULATION_DATA_2010

State Abbreviation	Population in 2010 ▼
CA	37,253,956
TX	25,145,561
NY	19,378,102
FL	18,801,310
IL	12,830,632
PA	12,702,379
OH	
MI	
GA	
NC	
NJ	

Figure 20. Adding STATE_WEBSITE as a Third Hidden Data Role to the List Table

Adding the column "State Website" to the hidden data role allows it to be used as a parameter source, but with the advantage of not displaying the URL as a column in the list table. See Figure 21.



Add URL Link Action

Source:
US States by Population

Name: Selected State's Government Website

URL: http://

Parameters

☑ Format Source: State Website Target: Optional

Figure 21. Creating a Parameter Using the Hidden Column STATE_WEBSITE

Now, when the report user selects a row in the list table (“CA,” for example), SAS Visual Analytics creates the base URL and dynamically appends the parameter’s value (“`ca.gov/`”), generates the complete URL: `http://ca.gov`.

This example creates a very simplistic dynamic link. However, since the hidden data role can have multiple assignments, complex URL links can be generated from multiple columns, all without displaying any of the parameter source columns in the list table. This opens a whole new world of possibilities for report designers when it comes to integrating their SAS Visual Analytics reports to external web applications.

CONCLUSION

In the world of software development, back-end (or hidden) processes are what make applications seem to work like magic. Sometimes the magic is a search box auto-completing a search after a few characters are typed. Other times it is SAS Visual Analytics delivering a user to a website based on the value they click in a list table. Either way, the “magic” is hidden from the report consumer. Hidden columns ushers the SAS Visual Analytics report designer into a brand-new world of creating reports that are impressive, useful, and dynamic.

CONTACT INFORMATION

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

Michael Drutar
100 SAS Campus Drive
Cary, NC 27513
SAS Institute Inc.
Michael.Drutar@sas.com
http://www.sas.com