Paper SAS567-2017

Wrangling Your Data into Shape for In-Memory Analytics

Gary Mehler, SAS institute Inc., Cary, NC

ABSTRACT

High-quality analytics works best with the best-quality data. Preparing your data ranges from activities like text manipulation and filtering to creating calculated items and blending data from multiple tables. This paper covers the range of activities you can easily perform to get your data ready. High-performance analytics works best with in-memory data. Getting your data into an in-memory server, as well as keeping it fresh and secure, are considerations for in-memory data management. This paper covers how to make small or large data available and how to manage it for analytics. You can choose to perform these activities in a graphical user interface or via batch scripts. This paper describes both ways to perform these activities. You'll be well-prepared to get your data wrangled into shape for analytics!

INTRODUCTION

SAS® Visual Analytics leverages SAS® high-performance analytic technologies and empowers organizations to explore huge volumes of data very quickly to identify patterns, trends, and opportunities for further analysis. The highly visual, drag-and-drop data interface of SAS Visual Analytics, combined with the speed of the SAS® Cloud Analytic Services (CAS) server in SAS® Viya™, accelerate analytic computations and enable organizations to derive value from massive amounts of data.

At the heart of everything is your data. It needs to be available, current, and report-ready. It might need processing to become analytics-ready so that analytics users can focus on their tasks. Just as SAS Visual Analytics has a highly visual interface, we will be talking about visual ways to import data, perform data manipulations, and manage it. We will walk through each step to discuss how to perform both basic as well as a few more advanced tasks.

If you are just getting started, you will want to add some data to your SAS Visual Analytics system. Self-service import offers an easy way to bring data into the CAS environment. Data can come from local files, server-based data, or web-based services. If your new data is report-ready, that's all there is to it. However, if some data manipulation is needed, then using SAS® Visual Data Builder is advised. Data manipulation in this context means either changing the shape of the data, such as filtering or adding rows or columns, or changing values, such as from a character to numeric type.

In SAS Visual Data Builder, several examples of basic to advanced data manipulations will be covered, and we will explain how these methods can support different capabilities in your Visual Analytics reports. In addition, more advanced topics such as securing your data will be covered, so you can be sure the right users have access to the right data for their analytics needs.

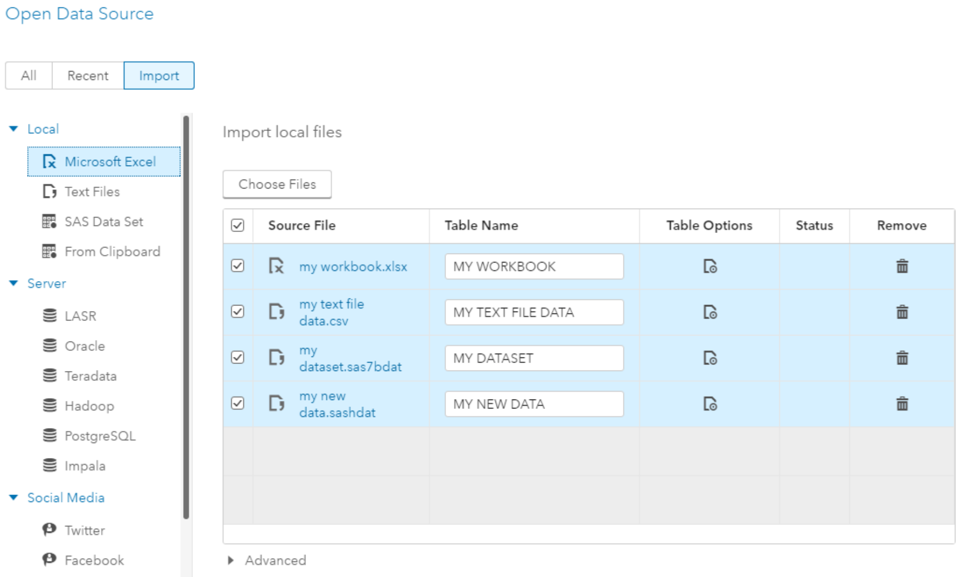
GETTING DATA INTO YOUR SYSTEM

If you are just getting started on a brand new system, you will want some data to work with. For many, this will mean using local data from their desktop computer. When you import a local data file from your desktop, such as a spreadsheet, a delimited text file, or a SAS data set, the file is loaded as data in the CAS Server. Once there, it is ready for data manipulation or analytics visualization.

If you are doing this by yourself, then self-service data import is for you. It is a user interface that is available in several applications that you can use to import your data into the CAS Server. Self-service means that you don't need a system administrator to do this for you, but it does mean you need to know how to access your data. But since you probably do, let's get started.

The self-service panel lets you specify the items you want to import in a few ways that we'll go through. When we are opening a data source, import is an option if your data is new. Import works with the types of data shown on the left side of the figure below: local files, server data, and web-based social media sources.

As you can see below, a number of local files have been added to the self-service panel. These can be added either by browsing your desktop computer and its network drives, or by dragging and dropping from another windows on your computer. Once there, you can use the Table Options area to specify how each item can be processed correctly.



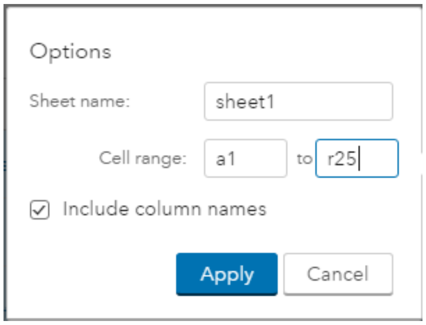
☑	Source File	Table Name	Table Options	Status	Remove
☑	my workbook.xlsx	MY WORKBOOK			🗑
☑	my text file data.csv	MY TEXT FILE DATA			🗑
☑	my dataset.sas7bdat	MY DATASET			🗑
☑	my new data.sashdat	MY NEW DATA			🗑

Figure 1. Importing Data

When loading a delimited text file or SAS data set, the process involves loading that data in its entirety for use. If the data is a spreadsheet, you can effect a filtering process by specifying either a subset of spreadsheets in a Microsoft Excel workbook or data ranges within a sheet. Getting data into the system correctly is important, so we will review the sets of options available for each type of file:

Excel workbooks, as shown in the figure below, might require you to specify unique options for your data, such as the following:

- A specific worksheet to load. If this is not specified, the first worksheet will be loaded. This is useful when a workbook contains many sheets, but you know which one is useful for analytics
- Cell range. If the desired sheet has extraneous content, such as heading rows for readability that aren't actually useful data, specifying these values will be needed. Another reason to consider this is if your worksheet has footnote-like documentation to the side or bottom of useable data. In this case, it pays to visually review your data before importing it.



Sheet name: sheet1

Cell range: a1 to r25

☑ Include column names

Figure 2.Import Options for an Excel Workbook

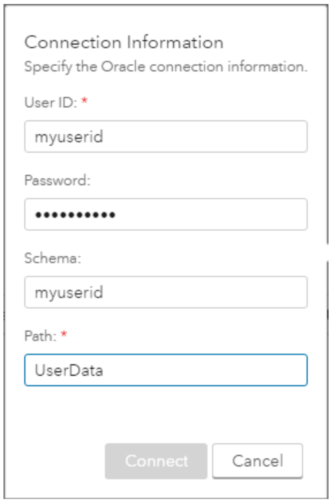
Text files have a range of options that are important to get right so that imported data is ready for analysis, including:

- Delimiter between fields that is often a comma or a tab, although any character could be used. If incorrect, your loaded data will not look like the rectangular shape you're expecting.
- Whether to use variable-length character strings, which are helpful for some types of character data such as data being loaded for text analytics. When these strings have widely varying lengths, specifying variable-length storage can save a lot of storage space when the data has been loaded in CAS.
- Encoding. If data is being brought forward from a legacy system, it is important to note that the CAS server uses UTF-8 encoding for its data. If your data has a different encoding, this might need to be known. This is also a helpful place to point out that UTF-8 storage can require more space than a more specific representation like Latin-1, which is common in the U.S. and Western European countries. But for our purposes here, we need to ensure that data is imported correctly.

SAS data sets and SASHDAT files can be read as well. You are probably familiar with data sets that usually have an extension of SAS7BDAT and have been in use by SAS for some time. SASHDAT files are a new type of file storage used in SAS Viya and allow new type of fields to be saved, such as variable-length character fields.

Both of these file types can support encryption to protect data at-rest. With sensitive data, you'll want to ensure that data is managed in a safe way throughout its lifecycle. So we later talk about where that data will be loaded and managed after being imported.

That covers the types of local files you can import, but other types of data can also be loaded into your Visual Analytics system, and we'll briefly cover those here. Below is an example panel into which basic parameters for an Oracle database system have been specified. Other types of systems have comparable panels where you can specify appropriate options for those types of systems.



Connection Information

Specify the Oracle connection information.

User ID: *

myuserid

Password:

••••••••••

Schema:

myuserid

Path: *

UserData

Figure 3. Connection Parameters for Oracle

After providing connection information, you can import a table to the CAS server from a database or from SAS® LASR Analytic Server. You can import data from the following databases: Teradata, Oracle, Hadoop, PostgreSQL, Impala. For each of these, you will need to know connection parameters such as the machine serving up the data, login credentials, and other values such as schema. Once specified, selected data from one of these systems can be brought into your system for analysis. It is worth pointing out that importing from a database requires an appropriate license, but that accessing SAS LASR data doesn't require any additional license.

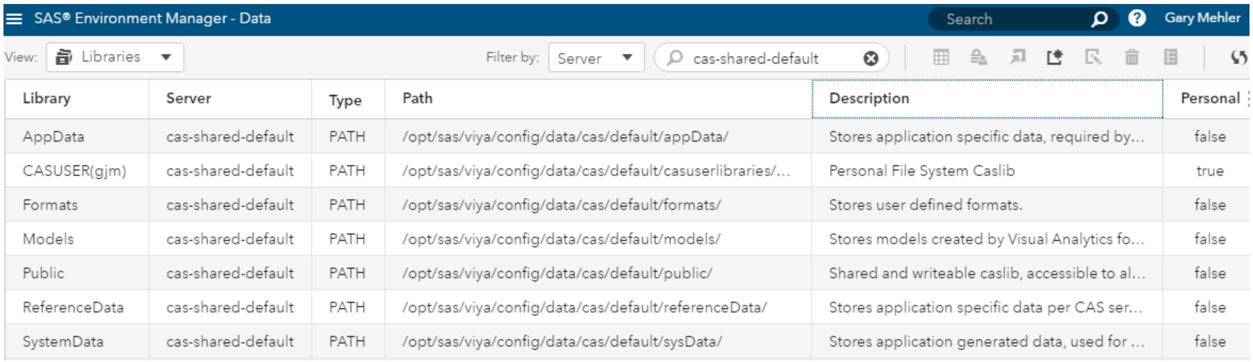
Since we are doing this as a self-service activity, you will need to know these basic parameters. If you don't, then a system administrator can create a permanent, shared definition in SAS® Environment Manager.

The last type of importable data we'll discuss is for web-based sources like social media systems. After authenticating with Facebook, Google Analytics, Twitter, or YouTube and providing search criteria, you can import data to the CAS server. Each source has a pre-defined schema of information that is returned and then loaded for analytic and reporting use.

But wait, there's one more. While not strictly a file that you can import, if you have data that you want to copy and paste from a source like a web page, you can do that via Clipboard import. To use Clipboard import, you will need to mark and copy (for example, using Control-C on a Windows PC) tabular information from a web page, which places that data into the clipboard buffer on your computer. Then, after selecting From Clipboard in the importer panel, just paste (using Control-V on a Windows PC) the clipboard contents and import. If you do use Clipboard to import from a web source, be aware of your rights to use data that is displayed on that page.

LIBRARIES AND STORAGE

As data is added to your system, let's discuss how it is stored and accessed when analytics are performed. SAS Visual Analytics has several libraries set up by default for data storage, and others can be created in the administrative interface SAS® Environment Manager. By default, initial actions for importing new data will default to a PUBLIC library, as described below. As you might expect, the PUBLIC library is accessible to all.



Library	Server	Type	Path	Description	Personal
AppData	cas-shared-default	PATH	/opt/sas/viya/config/data/cas/default/appData/	Stores application specific data, required by...	false
CASUSER(gjm)	cas-shared-default	PATH	/opt/sas/viya/config/data/cas/default/casuserlibraries/...	Personal File System Caslib	true
Formats	cas-shared-default	PATH	/opt/sas/viya/config/data/cas/default/formats/	Stores user defined formats.	false
Models	cas-shared-default	PATH	/opt/sas/viya/config/data/cas/default/models/	Stores models created by Visual Analytics fo...	false
Public	cas-shared-default	PATH	/opt/sas/viya/config/data/cas/default/public/	Shared and writeable caslib, accessible to al...	false
ReferenceData	cas-shared-default	PATH	/opt/sas/viya/config/data/cas/default/referenceData/	Stores application specific data per CAS ser...	false
SystemData	cas-shared-default	PATH	/opt/sas/viya/config/data/cas/default/sysData/	Stores application generated data, used for ...	false

Figure 4. Standard Libraries

The figure above shows a default configuration for SAS Visual Analytics 8.1. This panel of SAS Environment Manager lists the libraries that are initially created when Visual Analytics is deployed, and we discuss their expected usage below.

- PUBLIC library – this is the initial default location. It is an area that has fully Open permissions, meaning that any user can place data there, and that all data is accessible by all users. If the intent of loading data is to support general sharing, this is a good choice.
- CASUSER library – this is a personal and private location that is specific to each user running SAS Visual Analytics and CAS. This area is a good choice for data that is sensitive or otherwise needs to be accessible only by the person who has imported it. The physical location of this library is created when first needed, and file system permissions are set appropriately. Personal libraries are private to the extent that even a SAS Visual Analytics administrator doesn't have the ability to view their existence or contents in applications like SAS Environment Manager.
- FORMATS library – if the data uses SAS format libraries, they should be loaded into this area. Although not covered in detail here, information is available in online documentation for SAS Viya 3.2 Administration: Data under the heading User-Defined Formats.
- Other libraries have special purposes such as storage of data files for graphical map rendering and text analytics reference data, but will not be discussed in detail here.

IMPORTING AND PRE-DEFINED LIBRARIES

Why are we so concerned about these libraries? It is because the self-service import process stores a copy of imported data in the corresponding library to aid reloadability, which helps ensure continuity of data access for reports and analytics. This stored version is in SASHDAT format, and in the default libraries listed above is stored on one machine of the CAS system. On a single-machine (SMP) CAS system, this is on the machine on which CAS is running. On a multi-machine (MPP) CAS system, this is on the controller node of the CAS grid. Referring to the library list above, specific file-system paths are created during deployment for each library shown above.

For many Visual Analytics users, this storage copy is just a convenience that facilitates always-available data access. For more advanced users it is helpful to know where the physical files are actually stored in case advanced operations are required. Let's talk about always-available data access first.

The CAS Server is an in-memory server that provides very fast analytics. However, the nature of an in-memory server means that immediately after a server outage, such as for periodic hardware maintenance, data might not be physically resident in memory. Individual data tables can be unloaded and reloaded in SAS Environment Manager as needed to react to these outages, or a capability called Just-In-Time loading can be used in many cases.

Just-In-Time loading means that data that is referred to in an analytics report can be transparently loaded just as it is first needed, providing a seamless user experience for the consumer of that report. In our earlier example of hardware maintenance being performed on a CAS Server, once that machine is back up and running, new requests for data tables for reports that refer to those tables will be satisfied transparently for the user of those reports. In order to provide for that, the system needs a permanent on-disk copy of that data that will be loaded in this way. That is why the importer creates a permanent storage copy of the imported data tables.

When an import is completed to a library like PUBLIC, a storage copy is created and will remain in that location until manually removed by the user or a system administrator. Other libraries behave accordingly. It is worth pointing out that import is supported only for path-based libraries. Those are libraries that store files on disk for access. SAS, and now CAS, support various types of database libraries, but those function differently in Visual Analytics and Viya, so we'll discuss them next.

WORKING WITH OTHER LIBRARIES

Using SAS Environment Manager, other types of libraries can be created and used. These libraries are of the same types of databases from which imports can be performed. The difference is that these libraries are permanent and have access to all tables in the area of that database and schema. For example, a library that is defined for an Oracle server and schema allows access to all tables in the Oracle schema, and can be used directly in Visual Analytics.

Once database library data is used in reports, the Just-In-Time process works as described above. When a DBMS table is referenced from a report, it will be transparently loaded for the user of that report so that their data requests are satisfied. Some library types use distributed storage, which means the data can be accessed in parallel for quicker data load throughput.

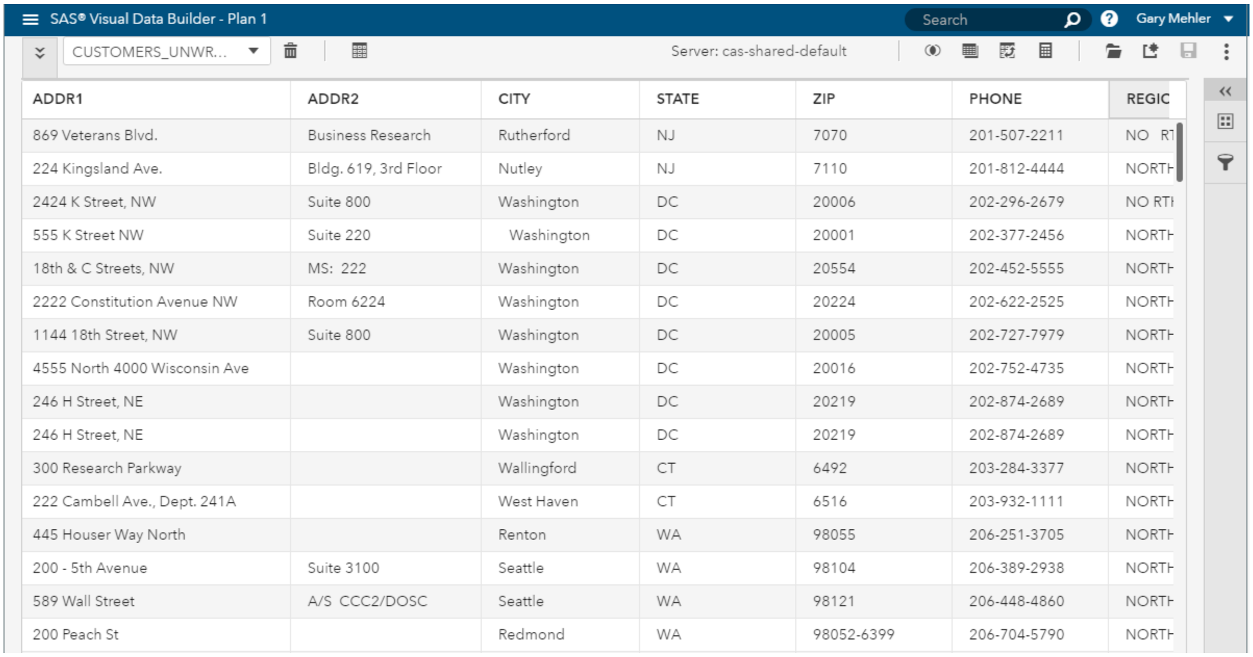
Another consideration for other libraries is that different authorization settings can be applied and managed in SAS Environment Manager. This is of vital importance for any sensitive data that is used by Visual Analytics users.

Related to authorization settings is the ability to protect stored files at-rest. To enable that, a setting can be checked for that library, and an associated encryption key specified. When used with a strong encryption key, this helps to ensure that anyone who can see the storage won't be able to use it in an unauthorized way. We will look at authorization settings for data later in this paper.

WRANGLING YOUR DATA

We've already discussed how data is added and accessed by Visual Analytics, so let's turn our attention to other ways to get it ready. For this section, we'll be using SAS Visual Data Builder. After selecting a table of interest, a tabular display of table values is shown. It is also a good time to familiarize yourself with available functionality in the application that we'll step through here.

The figure below shows the basic layout of SAS Visual Data Builder. The main area shows the rows and columns of your data, much like a spreadsheet would show. Menu options are at the top, and the right side is used for purposes that we will discuss below.

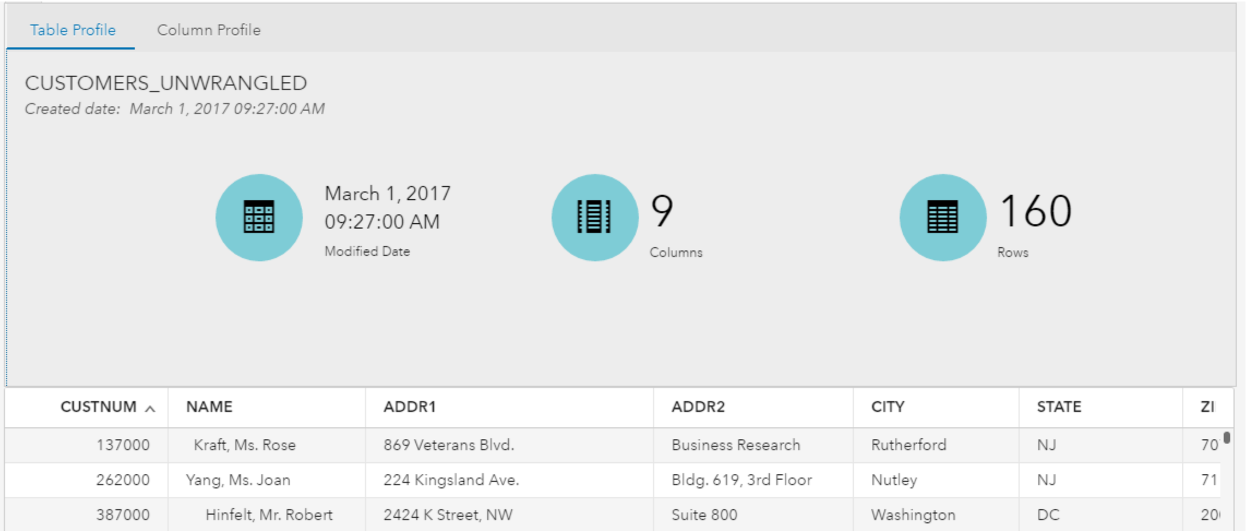


ADDR1	ADDR2	CITY	STATE	ZIP	PHONE	REGIO
869 Veterans Blvd.	Business Research	Rutherford	NJ	7070	201-507-2211	NO RT
224 Kingsland Ave.	Bldg. 619, 3rd Floor	Nutley	NJ	7110	201-812-4444	NORTH
2424 K Street, NW	Suite 800	Washington	DC	20006	202-296-2679	NO RTH
555 K Street NW	Suite 220	Washington	DC	20001	202-377-2456	NORTH
18th & C Streets, NW	MS: 222	Washington	DC	20554	202-452-5555	NORTH
2222 Constitution Avenue NW	Room 6224	Washington	DC	20224	202-622-2525	NORTH
1144 18th Street, NW	Suite 800	Washington	DC	20005	202-727-7979	NORTH
4555 North 4000 Wisconsin Ave		Washington	DC	20016	202-752-4735	NORTH
246 H Street, NE		Washington	DC	20219	202-874-2689	NORTH
246 H Street, NE		Washington	DC	20219	202-874-2689	NORTH
300 Research Parkway		Wallingford	CT	6492	203-284-3377	NORTH
222 Cambell Ave., Dept. 241A		West Haven	CT	6516	203-932-1111	NORTH
445 Houser Way North		Renton	WA	98055	206-251-3705	NORTH
200 - 5th Avenue	Suite 3100	Seattle	WA	98104	206-389-2938	NORTH
589 Wall Street	A/S CCC2/DOSC	Seattle	WA	98121	206-448-4860	NORTH
200 Peach St		Redmond	WA	98052-6399	206-704-5790	NORTH

Figure 5. SAS Visual Data Builder

PROFILING: UNDERSTANDING YOUR DATA

In addition to looking at rows and columns of data, a good way to get started with any new data is to look at basic profiling information. By selecting the upper left arrow in Visual Data Builder, the profiling panel is exposed to reveal information about the table that's being viewed. This is a good time to spot check the information to confirm that the size of the table in terms of rows and columns is what's expected. This is also a helpful panel to review after some types of data manipulations are performed to confirm that the results are generally as expected. In the figure below, I can validate that the right version of data is being used as well as it having the expected number of rows and columns.



CUSTOMERS_UNWRANGLED
Created date: March 1, 2017 09:27:00 AM

March 1, 2017
09:27:00 AM
Modified Date

9
Columns

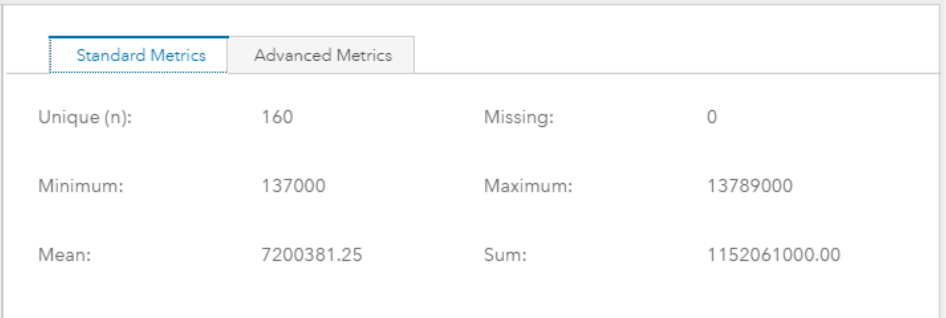
160
Rows

CUSTNUM ^	NAME	ADDR1	ADDR2	CITY	STATE	ZI
137000	Kraft, Ms. Rose	869 Veterans Blvd.	Business Research	Rutherford	NJ	70
262000	Yang, Ms. Joan	224 Kingsland Ave.	Bldg. 619, 3rd Floor	Nutley	NJ	71
387000	Hinfelt, Mr. Robert	2424 K Street, NW	Suite 800	Washington	DC	20

Figure 6. Table Profiling

As you are determining what you might need to do with the data, column profiling is often the next stop. Basic column profiling is useful to see what might need to be done to the data for analytics to function correctly. For numeric values, reviewing minimum, maximum, and missing value metrics can help confirm that the ranges are correct. For character values, seeing the number of distinct values and distribution can be useful to ensure that your data doesn't have outliers or a distribution that's not what was expected.

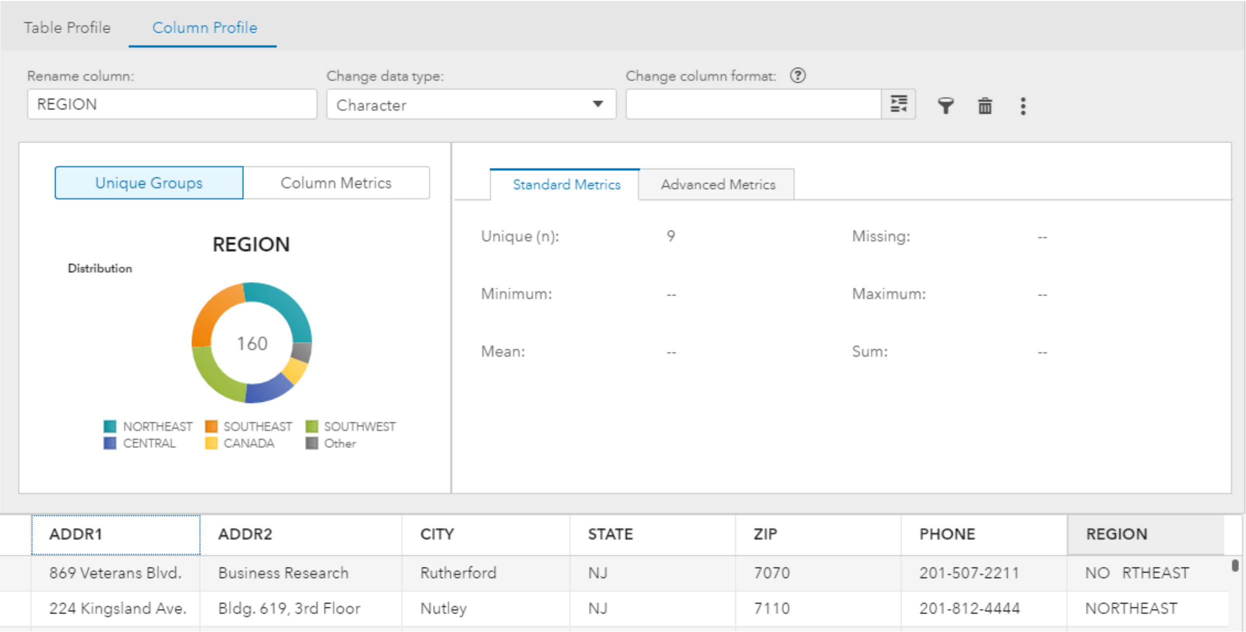
In the example above, we can select the first column, CUSTNUM, to see if it is what we expect. Switching to column profile (see Figure 7 below) we see that the data has 160 unique values for CUSTNUM and no missing values. That looks like a key, so if that's expected, we're happy with the quality of that column and can move on.



Unique (n):	160	Missing:	0
Minimum:	137000	Maximum:	13789000
Mean:	7200381.25	Sum:	1152061000.00

Figure 7. Column Profile Data

Let's look at a text column in Figure 5 to see whether that data is correct. Selecting column profile on the column REGION, a few things become apparent. Visual inspection shows that there are misspellings in that column, and the number of unique values is shown as 9. This is a problem if you expect there to be only six regions if the data is correct. Shown below in Figure 8, an easy-to-read chart on the left helps you understand that some inappropriate values maybe present. Later, we will see how to correct problems and use profiling to help confirm the results.



Rename column: REGION

Change data type: Character

Change column format:

Unique (n):	9	Missing:	--
Minimum:	--	Maximum:	--
Mean:	--	Sum:	--

ADDR1	ADDR2	CITY	STATE	ZIP	PHONE	REGION
869 Veterans Blvd.	Business Research	Rutherford	NJ	7070	201-507-2211	NO RTHEAST
224 Kingsland Ave.	Bldg. 619, 3rd Floor	Nutley	NJ	7110	201-812-4444	NORTHEAST

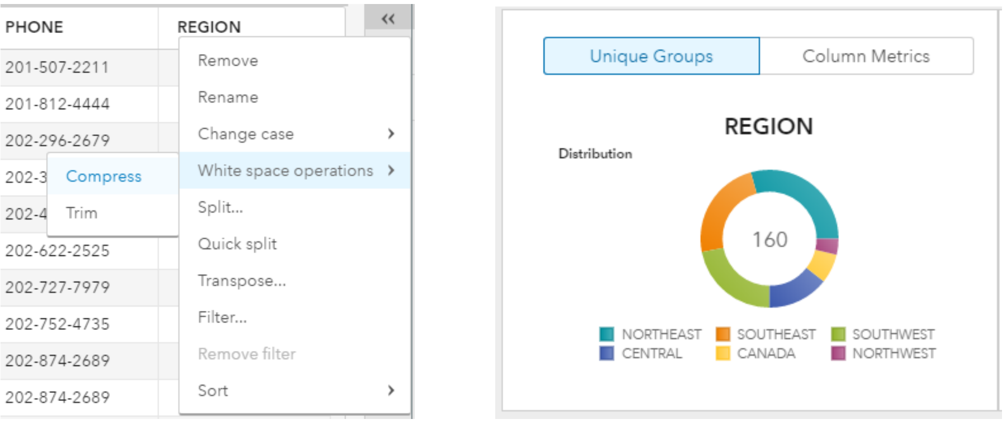
Figure 8. Column Profile for Data That Needs Remediation

While you are reviewing the contents of columns, it is a good time to look at the names and type information for each column. In some cases, new data might not have correct type (numeric versus character) based on source system data. If a data type needs to be changed, you can do that here. That will transform the data to its new representation. The same can be done for the format applied to a column. This might be needed when dates or other types are being used and might not have been read in correctly.

WORKING WITH COLUMNS

A range of column-level functions can be performed on your data in addition to renaming or changing their type or format. Types of common operations include splitting a column based on a delimiter, working with whitespace, or removing unneeded columns. We'll discuss each of these with examples of why they're useful.

In our example above, we saw that the REGION column had values (NORTHEAST and NO RTHEAST) suggesting extra whitespace was present in values of that column, perhaps as a data entry error. Also, reviewing Figure 6 above, note that leading whitespace is present for some values of the NAME column. This is data that needs to have blank characters removed using an operation called Compress. Running a Compress on the REGION column will remove extraneous white space, and a further spot check of the column profile can confirm that we are indeed seeing on the six expected values for that column.



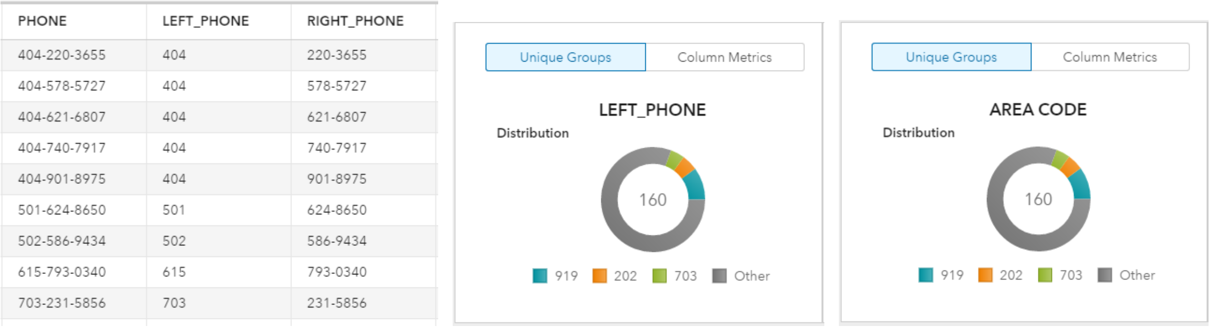
PHONE	REGION
201-507-2211	
201-812-4444	
202-296-2679	
202-3	
202-4	
202-622-2525	
202-727-7979	
202-752-4735	
202-874-2689	
202-874-2689	

Figure 9. Compressing and Trimming Helps Improve Our Data Profile

Had we not detected and corrected this, our analytics would have had inaccurate results, aggregating or otherwise calculating values for REGIONs that don't exist. So, being inquisitive about our data and using basic profiling helped us get to mission accomplished with correcting these quality issues.

TIME TO SPLIT

Another type of common action to perform on columns is to split into separate pieces for various types of analytics. In our example above, you will see a column for PHONE, indicating US area code and phone number. If analytics needs the area code to be split out for a calculation, that can be performed with a split operation. In fact, since the syntax is straightforward, we want to split by the first non-numeric or non-alphabetic character, meaning the hyphen, we can just select the Quick split operation to split on the first delimiter found in the data.



PHONE	LEFT_PHONE	RIGHT_PHONE
404-220-3655	404	220-3655
404-578-5727	404	578-5727
404-621-6807	404	621-6807
404-740-7917	404	740-7917
404-901-8975	404	901-8975
501-624-8650	501	624-8650
502-586-9434	502	586-9434
615-793-0340	615	793-0340
703-231-5856	703	231-5856

Figure 10. Split Creates More Columns for Detailed Analytics; Rename Makes the Column Useful

The output of a Split operation is a set of two columns, which have automatically generated names. In our example, since the initial column was PHONE, Split generates two new columns: LEFT_PHONE and RIGHT_PHONE. As we discussed earlier, you might want to rename columns for various purposes, and this is one of those purposes. LEFT_PHONE is really AREA CODE for analytic purposes, so let's do the rename here and then move on to other areas.

If we had more complicated needs, a general-purpose split is also available. There we can specify the delimiter and other parameters to meet our data needs.

FILTERING DATA

When using data that has many more rows than are needed, filtering can be used to subset the data to a useful section that will be used for analytics. Data can also be filtered dynamically in SAS Visual Analytics, so why would we consider doing that here? The answer has to do with performance and resource utilization. In our example above, if the data were only ever going to be used to analyze information about the region Canada, for example, then transporting data about all the other regions around will require more space to store and to load in memory for CAS. On the performance end of things, if the data needs to be filtered each time to only the Canada region, then every analytic user will be spending some time filtering those values out of their analytic report output. If everyone needs to do it, do it during the data wrangling process.

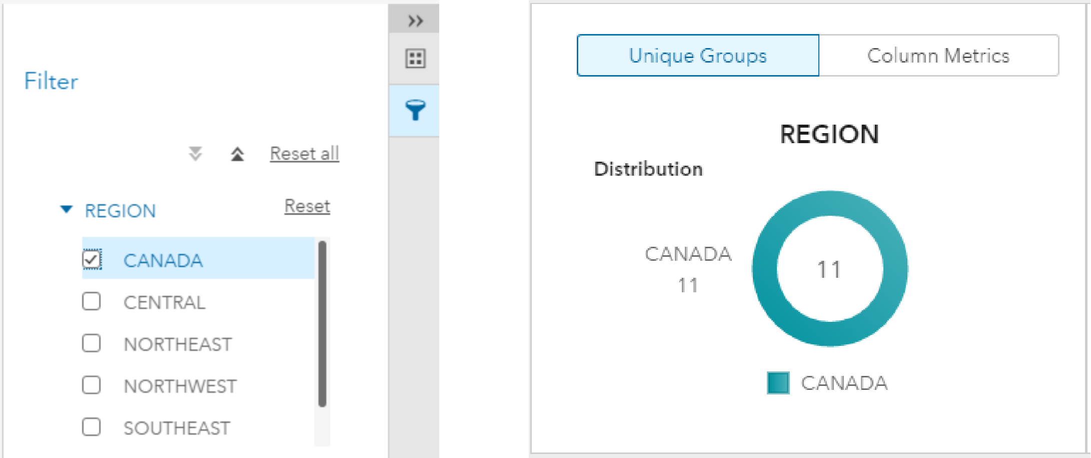


Figure 11. Filtering to a Specific Region and Verifying Results

Figure 11 above shows a filter panel in which a single value or multiple values can be specified. After applying the filter, we can again use the profile data to confirm that we have the right data in our post-filtered table. Whether filtering is done in SAS Visual Data Builder or in SAS Visual Analytics, the visual process is similar. The difference is that filtering performed in SAS Visual Data Builder acts on the table itself, and pre-applies the filter for all analytic reporting users.

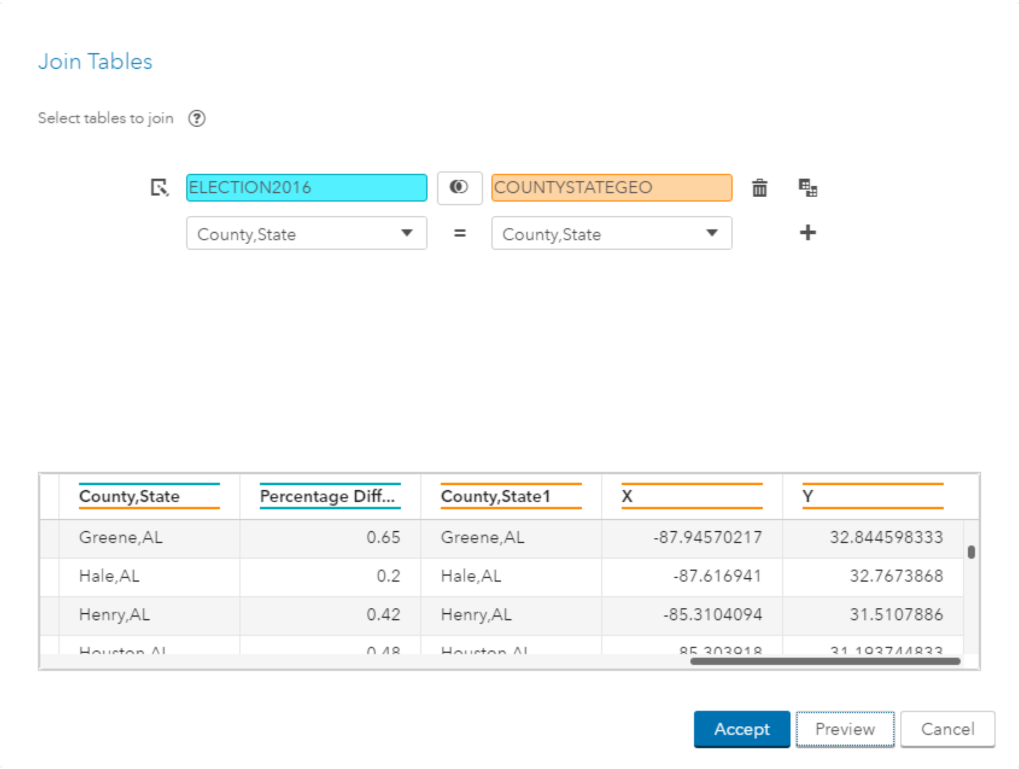
JOINING AND APPENDING TABLES

When preparing data for analytics, join is a very important capability, and we'll see how to visually combine tables for analytic reporting purposes. Note that all tables that are being joined need to be on the same CAS server. Joining is a way to enrich data with values from multiple tables to get a broader set of information on a per-row basis in your data. An example is adding latitude and longitude values to a data table that has U.S. county names, which will allow us to visually look at results on a geographic map.

We will step through an example with some different data to help us geographically map some data. In this case, it is county-based data from a recent United States election. SAS Visual Analytics can geographically map on a number of values like ZIP code, state name, or country name, but doesn't have geographic data at the County level. In that, or other special cases, latitude and longitude values can be graphics. So we will enrich the election data with columns from another table that contains that information. To start, we have loaded two tables: ELECTION2016 and COUNTYSTATEGEO. We want to add latitude and longitude values from the second table to the first, which is shown in Figure 12 below.

The Join operation is started by clicking on the ◉ icon on the application toolbar seen in Figure 5 above. Since we already had two tables loaded, SAS Visual Data Builder attempted to automatically determine the column on which to join. In this case, the column County,State, which exists in both tables, has the same type and compatible length, so it was automatically chosen as the join column. Of course, you can override that suggestion, but it is a good starting point in most cases.

SAS Visual Data Builder supports multi-key and multi-table joins as well as the ability to specify a join type. In our example, we want to perform a left join, meaning all the rows in the left table (election results) and columns from the right table (latitude/longitude per US County) that match the join key.



Join Tables

Select tables to join

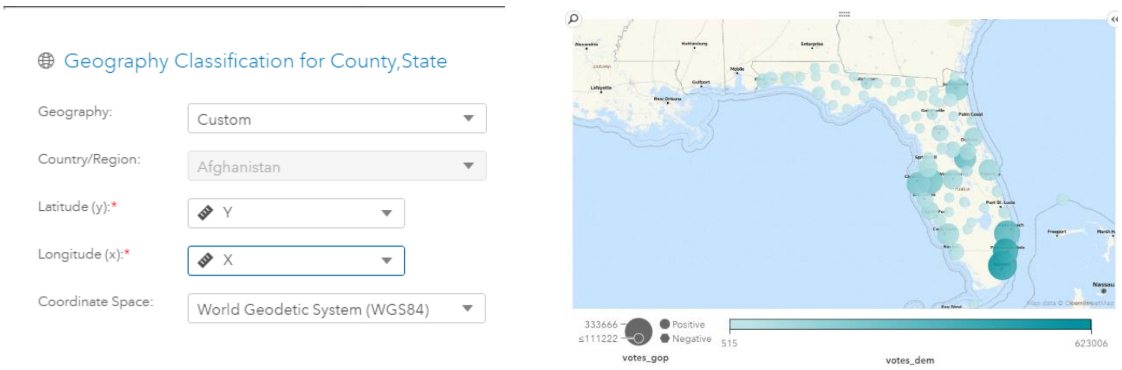
ELECTION2016 | COUNTYSTATEGEO

County,State = County,State

County,State	Percentage Diff...	County,State1	X	Y
Greene,AL	0.65	Greene,AL	-87.94570217	32.844598333
Hale,AL	0.2	Hale,AL	-87.616941	32.7673868
Henry,AL	0.42	Henry,AL	-85.3104094	31.5107886
Houston,AL	0.48	Houston,AL	-85.303918	31.193744833

Figure 12. Previewing a Join Operation on Two Tables

Since Join can be an iterative process, the Join panel has a preview area that can be used to visually inspect output rows to ensure that the expected values are being seen. After accepting the output of the Join, the data is ready to use in SAS Visual Analytics by creating a Geography classification on the variable that will be analyzed. We will be mapping using longitude and latitude values, which we added as X and Y in the previous step. Below we see how that classification is specified and then the outcome of our geographic mapping exercise.



Geography Classification for County,State

Geography: Custom

Country/Region: Afghanistan

Latitude (y):* Y

Longitude (x):* X

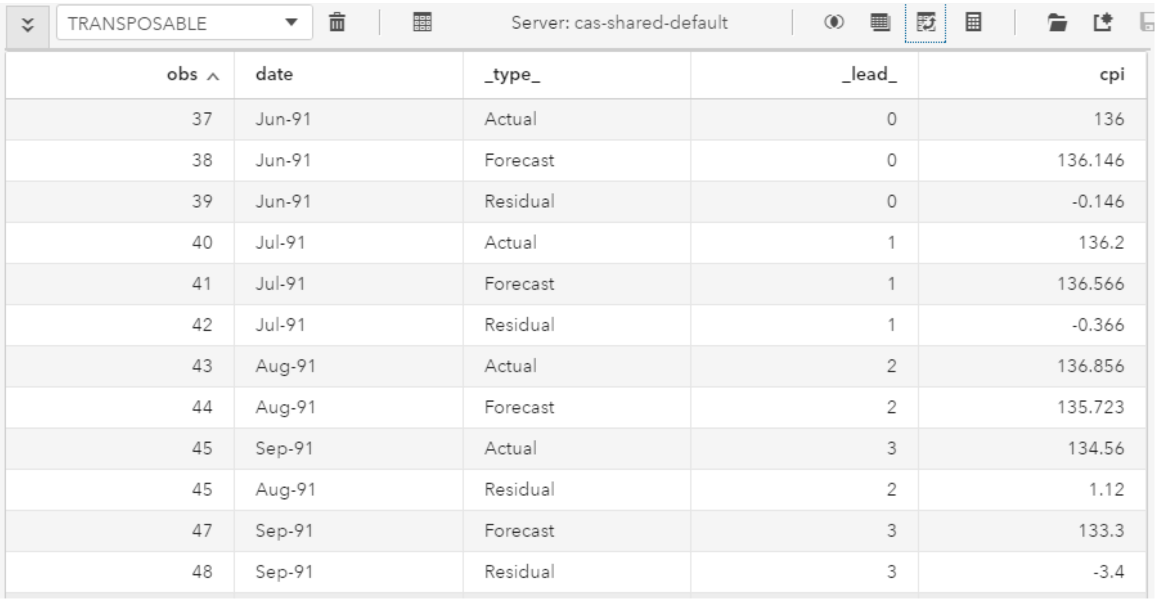
Coordinate Space: World Geodetic System (WGS84)

Figure 13. Using Added Geographic Data to Facilitate Mapping of Data

Although we won't discuss in detail here, it is also straightforward to append data from different tables together. That's another important consideration when making sure your data is ready for analysis since stale data isn't always what is expected. Append will let you add rows from a new table to a base table, and is helpful for data that is incrementally updated over time. A typical example is a set of basic sales data that has new sales added to it nightly. Appending the new data to the current base table helps make sure that the freshest data is always available.

TRANSPOSING DATA

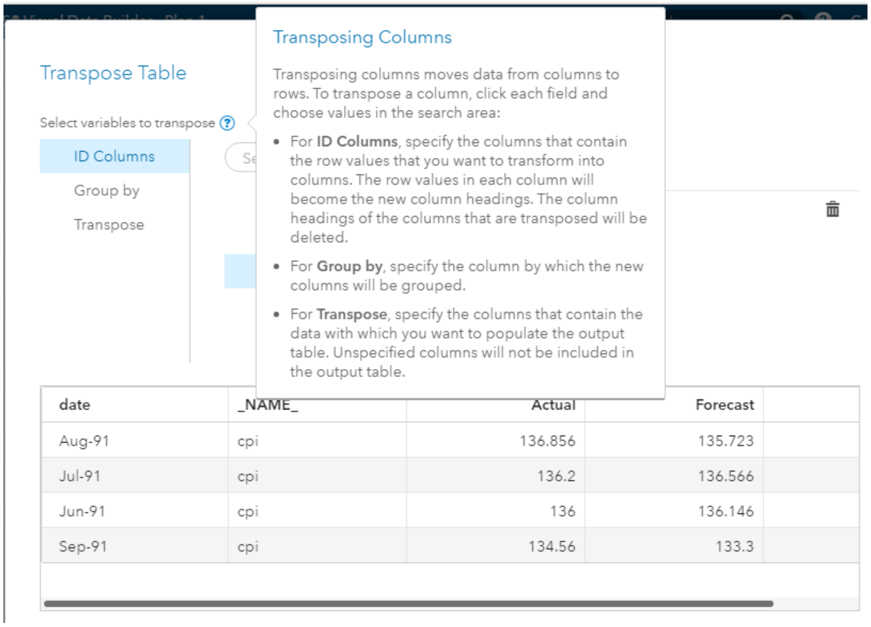
One last type of data transformation we'll look at involves the need to rotate or transpose data. It isn't unusual to receive data that hasn't been structured to easily run analytics. Some data can look like a time series that have interleaved values in a single column, which makes it difficult to aggregate as one might expect. In Figure 14 below we see such a table. The values for cpi are something we'd like to be able to average by _type_, meaning by Actual, Forecast, and Residual.



obs	date	_type_	_lead_	cpi
37	Jun-91	Actual	0	136
38	Jun-91	Forecast	0	136.146
39	Jun-91	Residual	0	-0.146
40	Jul-91	Actual	1	136.2
41	Jul-91	Forecast	1	136.566
42	Jul-91	Residual	1	-0.366
43	Aug-91	Actual	2	136.856
44	Aug-91	Forecast	2	135.723
45	Sep-91	Actual	3	134.56
45	Aug-91	Residual	2	1.12
47	Sep-91	Forecast	3	133.3
48	Sep-91	Residual	3	-3.4

Figure 14. Data that's Not Quite Ready for Analytics

To complete our work with this data, a quick transposition of cpi for each _type_ and having an output row for each date is a better format for our analysis. This is performed by using the Transpose panel, in which results can be previewed to ensure that the correct transposition has been performed, as we see below in Figure 15.



Transpose Table

Select variables to transpose ?

- ID Columns
- Group by
- Transpose

Transposing Columns

Transposing columns moves data from columns to rows. To transpose a column, click each field and choose values in the search area:

- For **ID Columns**, specify the columns that contain the row values that you want to transform into columns. The row values in each column will become the new column headings. The column headings of the columns that are transposed will be deleted.
- For **Group by**, specify the column by which the new columns will be grouped.
- For **Transpose**, specify the columns that contain the data with which you want to populate the output table. Unspecified columns will not be included in the output table.

date	_NAME_	Actual	Forecast
Aug-91	cpi	136.856	135.723
Jul-91	cpi	136.2	136.566
Jun-91	cpi	136	136.146
Sep-91	cpi	134.56	133.3

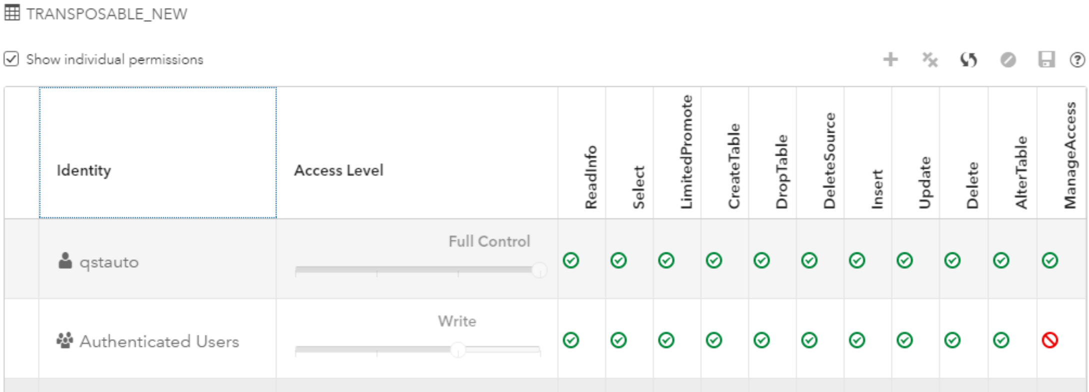
Figure 15. Transposing Data for Easier Aggregation

The rows of values shown in the new columns for Actual and Forecast (as well as Residual, not shown in the figure above) are now ready for basic analysis.

FINISHING UP

To complete our work, you will want to save our transformed table to a location that will be consumable by people in your organization who are going to be performing analytic reporting. Output can be saved to the same or another library based on that need. At this point we've prepared the data carefully, and it is a good idea to think about authorized access carefully too.

When saving your output table, you can choose a location that has the right level of access controls if your data is sensitive. In the figure below, table permissions are displayed that control who has access. The top row shows that the table owner has full control, while all Authenticated Users have Read and Write permission, but not the ability to change permission on that table. These are set because the table was saved to the PUBLIC library, which sets the basic permission scheme for tables that are stored there.



TRANSPOSABLE_NEW

☑ Show individual permissions

Identity	Access Level	ReadInfo	Select	LimitedPromote	CreateTable	DropTable	DeleteSource	Insert	Update	Delete	AlterTable	ManageAccess
qstauto	Full Control	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Authenticated Users	Write	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	🚫

Figure 16. Permissions for a Table

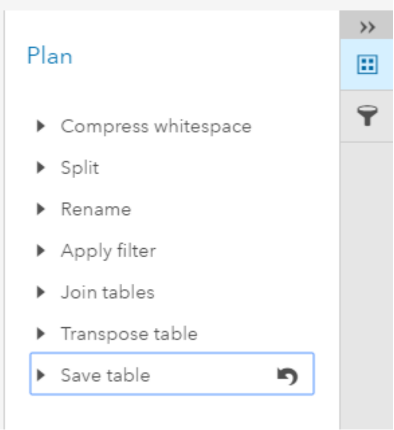
Although there are a rich set of permissions possible on the right side of the panel, basic permission changes can be made using the Access Level slide control in the middle of the panel. In this area, four basic settings can be easily selected: no access, read-only, read/write, or full control.

PLANS: WHAT'S BEEN DONE AND CAN BE RE-RUN

So far, we've seen a series of transformational operations performed to work with whitespace, change types, filter, remove or rename a column, join, or transpose. It is helpful that we've transformed our data for analysis, but what if we want to either modify those steps or re-run at a later date on new data?

Both of these are possible because the steps we've been performing are saved in a Data Preparation Plan. The Plan is a sequence of steps that can be reviewed at any time and can have prior steps reversed, or undone. When your data work for your session is complete, you can save the plan for another day. Re-opening that plan will re-run the steps that have been saved in that plan, effectively replaying those steps. If the steps are complete for your goal to transform the data, that's all you need to do.

The figure below shows the Plan of steps performed in this paper. The list of steps has an order, of course, and each step can be expanded to see details. If there are incorrect steps that are no longer needed, the last step can be undone, as indicated in the figure below with the back-arrow icon next to Save table.



Plan

- Compress whitespace
- Split
- Rename
- Apply filter
- Join tables
- Transpose table
- Save table

Figure 17. A Data Preparation Plan Containing the Steps We've Performed So Far

If you saved the Plan at the end of one day to continue that work later, re-opening not only re-runs the steps to transform the data to the point you left off, but you can add more steps to the Plan if that's needed. Plans can be saved in multiple copies and versions, either to checkpoint your work, or to have smaller sets of steps that can be run at some later date.

CONCLUSION

SAS Visual Analytics can provide high-quality analytic reports but needs high-quality data to ensure the best outcome. Capabilities in SAS Visual Data Builder to import and wrangle data can help you rise to the challenge. Making use of basic profiling to explore your data and then to remedy any issues is a good way to step through the wrangling process.

REFERENCES

SAS support site (http://support.sas.com/documentation/)

RECOMMENDED READING

- “SAS Visual Analytics.” SAS Institute Inc. Available https://www.sas.com/en_us/software/business-intelligence/visual-analytics.html
- SAS Viya Documentation http://support.sas.com/documentation/onlinedoc/viya/

RESOURCES

SAS Institute Inc. 2017. *SAS Viya 3.2 Administration*. Cary NC: SAS Institute Inc. Available at http://support.sas.com/documentation/prod-p/cal/index.html

CONTACT INFORMATION

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

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