Using SAS® Enterprise Guide in SAS® Viya

Charu Shankar
Senior Technical Training Consultant at SAS

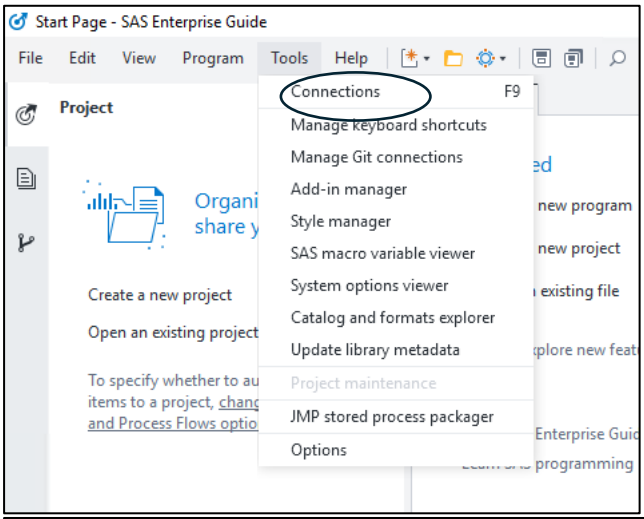
Demo 1: Connecting to a SAS Viya Server

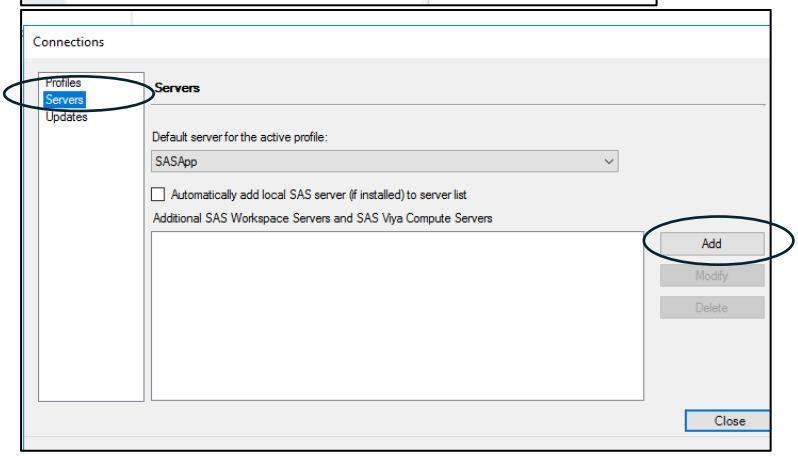
In this demonstration, we use SAS Enterprise Guide and establish a server connection to SAS Viya. This integration enables you to continue using your SAS Enterprise Guide Windows application and offers enhanced capabilities, such as providing access to a cloud-native, high-performance analytics platform that supports multi-language coding.

Currently SAS Enterprise Guide must be licensed through SAS 9, but a standalone installer will soon be available that negates this requirement. For SAS Viya, SAS Studio Analyst or SAS Studio Engineer must be licensed.

Let's begin by adding a new SAS Server connection.

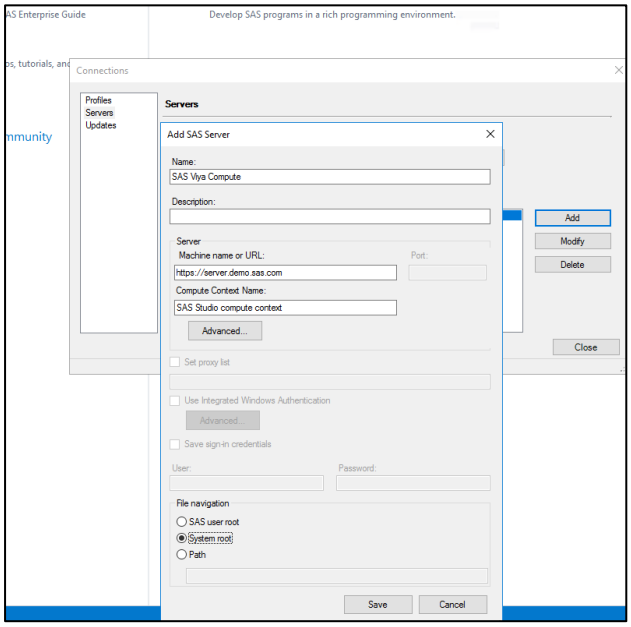
1. Open SAS Enterprise Guide.
2. Use the application toolbar in SAS Enterprise Guide and select **Tools, Connections** and in the Connections window select **Servers** and click **Add**.





3. In the **Add SAS Server** window, enter **SAS Viya Compute** as the server name, or chose a name of your preference. This name will be used to identify the server within the Enterprise Guide interface. You can also include an optional description.

4. Next, specify the SAS Viya URL in the **Machine name or URL** text field. The URL must begin with either http or https, in our example, it's https://server.demo.sas.com**.**



If your SAS Viya environment is version 2024.02 or later, as in our case, this is the minimum required information to establish a connection unless a dedicated client ID has been registered. For earlier releases of SAS Viya, a dedicated client ID and secret must be registered by your SAS Administrator and entered using the Advanced settings. Please contact your SAS Administrator for site-specific requirements.

5. Select **System root** as the starting point for the file navigation.

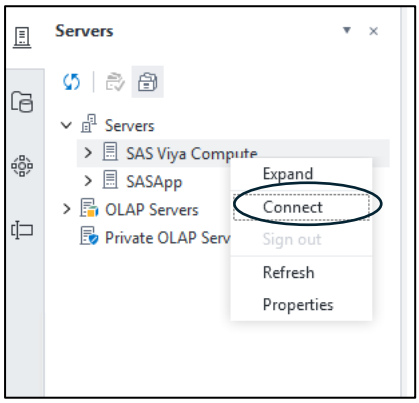
6. Click **Save.**

7. In the **Connections** window, you have the opportunity to set the default server. I will choose to have **SASApp** as the default server but switching to the SAS Viya Compute server can offer better performance, enhance resource optimisation and improve collaboration.

8. Click **Close**.

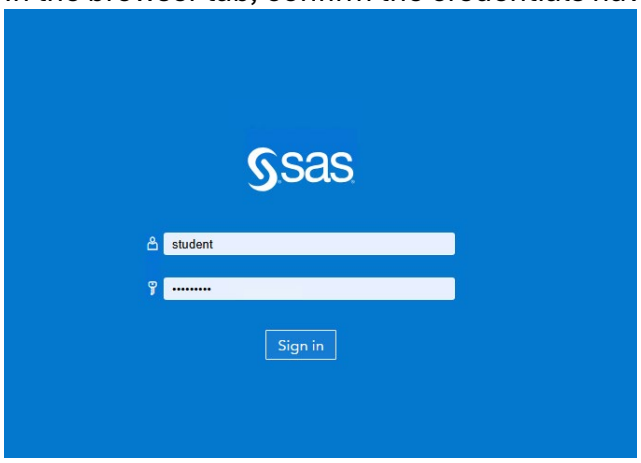
With the connection to the SAS Viya Compute server established, we can now connect to it.

9. In the **Servers** pane, **right-click** the **SAS Viya Compute** server and click **Connect**.



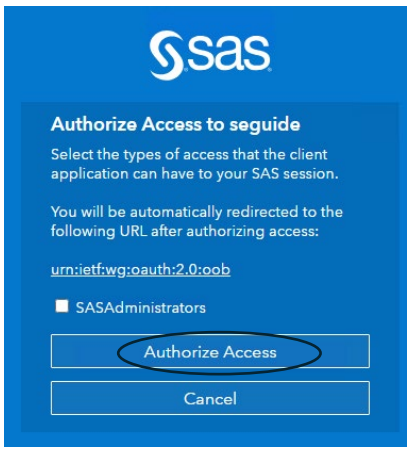
Upon initial connection, you will be prompted to provide an authorization code based on your user credentials.

10. In the browser tab, confirm the credentials have been entered and click **Sign In**.

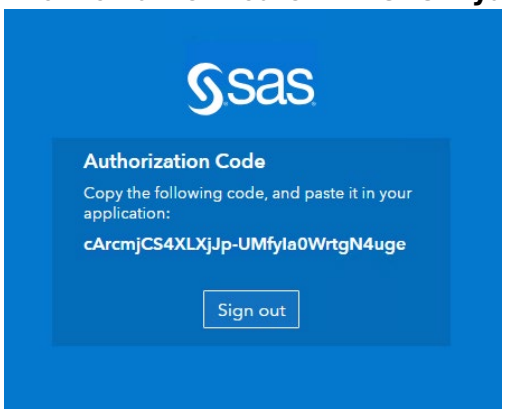


If the SAS Viya environment is configured for Single Sign-On (SSO) no credentials prompt will appear.

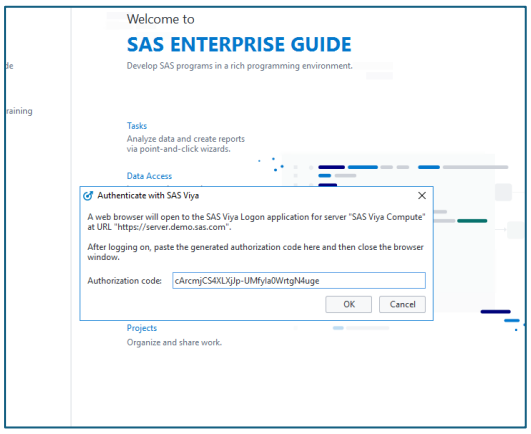
11. Admin access is not required so click **Authorize Access** to generate an authorization code.



12. Copy the authorization code and return to SAS Enterprise Guide and paste the code into the **Authenticate with SAS Viya** pop-up window. Click **OK**.

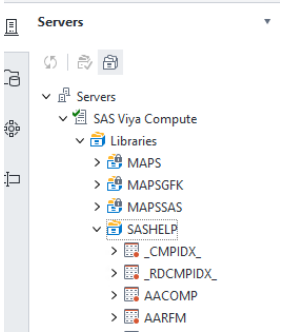


Reauthorization of your credentials will be required if you sign out of the SAS Viya Compute server in SAS Enterprise Guide and may also be required after 90 days, though this default can be adjusted by your SAS Viya administrator.

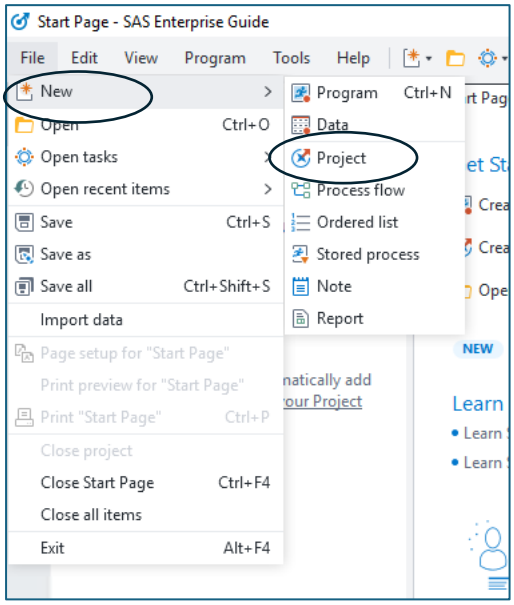


SAS Enterprise Guide will now start the SAS Viya compute session, allowing us to browse the list of SAS libraries and navigate the file system linked to the SAS Viya Compute server.

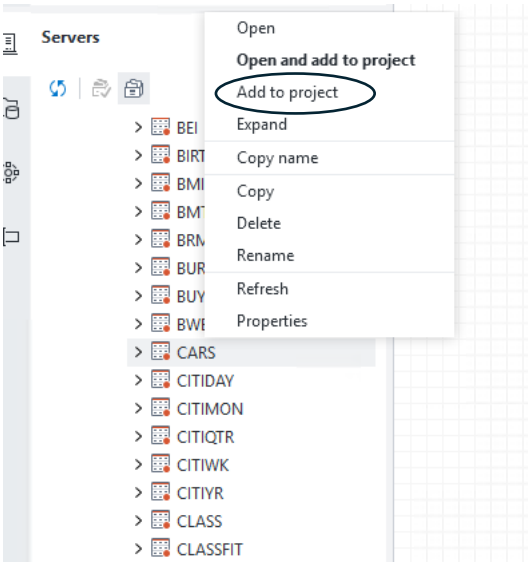
13. Expand **Libraries**, **SASHELP** and double-click **CARS** to open the table in the table viewer. As with SAS 9 connections, you can filter rows using the table viewer, however, for SAS Viya connections, the table viewer is read-only. Close the table viewer. No need to save changes to the project.



14. Use the application tool bar in SAS Enterprise Guide and click **File**> **New**>**Project**.



15. Right-click the **CARS** SAS table from the **SASHELP** library on the SAS Viya Compute server and select **Add to project**. Use the mouse to hover over the **CARS** Table shortcut on the canvas and verify that the server is SAS Viya Compute. Any tasks associated with this Table shortcut will be executed on the SAS Viya Compute server.



Multiple servers can be used in the same process flow. If a task is created in this process flow that uses data that resides in the SASApp server, and the complete process flow is executed, the tasks will run in parallel.

16. Return to the application tool bar in SAS Enterprise Guide and click **File**> **New**> **Program**. The new SAS Program will execute on the default server, however, the SAS Program toolbar provides a server selection dropdown menu, allowing you to select the execution environment.

17. On the SAS Enterprise Guide toolbar, select **File** and **Close project**. You do not need to save the project.

Features that Require a SAS Metadata Server

Certain features in SAS Enterprise Guide require a connection to a SAS Metadata Server. However, access to a SAS Metadata Server and SAS Metadata Repository is not available when working with a local server or using server definitions for SAS Viya compute servers and SAS 9 workspace servers.

The following features rely on a SAS Metadata Server so are not available when connecting to SAS Viya or a local server:

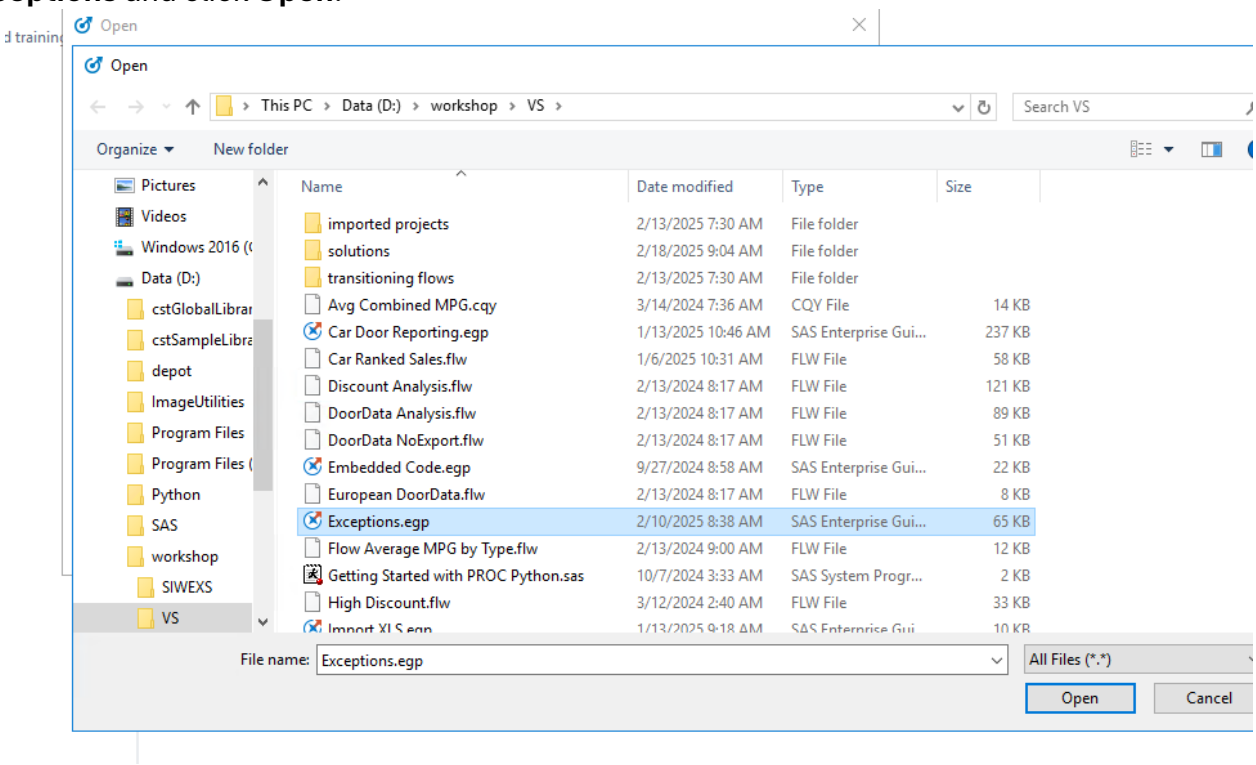
- Stored processes
- Information maps
- Tables, cubes, and jobs that are registered in metadata
- SAS language elements that are specific to SAS metadata:
    - Procedures that are specific to metadata, such as PROC METADATA and PROC METALIB
    - Metadata DATA step functions
    - Metadata LIBNAME statements
- Project prompts that rely on metadata:
    - Dynamically populated text or numeric prompts
    - Data source prompt
    - Data source item prompt
    - Data library prompt

Demo 2: Updating Existing SAS Enterprise Guide Projects

When working with a project created to run on SAS 9, you may need to convert nodes or the entire project to run on the SAS Viya Compute server.

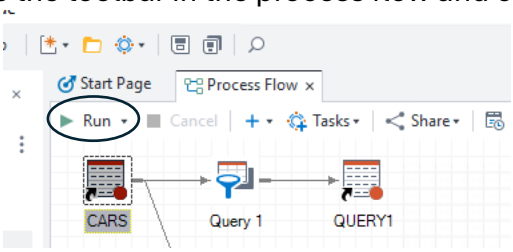
We'll start by modifying individual nodes followed by an alternative method to update all nodes using the Project Maintenance Wizard.

1. Use the application toolbar in SAS Enterprise Guide and select **File** and **Open**. Select **My Computer**, click **Browse** and navigate to **Data (D:)**, **workshop**, **VS**. Select **Exceptions** and click **Open**.

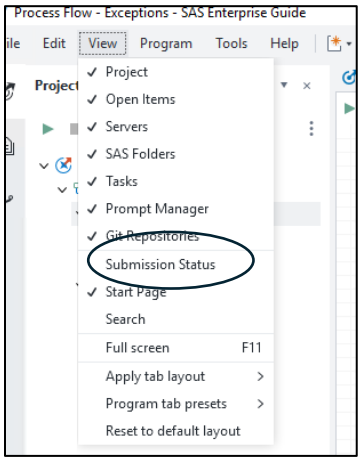


This is a simple project with one process flow, two queries and a SAS Studio Bar Chart Task.

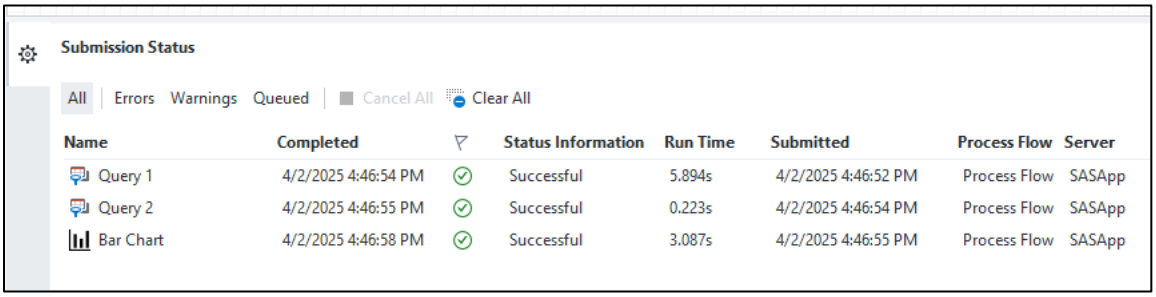
2. Use the toolbar in the process flow and click **Run**.



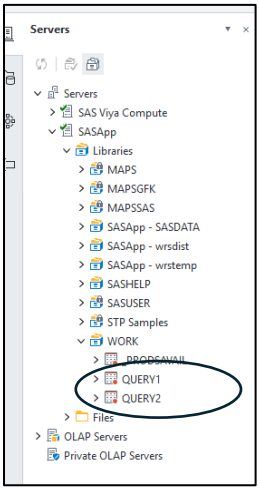
3. Use the application toolbar in SAS Enterprise Guide to select **View** and **Submission Status**. This pane enables you to view and manage the status of submitted programs, tasks and stored processes.



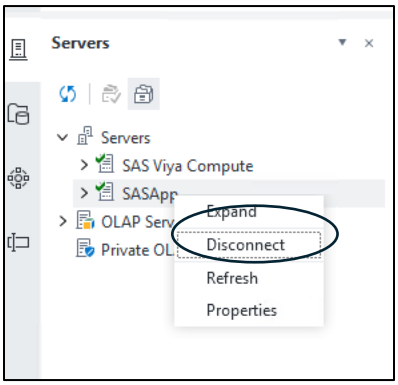
4. The Submission Status pane adds additional information including the submission and completion times and the server each operational node is executed on. In this case the SASApp server which is the SAS 9 server.



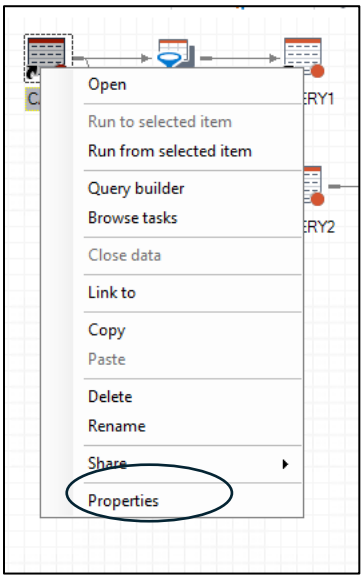
5. To avoid confusion, let's disconnect from the SASApp server which will delete the QUERY1 and QUERY2 tables from the WORK library.



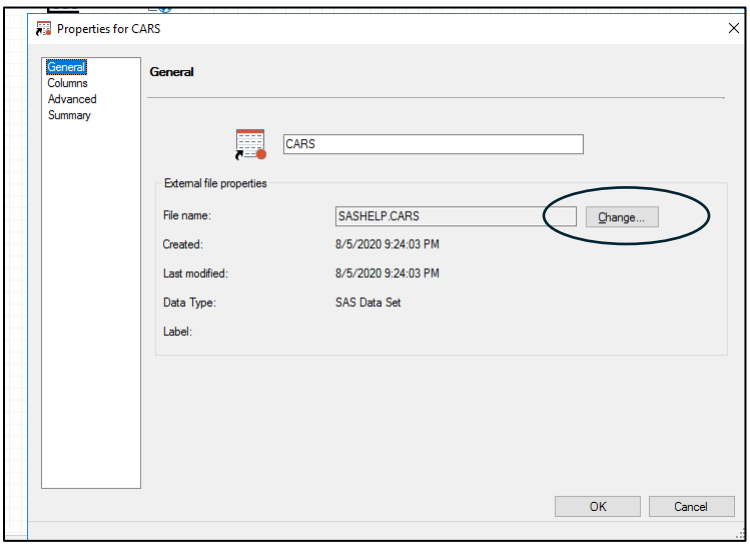
6. Using the Servers pane, right-click the **SASApp** server and select **Disconnect**.



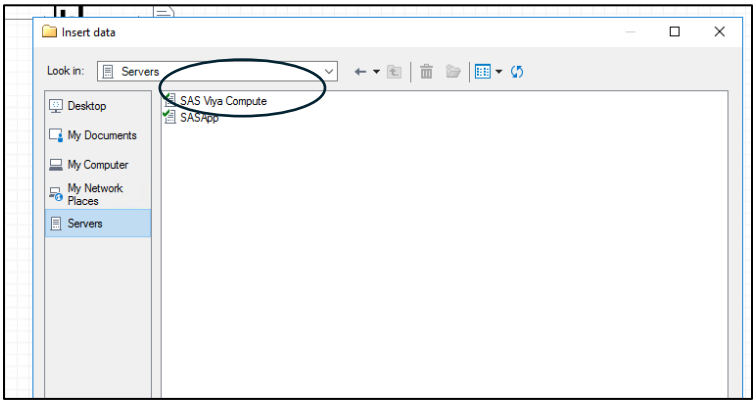
7. Right-click the CARS Table node and select **Properties**.

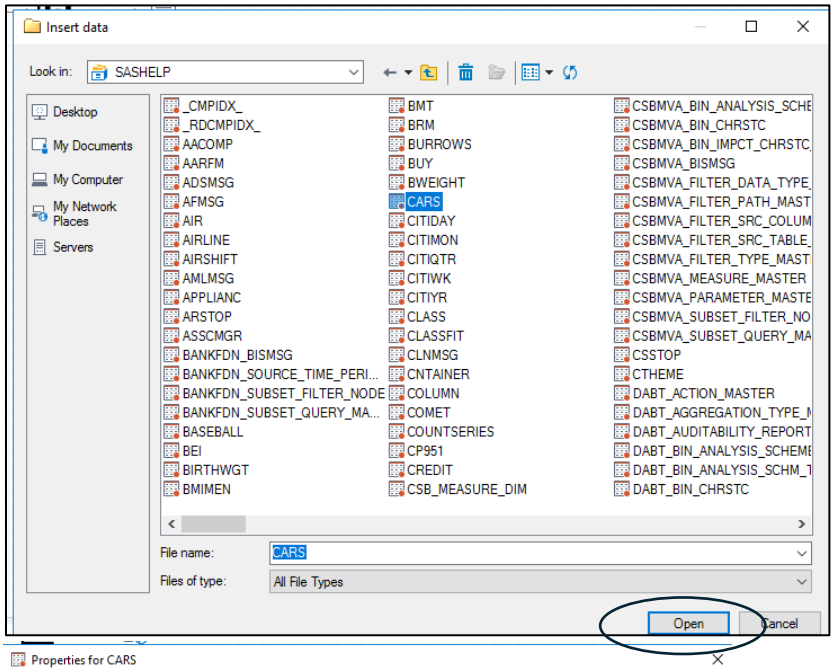


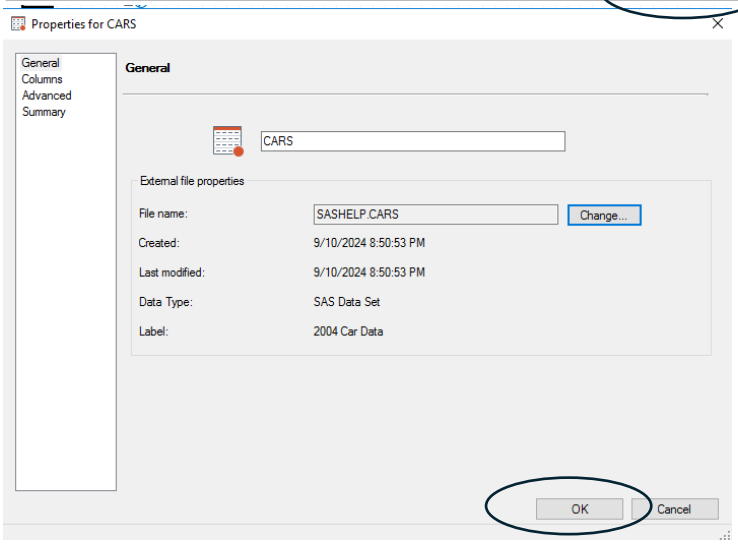
8. Click **Change** and select, **Servers**.



9. Double click, **SAS Viya Compute**, **Libraries** and **SASHELP** and select the **CARS** table. Click **Open** and click **OK**.



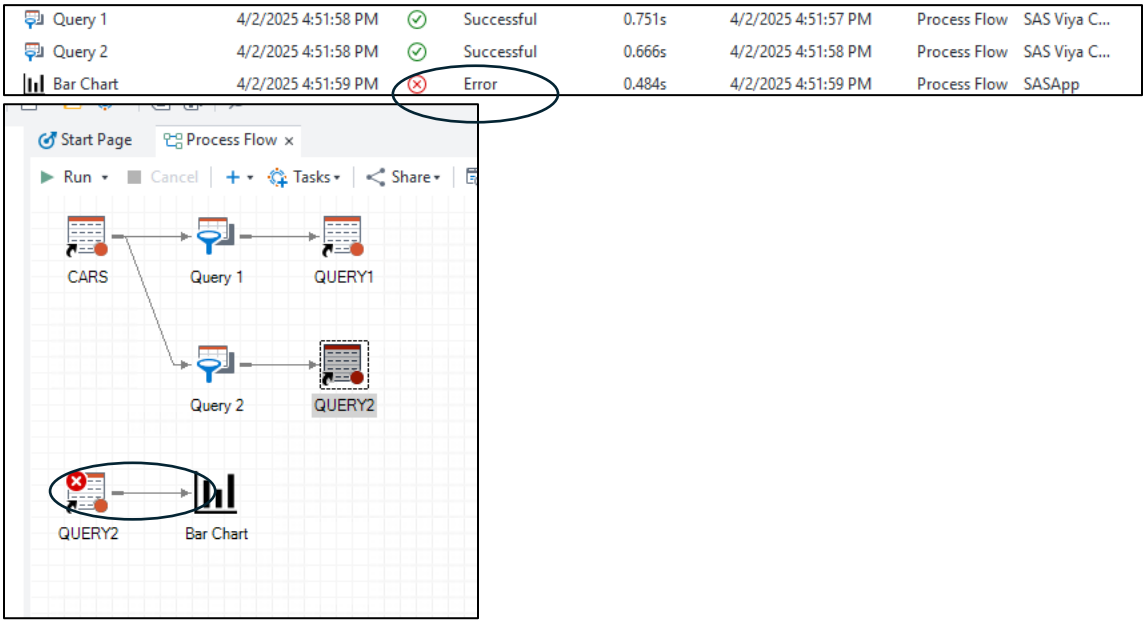




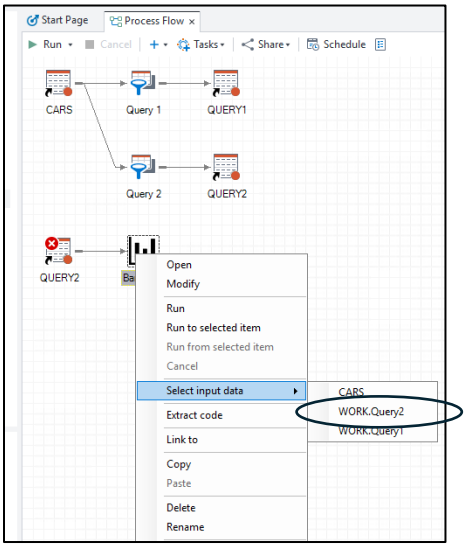
10. Use the toolbar in the process flow and click **Run**.

11. The Submission Status window shows that Query 1 and Query 2 were submitted on the SAS Viya Compute server, with both queries generating new data nodes that reference this server. However, the SAS Studio Bar Chart references the WORK.QUERY2 table on the SASApp server which has not been created, leading to a task error.

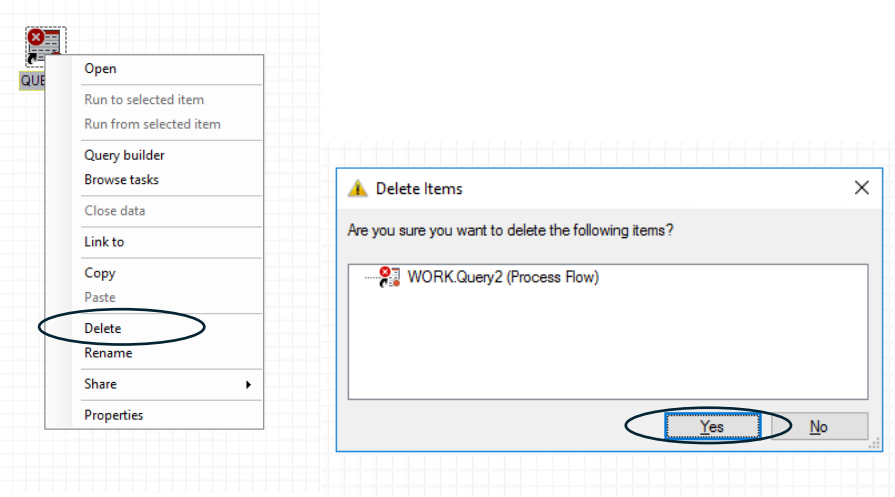
A.SAS.studio.task.icon.renders.in.black.whereas.a.SAS.Enterprise.Guide.task.renders.in.up.in.blue¡



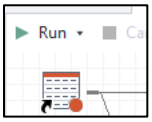
12. The reference for the SAS Studio Bar Chart task can be easily updated by right-clicking the task labelled B**ar Chart,** selecting **Select input data**, and choosing **WORK.QUERY2**.



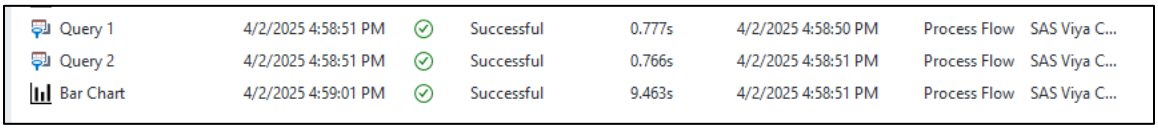
13. Right-click the redundant **QUERY2** Table node, select **Delete** and click **Yes** to confirm deletion.



14. Use the toolbar in the process flow and click **Run**.

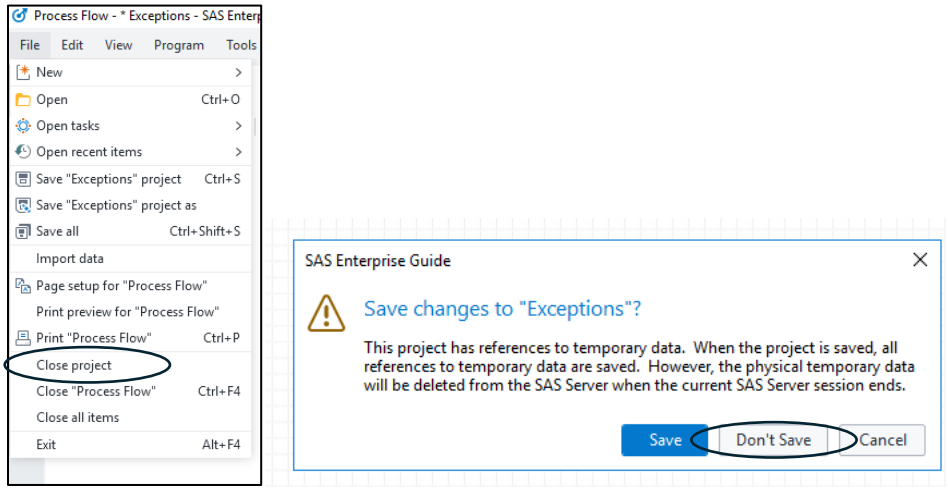


Use the Submission Status window to confirm that all operational nodes were successful and were executed on the SAS Viya Compute server.



An alternative and more efficient method is to use the Project Maintenance wizard to update a variety of references including reassigning servers and libraries.

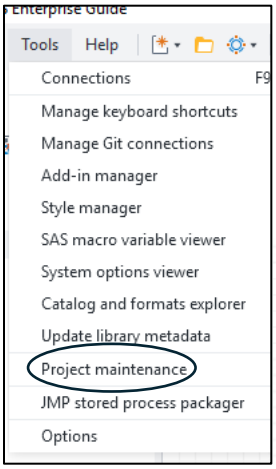
15. Use the application toolbar in SAS Enterprise Guide and select **File** and **Close project**. You do **not** need to save the project.

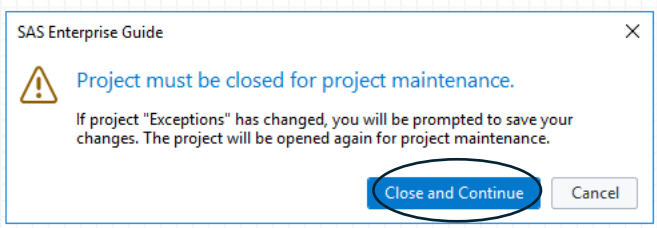


16. Use the **Recent Items** list in the **Start Page** tab and select **Exceptions.egp** to reopen the project.

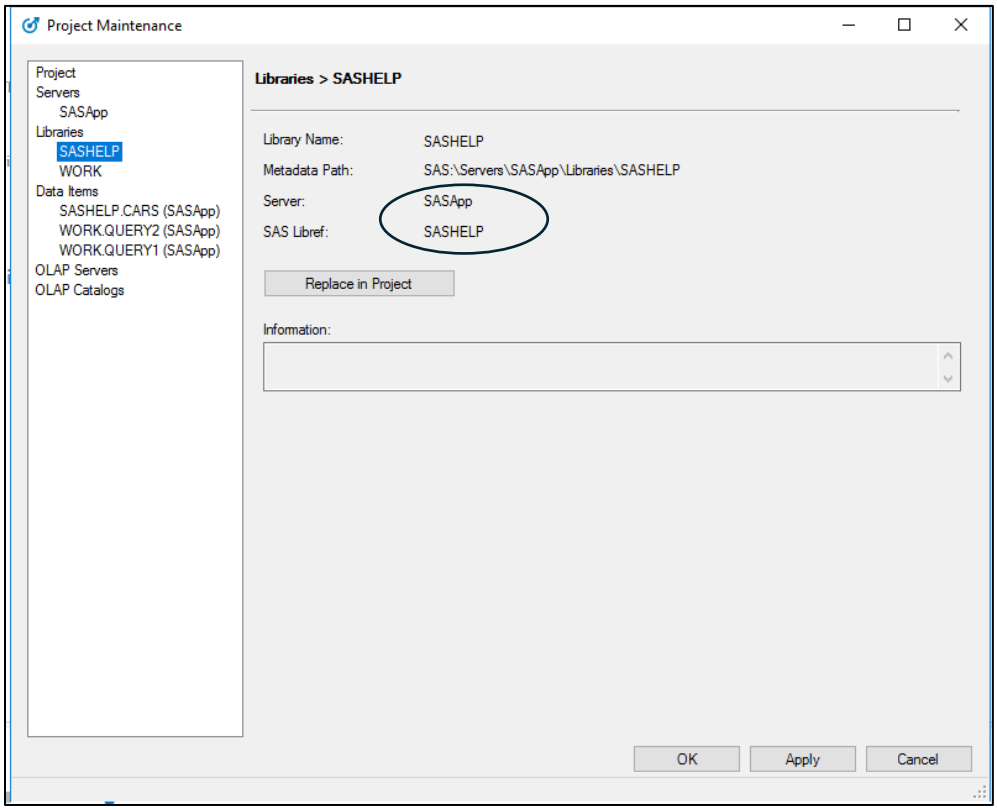
    Project Maintenance will overwrite the existing project. If required, make a copy of the project prior to running the Project Maintenance to create a backup.

17. Use the application toolbar and click **Tools** and **Project Maintenance**. Projects must be closed to proceed. Click **Close and Continue**.

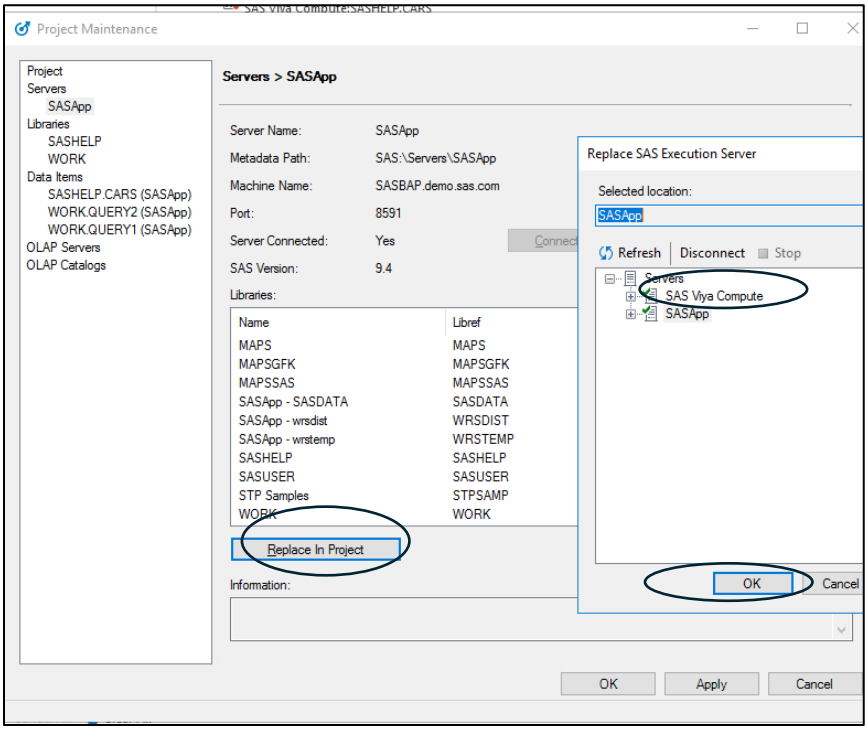




18. Select **Libraries** in the navigation pane and confirm that SASHELP is currently located on the **SASApp** server.

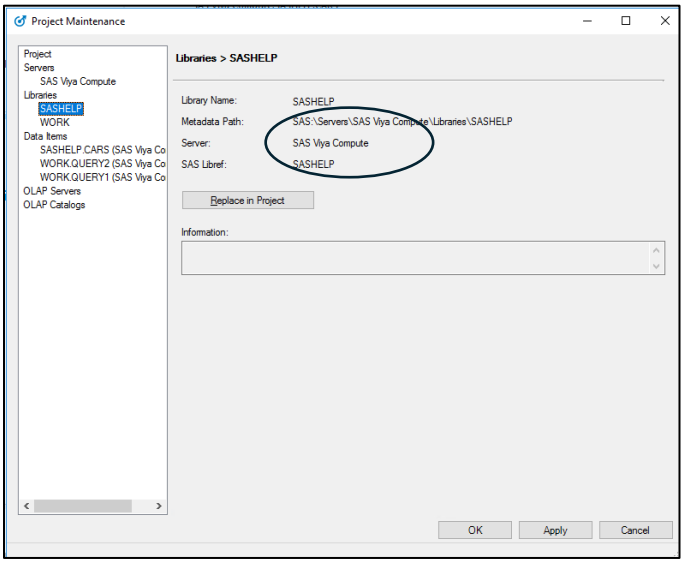


19. Select **Servers** in the navigation pane. You have the option to replace the execution server in the project. Select **Replace In Project**, select the new **SAS Viya Compute** server and click **OK**.

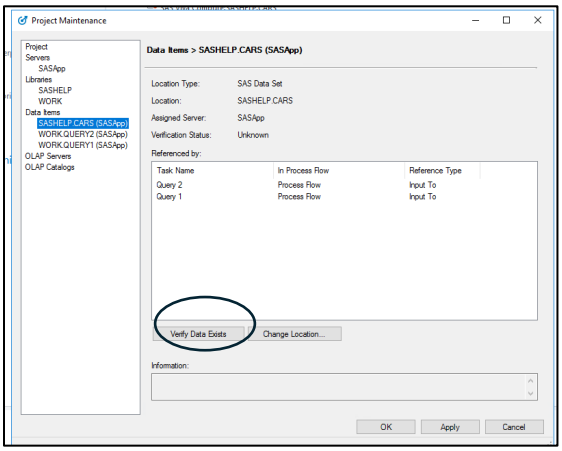


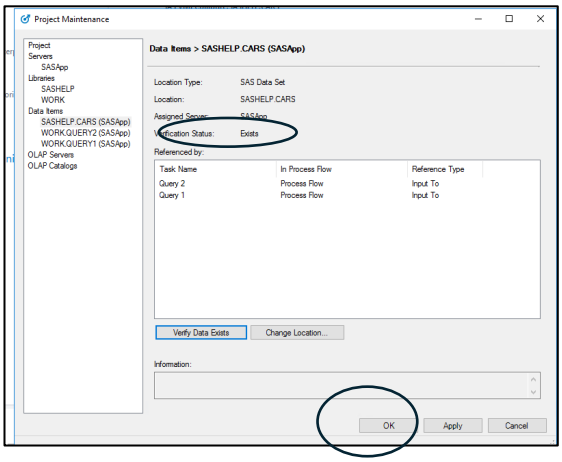
Libraries and data items have been mapped to objects with the same name on the new server.

20. Select **Libraries**. SASHELP is now located on the **SAS Viya Compute** server. If your data items reside in libraries with different names, you can update the references by selecting Replace in Project and navigating to the appropriate library on the SAS Viya Compute server.



21. You can also use Project Maintenance to verify that the data exists on the SAS Viya Compute server. Select **Data Items** and click **Verify Data Exists** for the SASHELP.CARS table. The Verification Status will change to Exists if the data is available on the server. If the data did not exist, you can change the location using Project Maintenance.





22. Click **OK** and select **Yes** to save your changes.

23. The updated project automatically opens. Select **Run**. Use the submission status window to confirm that all operational nodes are now submitted on the SAS Viya Compute server.

24. Use the application toolbar in SAS Enterprise Guide and select **File** and **Close project**. You do **not** need to save the project.

The Project Maintenance tool updates a single project while the SAS Enterprise Guide Migration Wizard offers similar functionality but for multiple projects. The migration of older SAS Enterprise Guide files to later versions is also possible using the Migration Wizard.

Demo 3: Leveraging SAS Viya Capabilities within a SAS Enterprise Guide Project

Connecting to SAS Viya through SAS Enterprise Guide enables code to be executed on a SAS Viya server. This optimizes performance and resources while promoting collaboration.

Benefits include, parallel data processing and advanced analytics via the SAS Cloud Analytics Services, often referred to as CAS, as well as enhanced collaboration through the integration of open-source languages.

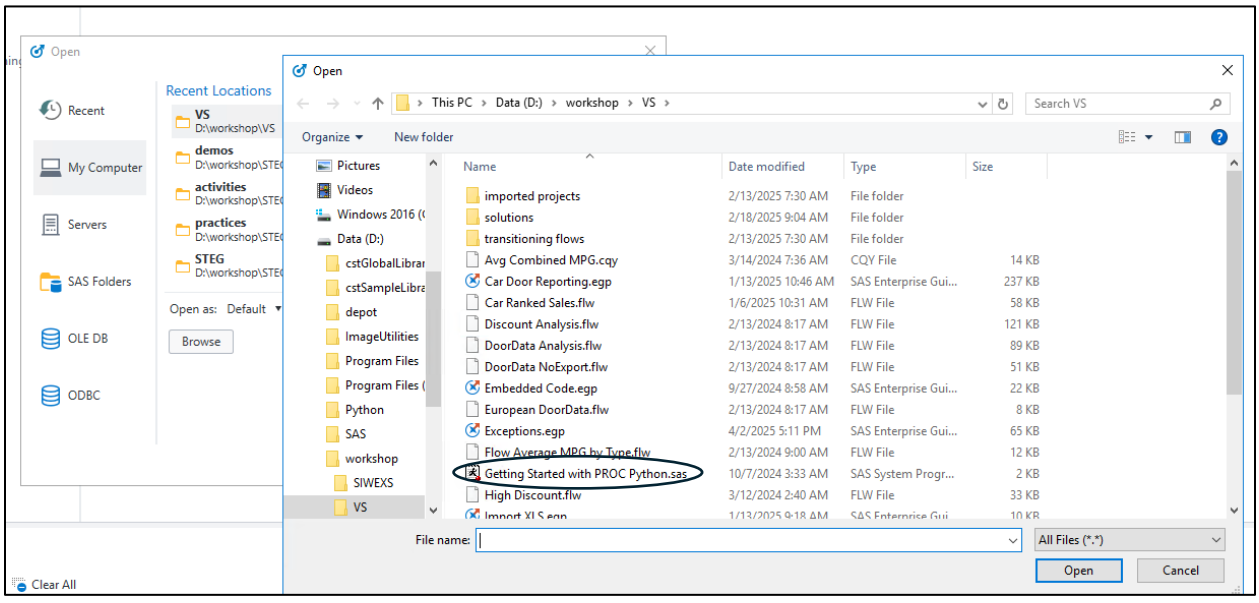
This demonstration highlights how easy it is to execute Python code in SAS Enterprise Guide and upload the data to the CAS server, for facilitating collaboration with other SAS Viya applications.

We will begin by using Python with Pandas to download and prepare the data, upload the DataFrame to the CAS server in SAS Viya using SAS code, and then leverage SAS Visual Analytics for visualization.

Note: Before proceeding with this demonstration, ensure the SAS Viya Compute server is available in SAS Enterprise Guide.

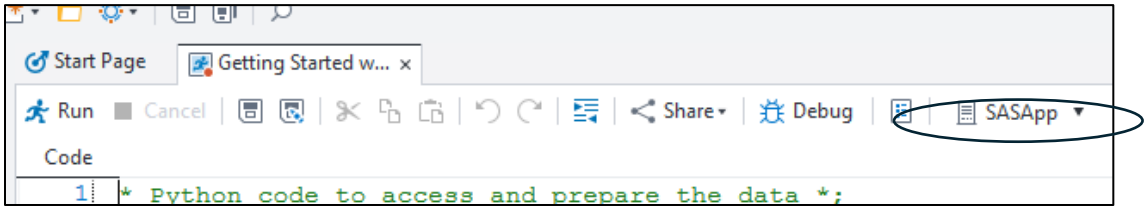
SECTION 1 : EXECUTING PYTHON CODE

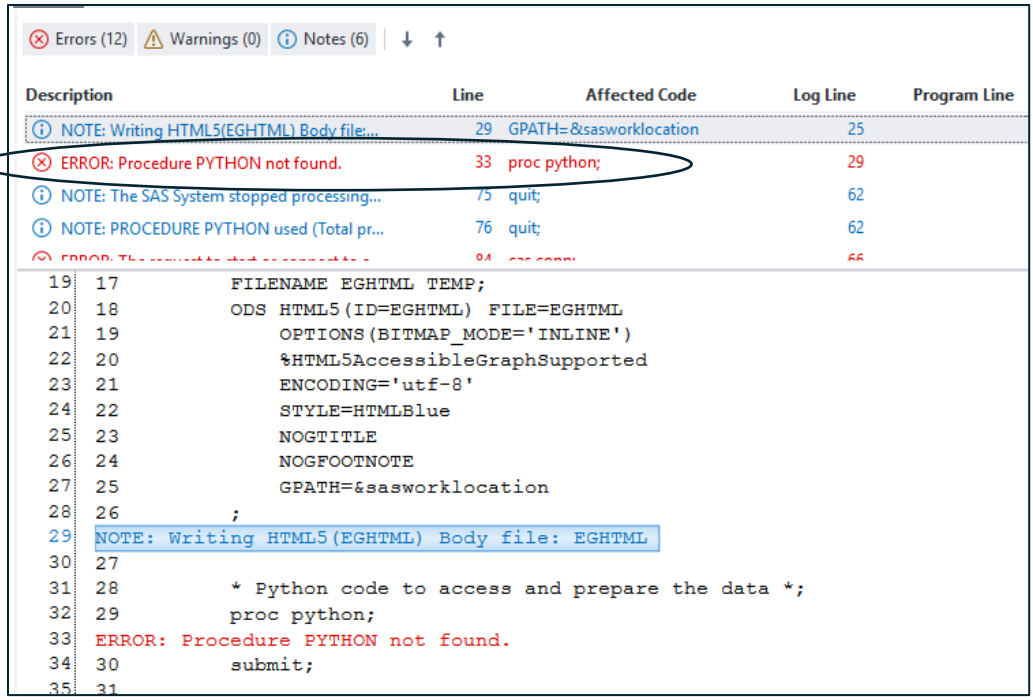
1. Use the application toolbar in SAS Enterprise Guide and select **File** and **Open**. Select **My Computer**, click **Browse** and navigate to **Data (D:)**, **workshop**, **VS**. Select **Getting Started with PROC Python.sas** and click **Open**.



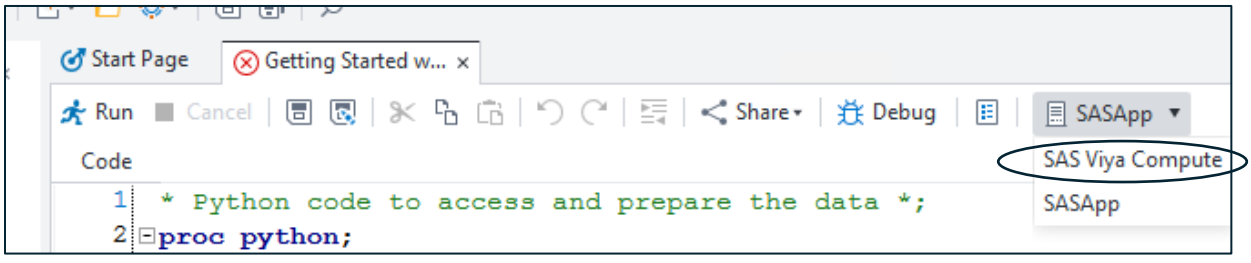
2. This SAS Program uses PROC PYTHON to execute Python code, followed by SAS code to load the data to an in-memory table accessible by the CAS server.

3. The SAS Program toolbar indicates the default environment of SASApp has been selected. Out of curiosity, click **Run** on the SAS Program toolbar. The program returns an error message of Procedure PYTHON not found.





4. Use the drop-down menu on the SAS Program toolbar and change the execution environment from **SASApp** to **SAS Viya Compute**.



Let's review and execute each section of code individually to gain a deeper understanding of its functionality.

5. In line 2, the PROC PYTHON statement begins the step, followed by the SUBMIT statement.

6. The Python code is included below the SUBMIT statement. The program imports the pandas and numpy packages, then sets a DataFrame column display option.

7. Under ##Access Data, in line 11 and 12, the read_csv pandas method reads the home_equity.csv file from the SAS Support website and creates a DataFrame.

8. In the ##Prepare Data section, from line 15 to line 23, pandas is used to prepare the DataFrame by replacing missing numeric values in the numeric columns with the mean of their respective columns, replacing character values in the character columns with the mode of their respective columns, creating two new columns, and then renaming columns by converting column names to lowercase and removing the underscore.

9. The code in line 27 uses the head method to print the first five rows of the DataFrame. A count of the number of missing values in each column is produced with the isna and sum methods as we can see from the code in line 28.

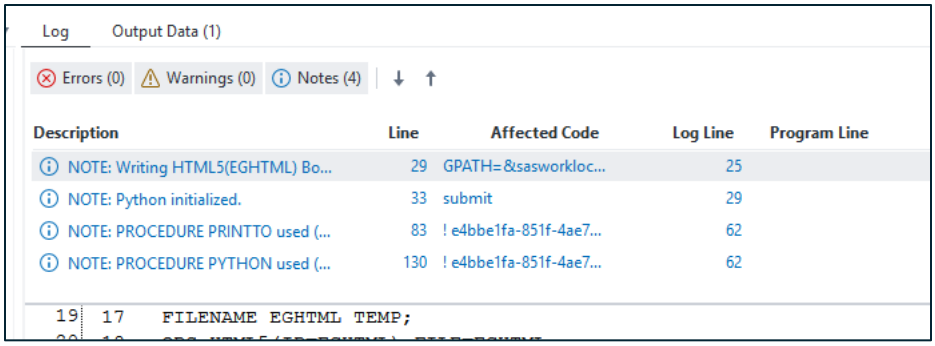
   This line of code is used to fill missing values (NaNs) in a pandas DataFrame with the most frequently occurring (mode) values for categorical columns.

10. To process the DataFrame in either the Compute Server or the CAS server in SAS Viya, it must be converted to a SAS data table. The SAS.df2sd method in line 32, transfers data from the specified DataFrame, df, to the SAS data table **home_equity_compute_sas** in the **WORK** library.

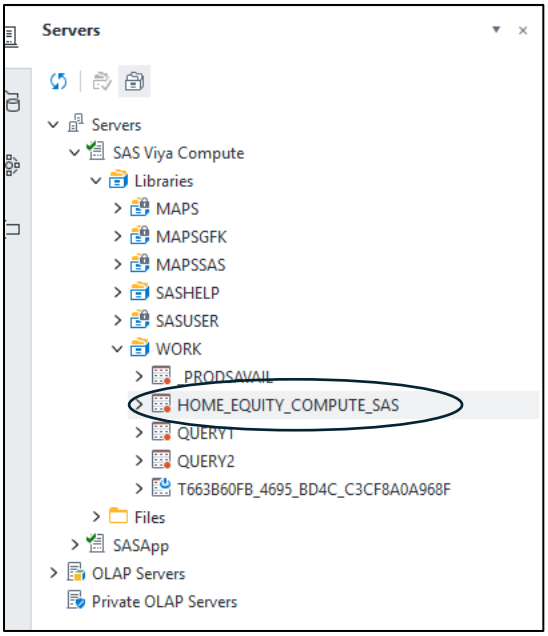
11. The **PYTHON** procedure ends with the ENDSUMBMIT and QUIT statements in line 34 and 35.

12. Highlight the code from line 2, PROC PYTHON, to line 35, the QUIT statement, and select RUN on the SAS Program toolbar.

13. Select the **Log** tab and confirm there are no warnings or errors. The log also displays the results from the Python code, for example, the first five rows of the DataFrame were printed using the head method and confirmation that missing values no longer exist in the columns.



14. To view the SAS data table on the SAS Viya Compute server, use the **Servers** pane and expand **SAS Viya Compute**, **Libraries** and **WORK**. Confirm that the **home_equity_compute_sas** data table was created successfully.



SECTION 2 : LOADING DATA INTO CAS(CLOUD ANALYTICS SERVER)

Traditional SAS code can now be used to further process the data. We will load it into an in-memory table on the CAS server, making it accessible for visualization in SAS Visual Analytics.

Again, let's review the code before executing it.

15. In line 39, the CAS statement is used to create a CAS session named **conn**.

16. To access and manipulate data that is stored in the CAS session from SAS Enterprise Guide, we can use the LIBNAME statement to map a SAS library to the CASUSER caslib. This is not required for uploading the data, but serves as a useful verification method to confirm that the upload is successful.

17. Next, the **CASUTIL** procedure is used to manage and transfer data between the SAS Compute Server and the CAS server.

    a. The **DROPTABLE** statement drops the table if it already exists on the CAS server.

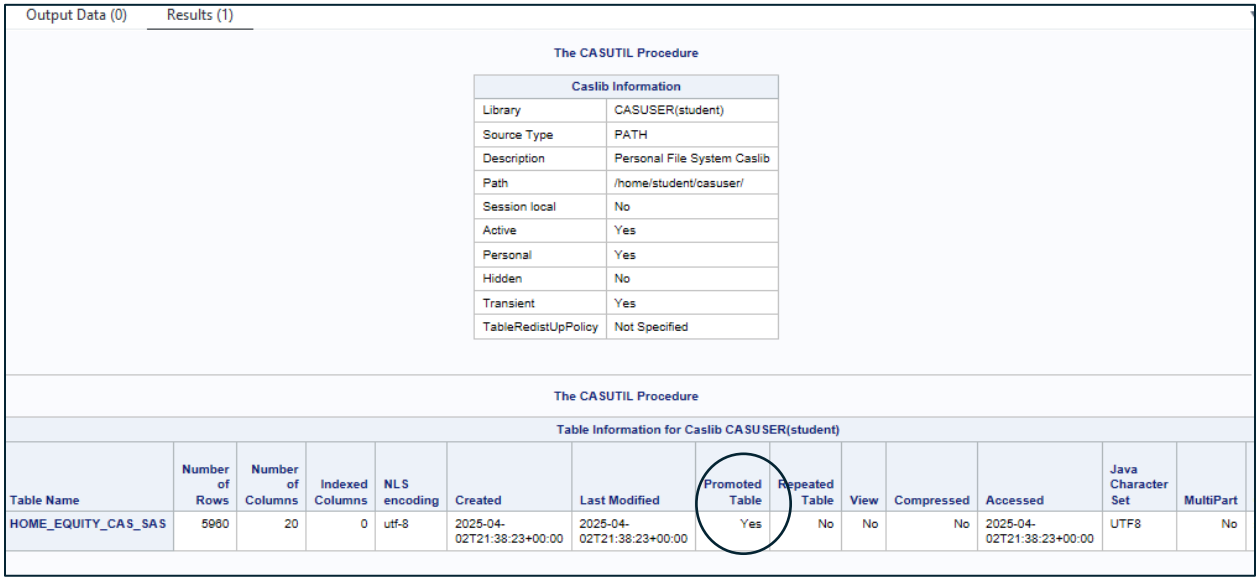
    b. The LOAD statement reads the SAS table **home_equity_compute_sas** and creates an in-memory CAS table named **home_equity_cas_sas**, within the **CASUSER** caslib on the CAS server.

    c. The **PROMOTE** option on the **LOAD** statement specifies the table should have global scope, which makes the table available to all sessions that use the CASUSER caslib, subject to access controls.

    d. The **LIST TABLE** statement lists all available CAS tables so you can confirm the table was loaded.

18. Highlight and run the code from line 39 to line 54.

19. In the Results tab, view the report and confirm that the CAS table has been successfully promoted to ensure its availability in SAS Viya and it's persistence beyond the current session.



Output Data (0) Results (1)

The CASUTIL Procedure

Caslib Information	
Library	CASUSER(student)
Source Type	PATH
Description	Personal File System Caslib
Path	/home/student/casuser/
Session local	No
Active	Yes
Personal	Yes
Hidden	No
Transient	Yes
TableRedistUpPolicy	Not Specified

The CASUTIL Procedure

Table Information for Caslib CASUSER(student)

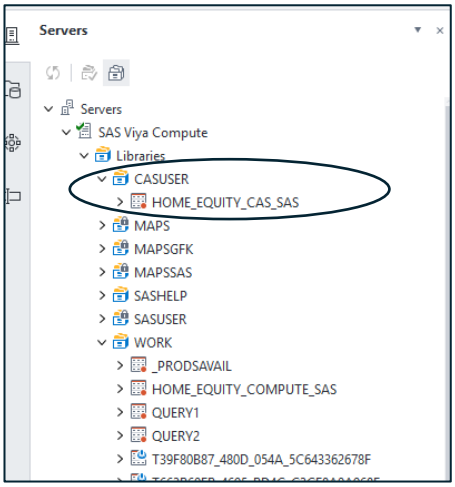
Table Name	Number of Rows	Number of Columns	Indexed Columns	NLS encoding	Created	Last Modified	Promoted Table	Repeated Table	View	Compressed	Accessed	Java Character Set	MultiPart
HOME_EQUITY_CAS_SAS	5960	20	0	utf-8	2025-04-02T21:38:23+00:00	2025-04-02T21:38:23+00:00	Yes	No	No	No	2025-04-02T21:38:23+00:00	UTF8	No

20. Use the **Log** to confirm that the libref CASUSER was successfully assigned.

```
30
NOTE: Libref CASUSER was successfully assigned as follows:
      Engine:        CAS
      Physical Name: 5fc537de-35a6-8444-aa18-b063ff97e997
32
```

21. In the Server section, right-click the **SAS Viya Compute** server and select **Refresh**.

22. Expand **CASUSER** to view the in-memory table, **home_equity_cas_sas.**



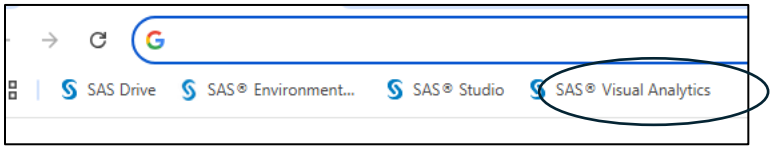
23. Finally, highlight and run the CAS TERMINATE statement in line 57 to end the session named **conn**. The Log confirms that the Libref CASUSER has been deassigned.

```
28   cas conn terminate;
NOTE: Libref CASUSER has been deassigned.
NOTE: Deletion of the session CONN was successful.
NOTE: The default CAS session CONN identified by SAS option SESSREF= was terminated. Use the OPTIONS statement
      option to an active session.
NOTE: Request to TERMINATE completed for session CONN.
29
30
```

SECTION 3 : EXPLORE AND VISUALIZE WITH SAS VISUAL ANALYTICS

Now that the CAS tables are loaded into memory with global scope, other SAS Viya applications can be used to manipulate, visualize or perform advanced analytics on the data.

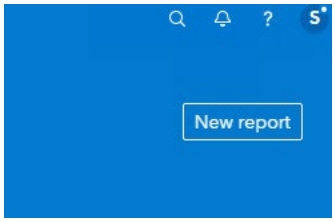
24. Open the Chrome Browser & click on the SAS Visual Analytics icon to launch.

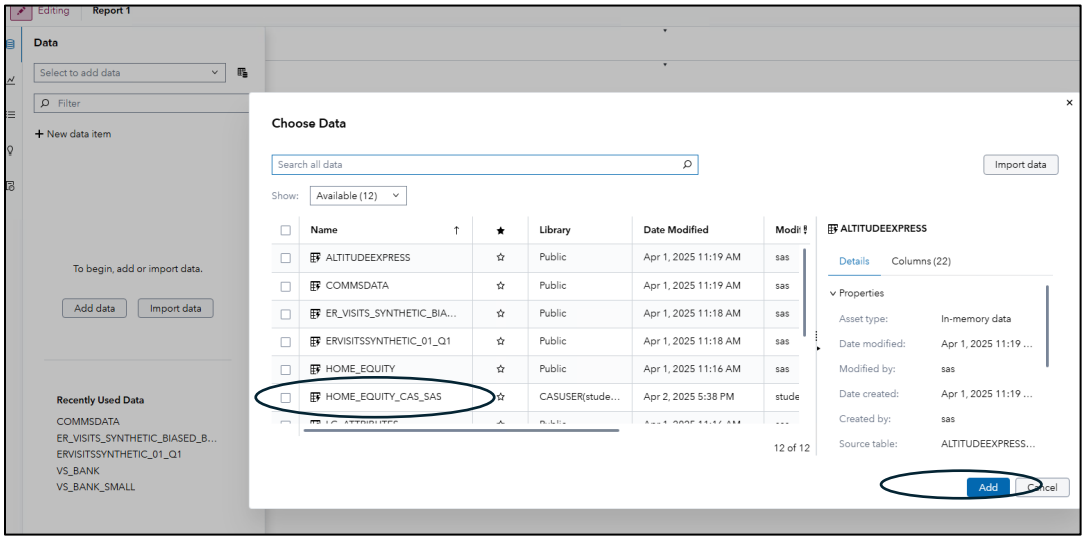


SAS Visual Analytics can be used to visualize and discover relevant relationships in your data. You can create and share interactive reports and dashboards, and use self-service analytics to quickly assess probable outcomes for smarter, more data-driven decisions.

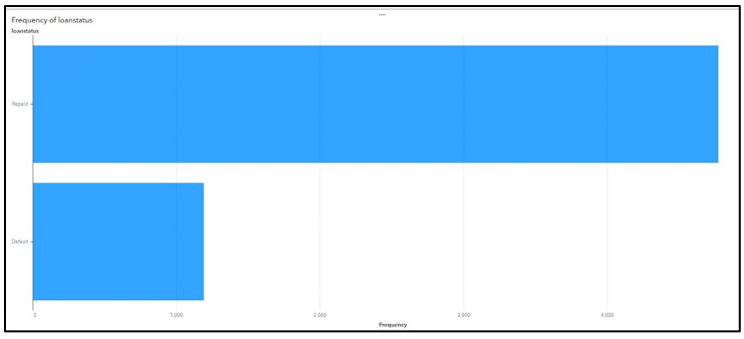
We can use SAS Visual Analytics to create a bar chart to display the frequency of repaid loans and use the automated analysis feature to gain deeper insights from the data.

25. To begin creating a report, select **New report** and click **Add data**. The Available tab shows all available CAS tables in the environment. Select **HOME_EQUITY_CAS_SAS** and click **Add**.

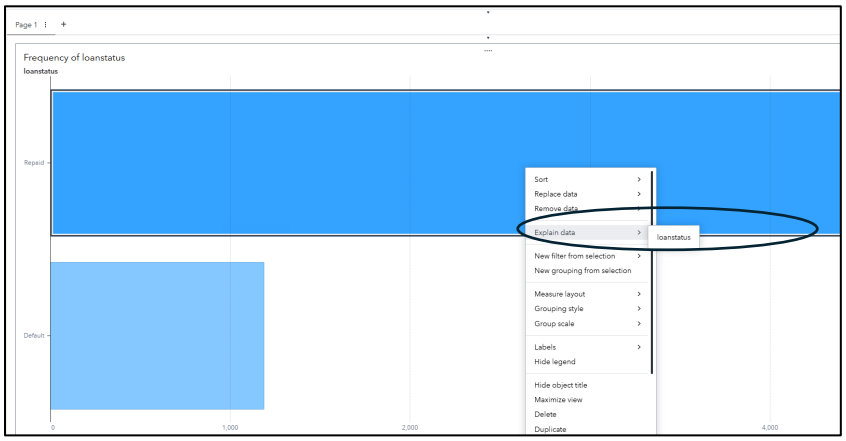


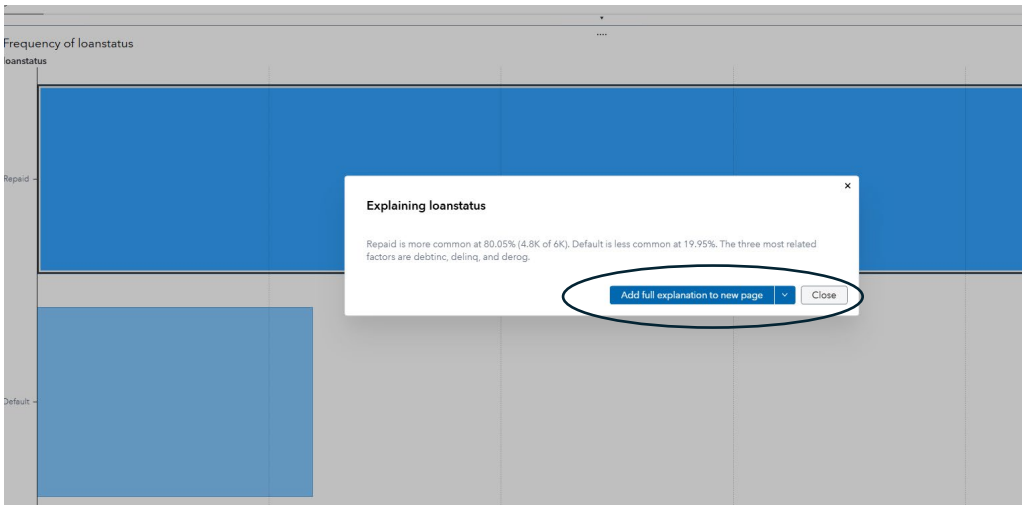


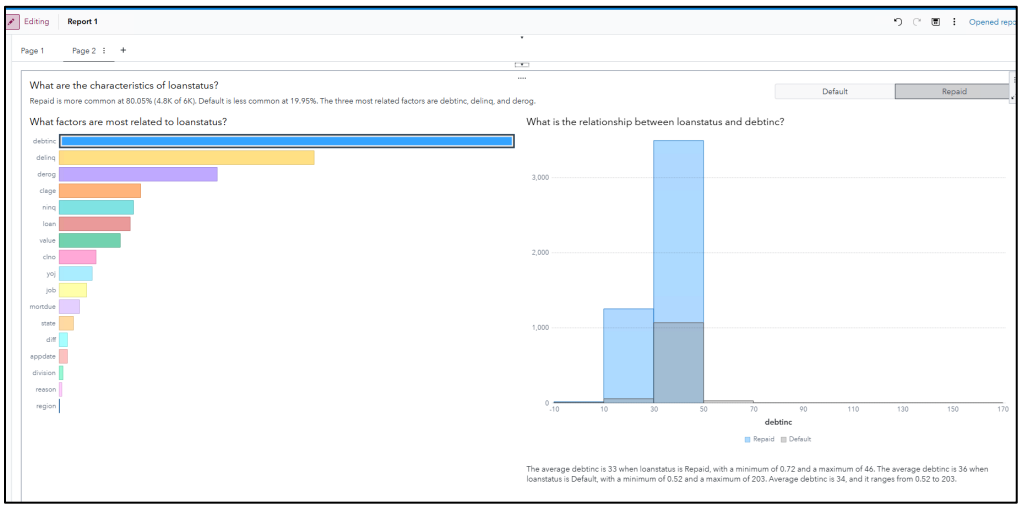
26. With data selected in SAS Visual Analytics, you can now easily visualize and describe the in-memory data. For example, to create a bar chart to see the frequency for loan status, simply drag the **loanstatus** item to the canvas. The graph indicates that the majority of loans have been repaid.



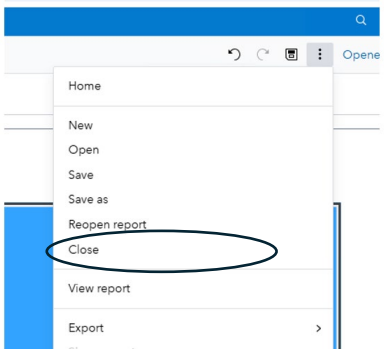
27. Right-click the bar chart and select **Explain Data** and **loanstatus**. Click **Add full explanation to new page** and review the added contents. This automated analysis identifies key factors influencing the data and uncovers potential relationships between variables, all without the need for coding.







28. Click ⋮ (**More**) on the report toolbar and select **Close** to close the report. You do not need to save the report.



We have only briefly covered some of the amazing features of SAS Visual Analytics. For a more comprehensive exploration, please see the SAS Visual Analytics 1 for SAS Viya: Basics course.

In summary, integrating SAS Enterprise Guide with SAS Viya, enables you to continue using your Windows desktop client while providing access to a wide range of SAS Viya applications, open source tools and many other benefits.