

Modern Data Science with SAS Viya Workbench: Unified Development with SAS, Python, and R

Shelby Taylor, SAS Institute

ABSTRACT

SAS® Viya® Workbench is a modern, code-first analytics environment that brings SAS®, R, and Python together within the SAS Viya ecosystem. It supports both learning and professional analytics workflows through flexible interfaces and integrated development tools.

Getting started involves launching sessions, selecting compute resources, and navigating a project-based file system. Core functionality includes executing SAS programs, reviewing logs and results, and managing output files, with SAS notebooks available for interactive and exploratory analysis.

The environment also enables development in Python and R within the same workspace. Collaboration and reproducibility are supported through integration with GitHub® for version control, code review, and traceable workflows, along with connectivity to external data sources such as Snowflake® and Google® Cloud Storage.

INTRODUCTION

SAS Viya Workbench is a lightweight, on-demand cloud development environment designed specifically for programmers and modelers. Users can choose the amount of compute necessary for their project and their preferred coding language. You can code in your language of choice, including SAS, Python, and R, and use your interface of choice, such as Visual Studio Code®, SAS Enterprise Guide®, JupyterLab or Jupyter Notebook.

One of the main benefits of SAS Viya Workbench is its centralized development for different users. SAS and open-source developers can both code in the same environment and collaborate using GitHub for distributed version control. With SAS Viya Workbench, users also have platform optionality, meaning they can tailor their environment to their specific needs.

Workbench can enhance performance through SAS's best-in-class analytics. Open-source programmers can achieve faster results by leveraging SAS analytics through APIs for Python and R with nearly identical syntax.

Users can deploy multiple Workbenches to accommodate a wide range of compute tasks. And these Workbenches can then be connected to the enterprise SAS Viya environment to share, govern, and deploy assets. By connecting to the enterprise analytics platform, content created in Workbench can be serviced and managed across the organization. However, since SAS Viya Workbench is licensed separately, it's also a great way to experiment with Viya analytics before investing fully into Viya.

START AND OPEN A SAS VIYA WORKBENCH

You begin by logging in to SAS Viya Workbench and navigating to the Home page. The left navigation panel provides access to available workbenches, storage, and the activity log.

From the Home page, you create a new on-demand computing environment by selecting **New Workbench**. When prompted, you assign a name to the environment, such as *MyWorkbench*. You then select the compute resources available for the session, including the number of cores to allocate. The available options depend on the system configuration and the permissions defined by the site administrator. In some environments, additional resources such as GPU-enabled compute might also be available.

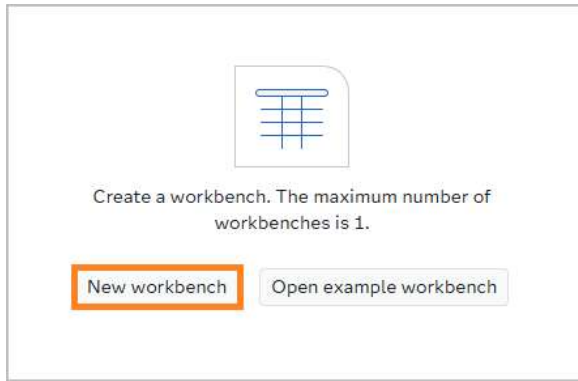


Figure 1: Creating a New Workbench

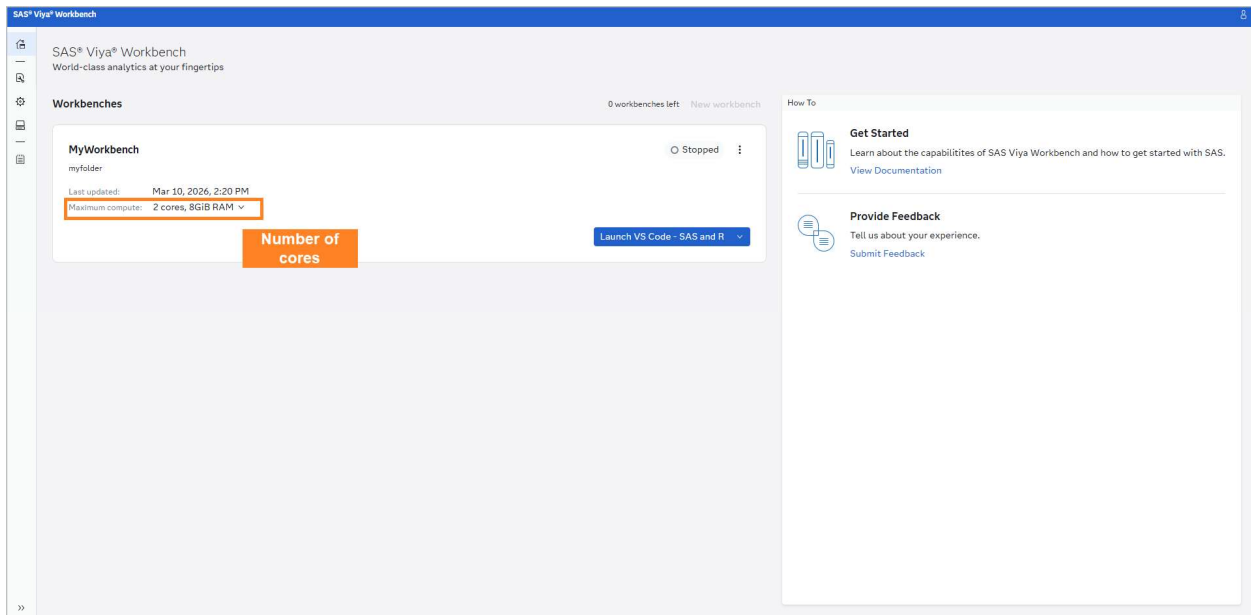


Figure 2: SAS Viya Workbench Home Page

Once the workbench is created, you manage it through the Workbench menu by selecting **Settings**, where configuration details can be reviewed or modified. You can also create additional workbenches for different analytic projects, allowing you to maintain separate and organized development environments.

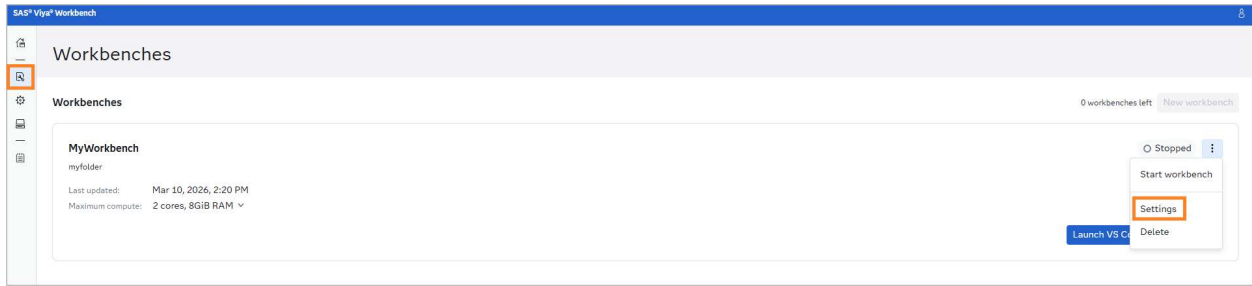


Figure 3: Workbench Settings

After defining the workbench, you start the compute environment. Within a few seconds, the workbench becomes active and prompts you to select a preferred programming interface. You can choose Visual Studio Code, JupyterLab, Jupyter Notebook, or Enterprise Guide. In this example, we select Visual Studio Code.

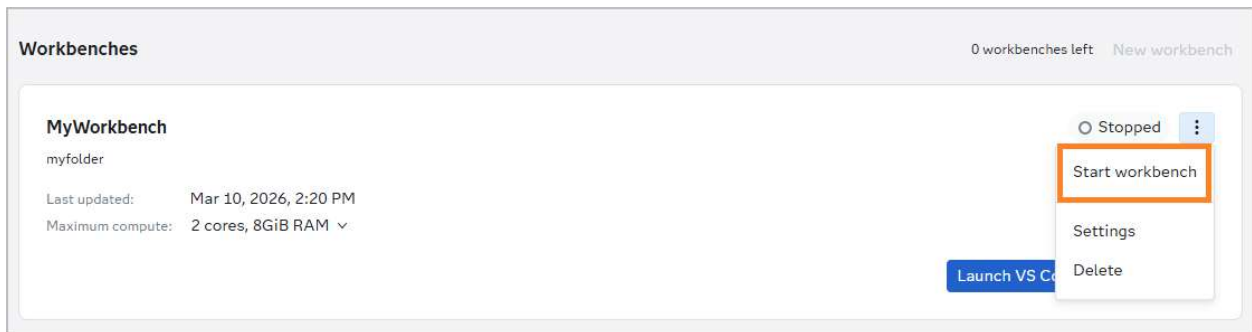


Figure 4: Start a Workbench

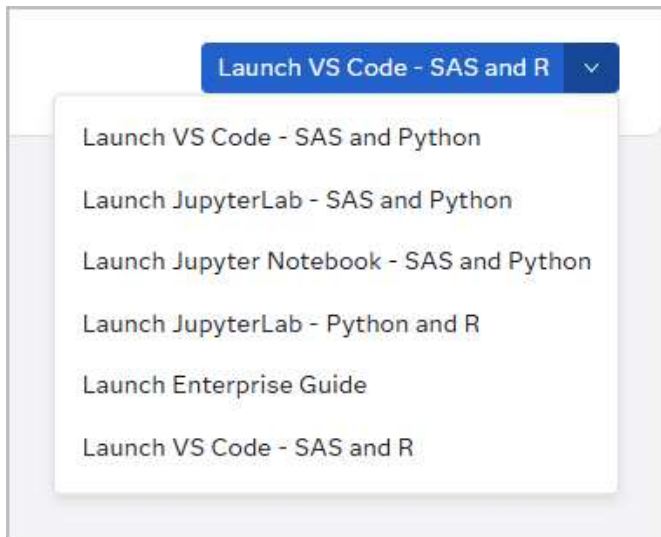


Figure 5: Select Programming Interface

When the Visual Studio Code interface opens, the navigation bar appears on the left side of the screen. In the Explorer panel, you access the default project folder, which serves as the primary location for saving workbench files and project assets.

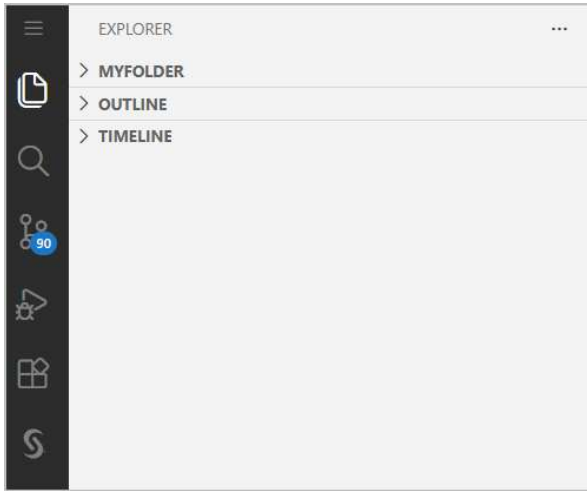


Figure 6: Explorer Tab

DEVELOP SAS CODE

In SAS Viya Workbench, you can create and run SAS programs directly in the cloud. For example, you can create a SAS program and run the code shown in Program 1.

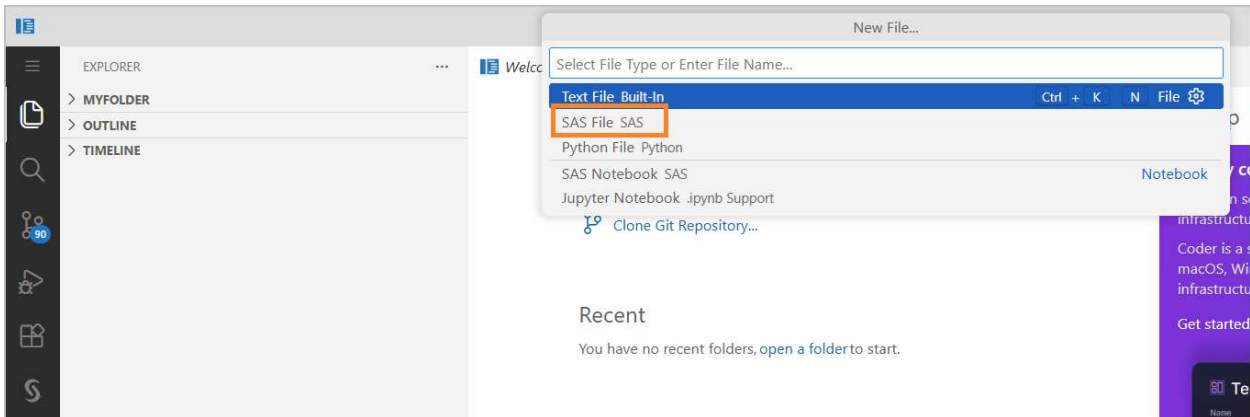


Figure 7: Creating a SAS Program

```

ods html5 style=illuminate;
ods select Variables;
proc contents data=sashelp.cars;
run;

proc print data=sashelp.cars(obs=5);
run;

proc freq data=sashelp.cars order=freq;
  tables Make Origin EngineSize / plots=freq;
run;

proc means data=sashelp.cars mean;
  class Make;
  var MPG_City MPG_Highway;
run;

```

Program 1: Example SAS Code

This code uses the CONTENTS procedure to display the SASHELP.CARS table metadata, PRINT to preview the first few rows of the table, FREQ to show frequency distributions of certain variables, and MEANS to compute summary statistics. Running the program executes the code and returns results alongside the SAS log.

The screenshot shows the SAS Studio interface. On the left, the SAS program code is displayed in a text editor. On the right, the results of the CONTENTS procedure are shown in a table format. The table is titled "The CONTENTS Procedure" and "Alphabetic List of Variables and Attributes".

#	Variable	Type	Len	Format	Label
9	Cylinders	Num	8		
5	DriveTrain	Char	5		
8	EngineSize	Num	8		Engine Size (L)
10	Horsepower	Num	8		
7	Invoice	Num	8	DOLLAR8.	
15	Length	Num	8		Length (IN)
11	MPG_City	Num	8		MPG (City)
12	MPG_Highway	Num	8		MPG (Highway)
6	MSRP	Num	8	DOLLAR8.	
1	Make	Char	13		
2	Model	Char	40		
4	Origin	Char	6		

Below the table, the SAS log is visible, showing the execution of the program and the results of the MEANS procedure. The log includes the following text:

```

20 var MPG_City MPG_Highway;
21 run;

NOTE: There were 428 observations read from the data set SASHELP.CARS.
NOTE: PROCEDURE MEANS used (Total process time):
      real time      0.19 seconds
      cpu time       0.18 seconds

22
23 ;*';**;/;run;quit;ods html5(id=vscode) close;
24

```

Figure 8: SAS Program, Results, and Log

You can also view libraries and tables through the Visual Studio Code interface. Expanding a library such as SASHELP lets you preview tables, and you can also upload local data sets, Python, SAS, and R scripts to the cloud environment.

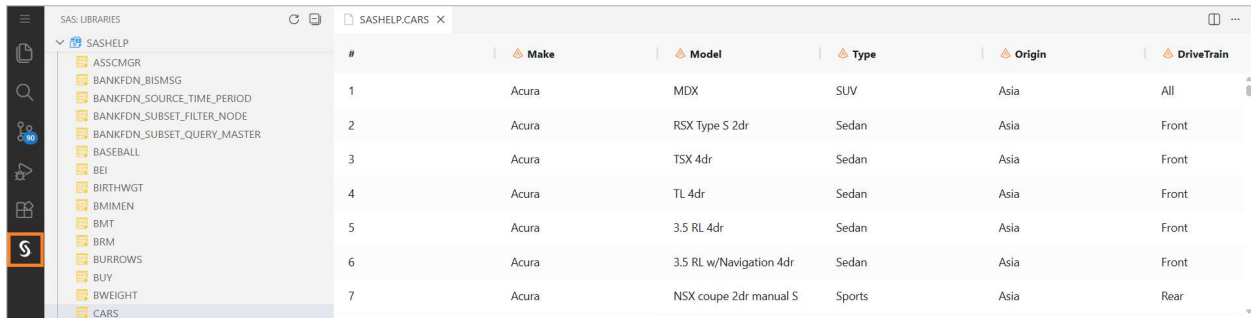


Figure 9: View Libraries and Preview Tables

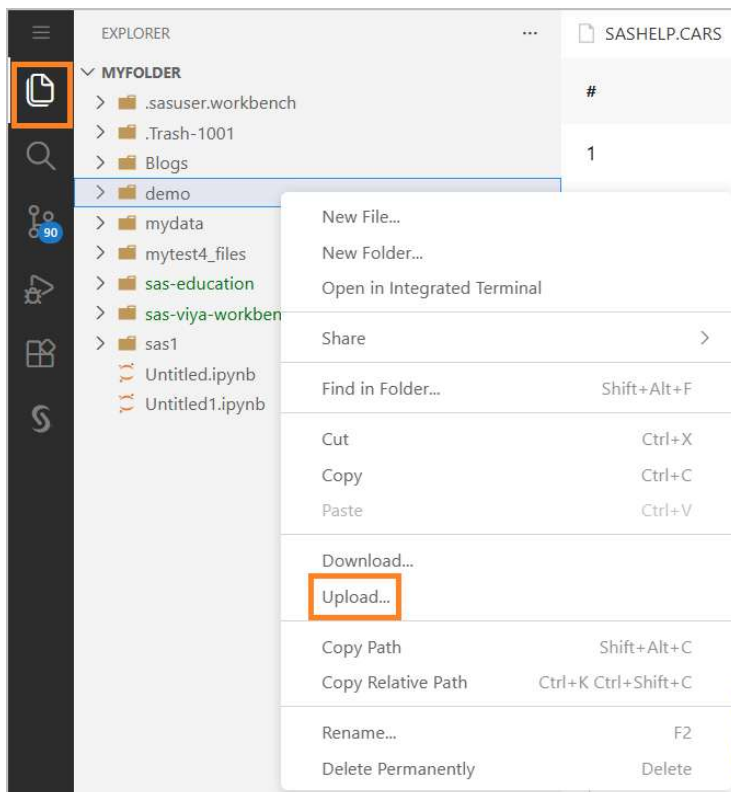


Figure 10: Upload Local Files to Workbench

In addition, SAS Viya Workbench supports SAS notebooks for interactive analysis. Notebooks combine markdown and executable code cells. You can run SAS, SQL, and Python code in separate cells and immediately view outputs. This interactive approach allows you to document and analyze data within the same environment.



Figure 11: Example SAS Notebook

You can also configure SAS auto-execution files to automatically run predefined code when a SAS session begins. These files are commonly used to establish environment settings, define library references, or load frequently used macros. By centralizing these setup steps in an auto-execution file, you ensure that the same configuration is applied each time the environment starts, which promotes consistency and reduces repetitive setup tasks.

DEVELOP PYTHON AND R CODE

What if you want to use Python or R? SAS Viya Workbench offers a secure, governed cloud environment that supports Python and R. You can leverage our pre-installed Python and R environments seamlessly or install your own packages.

The following Python program imports the pandas package, reads a CSV data set from the SAS Viya Examples Data Sets website, and performs basic exploration of the resulting data frame. The program inspects the data types of each column, displays the first few rows, and generates descriptive statistics, providing a preliminary understanding of the dataset's structure and contents.

```

import pandas as pd

df =
pd.read_csv('https://support.sas.com/documentation/onlinedoc/viya/exampledatasets/cars.csv')

print(df.dtypes)

print(df.head())

print(df.describe())

```

Program 2: Example Python Code

```

import pandas as pd.py X
demo > import pandas as pd.py > ...
1 import pandas as pd
2
3 df = pd.read_csv('https://support.sas.com/documentation/onlinedoc/viya/EXAMPLEDATASETS/cars.csv')
4
5 print(df.dtypes)
6
7 print(df.head())
8
9 print(df.describe())

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

/workspaces/myfolder
> /bin/python3.11 "/workspaces/myfolder/demo/import pandas as pd.py"
Make          object
Model         object
Type          object
Origin        object
DriveTrain    object
MSRP          float64
Invoice       float64
EngineSize    float64
Cylinders     float64
Horsepower    float64
MPG_City      float64
MPG_Highway   float64
Weight        float64
Wheelbase     float64
Length        float64
dtype: object

```

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepower	MPG_City	MPG_Highway	Weight	Wheelbase	Length
0	Acura	MDX	SUV	Asia	All	36945.0	33337.0	3.5	6.0	265.0	17.0	23.0	4451.0	186.0	189.0
1	Acura	RSX Type S	2dr Sedan	Asia	Front	23820.0	21761.0	2.0	4.0	200.0	24.0	31.0	2778.0	161.0	172.0
2	Acura	TSX 4dr	Sedan	Asia	Front	26990.0	24647.0	2.4	4.0	200.0	22.0	29.0	3230.0	195.0	183.0
3	Acura	TL 4dr	Sedan	Asia	Front	33195.0	30299.0	3.2	6.0	270.0	20.0	28.0	3575.0	188.0	186.0
4	Acura	3.5 RL 4dr	Sedan	Asia	Front	43755.0	39014.0	3.5	6.0	225.0	18.0	24.0	3880.0	115.0	197.0

```

count      428.000000      428.000000      428.000000      426.000000      428.000000      428.000000      428.000000      428.000000      428.000000      428.000000      428.000000      428.000000      428.000000      428.000000      428.000000
mean      32774.851140      30014.700935      3.196729      5.807512      215.885514      20.060748      26.843458      3577.953271      108.154206      186.362150
std       19431.716674      17642.117750      1.108595      1.558443      71.836032      5.238218      5.741201      758.983215      8.311813      14.357991
min       10280.000000      9875.000000      1.300000      3.000000      73.000000      10.000000      12.000000      1850.000000      89.000000      143.000000

```

Figure 12: Python Code Results

The following R program performs the same actions as the python program, this time loading the readr package to access the read_csv() function.

```

library(readr)

df <-
read_csv("https://support.sas.com/documentation/onlinedoc/viya/EXAMPLEDATASETS/cars.csv")

str(df)

head(df)

summary(df)

```

Program 3: Example R Code

```
demo > R rdemor > ...
1 library(readr)
2
3 df <- read_csv("https://support.sas.com/documentation/onlinedoc/viya/empldatssets/cars.csv")
4 str(df)
5 head(df)
6 summary(df)
7
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS 1

Mode :character Mode :character Mode :character Mode :character

DriveTrain	MSRP	Invoice	EngineSize
Length:428	Min. : 10280	Min. : 9875	Min. : 1.300
Class :character	1st Qu.: 20334	1st Qu.: 18866	1st Qu.: 2.375
Mode :character	Median : 27635	Median : 25294	Median : 3.000
	Mean : 32775	Mean : 30015	Mean : 3.197
	3rd Qu.: 39285	3rd Qu.: 35718	3rd Qu.: 3.900
	Max. : 192465	Max. : 173560	Max. : 8.300

Cylinders	Horsepower	MPG_City	MPG_Highway
Min. : 3.000	Min. : 73.0	Min. : 10.00	Min. : 12.00
1st Qu.: 4.000	1st Qu.: 105.0	1st Qu.: 17.00	1st Qu.: 24.00
Median : 6.000	Median : 210.0	Median : 19.00	Median : 26.00
Mean : 5.808	Mean : 215.9	Mean : 20.06	Mean : 26.84
3rd Qu.: 6.000	3rd Qu.: 255.0	3rd Qu.: 21.25	3rd Qu.: 29.00
Max. : 12.000	Max. : 500.0	Max. : 60.00	Max. : 66.00
NA's : 2			

Weight	Wheelbase	Length
Min. : 1850	Min. : 89.0	Min. : 143.0
1st Qu.: 3104	1st Qu.: 103.0	1st Qu.: 178.0
Median : 3474	Median : 107.0	Median : 187.0
Mean : 3578	Mean : 108.2	Mean : 186.4
3rd Qu.: 3978	3rd Qu.: 112.0	3rd Qu.: 194.0
Max. : 7190	Max. : 144.0	Max. : 238.0

Figure 13: R Code Results

You can also create Jupyter notebooks within Visual Studio Code to run either Python or R code.

The following R loads the readr package, reads the cars.csv file to create a data frame, and runs the head() function.

```
library(readr)

df <-
read_csv("https://support.sas.com/documentation/onlinedoc/viya/empldatssets/cars.csv")

head(df)
```

Figure 14: Example R Code for Jupyter Notebook

```

rdemo2.ipynb X
demo > rdemo2.ipynb > library(readr)
+ Code + Markdown | Run All Clear All Outputs Outline ...
Select Kernel

library(readr)
df <- read_csv("https://support.sas.com/documentation/onlinedoc/viya/exampldatasets/cars.csv")
head(df)

...
`curl` package not installed, falling back to using `url`
Rows: 428 Columns: 15
— Column specification —
Delimiter: ","
chr (5): Make, Model, Type, Origin, DriveTrain
dbl (10): MSRP, Invoice, EngineSize, Cylinders, Horsepower, MPG_City, MPG_Hi...
i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.

...
A tibble: 6 × 15
  Make      Model  Type  Origin DriveTrain  MSRP  Invoice  EngineSize  Cylinders  Horsepower  MPG_City  MPG_Highway  Weight  Wheelbase  Length
<chr>    <chr> <chr> <chr> <chr>    <dbl> <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
1 Acura    MDX     SUV   Asia    All      36945  33337    3.5       6         265        17        23         4451     106      189
2 Acura    RSX Type S 2dr Sedan Asia    Front  23820  21761    2.0       4         200        24        31         2778     101      172
3 Acura    TSX 4dr  Sedan Asia    Front  26990  24647    2.4       4         200        22        29         3230     105      183
4 Acura    TL 4dr   Sedan Asia    Front  33195  30299    3.2       6         270        20        28         3575     108      186
5 Acura    3.5 RL 4dr Sedan Asia    Front  43755  39014    3.5       6         225        18        24         3880     115      197
6 Acura    3.5 RL w/Navigation 4dr Sedan Asia    Front  46100  41100    3.5       6         225        18        24         3893     115      197

```

Figure 15: Jupyter Notebook Results

STOPPING A WORKBENCH

Returning to the SAS Viya Workbench Home page, you notice that the workbench is still running. You can leave the environment running, and it will auto-suspend after the default timeframe. Or you can click **Running > Stop Workbench** to shut down the environment and save cloud costs.

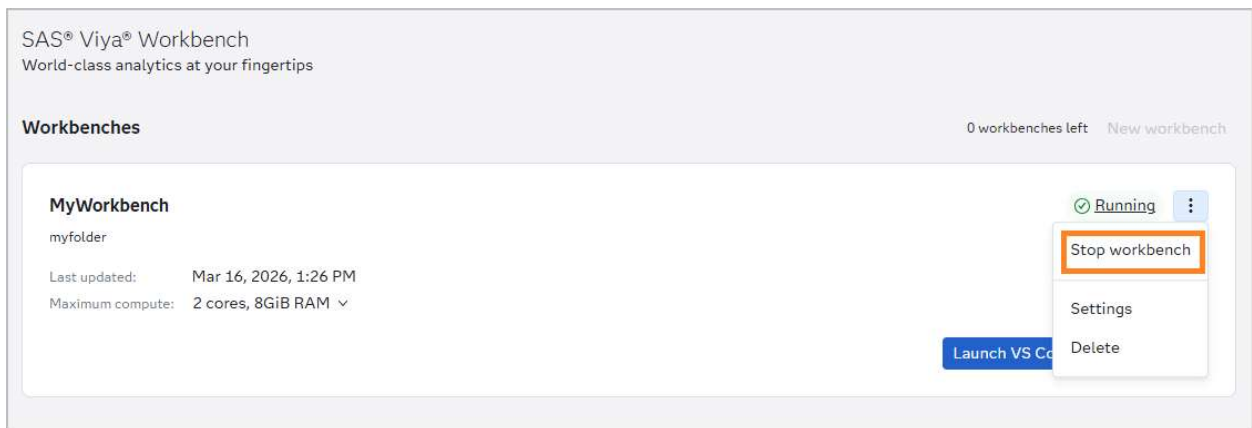


Figure 16: Stopping a Workbench

If you need more computing power, you can easily modify the compute power if permitted by your administrator. Note that you do need to stop the environment to change the compute settings. After the environment restarts, you can reopen your programming environment and continue where you left off.

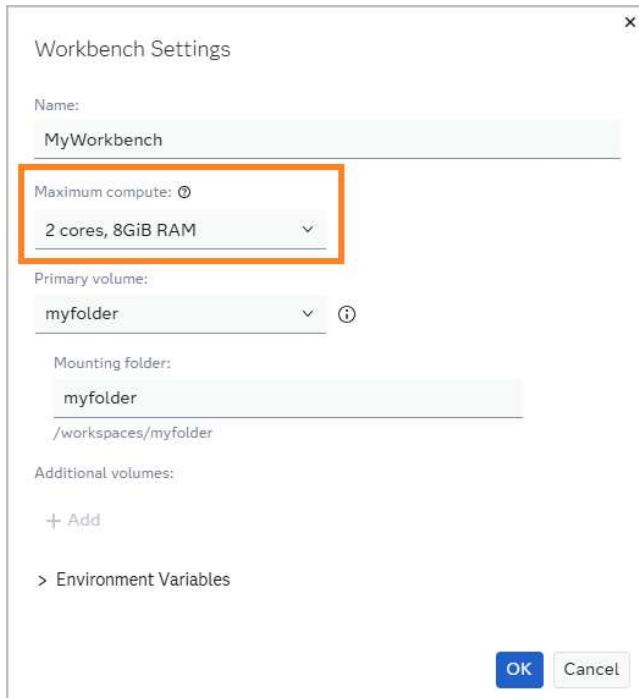


Figure 17: Changing Compute Resources

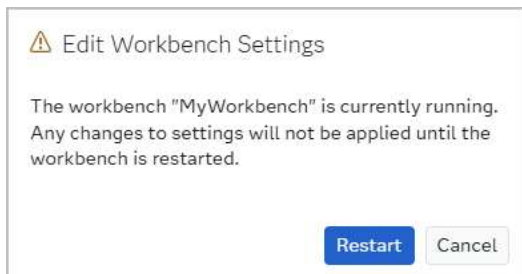


Figure 18: Restart Message When Editing Workbench Settings

ADDITIONAL FEATURES

SAS Viya Workbench also includes several features that support collaboration and integration with external data sources. These capabilities allow you to incorporate Workbench into modern development workflows and connect to enterprise data environments.

GITHUB INTEGRATION

Workbench can be connected to repositories hosted on GitHub, enabling you to manage code using version control. After connecting a repository, you can clone projects directly into the Workbench environment, track changes, and synchronize updates between local files and the remote repository. This integration supports collaborative development by allowing multiple users to contribute to the same project while maintaining a clear history of code revisions.



Figure 19: Source Control Tab

SAS AUTOEXEC FILES

Within SAS Viya workbench, you can create an autoexec file which contains SAS statements that are executed automatically when you open your programming interface. This feature is particularly useful in customizing your SAS environment. The autoexec file eliminates the need for you to manually execute repetitive code at startup, allowing them to tailor their environment to their preferences and specific requirements.

ACCESSING DATABASES

In addition to local files and example data sets, Workbench supports connections to external data platforms. You can read data directly from cloud-based storage systems and databases, including services such as Snowflake and Google Cloud Storage. These integrations enable you to access large datasets without manually downloading or transferring files, allowing analytics workflows to operate directly on enterprise-scale data stored in cloud environments.

BRIDGING SAS VIYA WORKBENCH AND SAS VIYA

By connecting SAS Viya Workbench to a separate SAS Viya environment, you can register models created in workbench into the SAS model manager repository for overall model governance. The repository is a safe place to securely store an official copy of models as you get them production-ready for deploying, tracking, and auditing.

CONCLUSION

SAS Viya Workbench provides a flexible cloud-based environment for developing and executing analytic workflows. Within a single interface, you can write and run SAS, Python, and R code, explore datasets, and perform analysis using both traditional programs and interactive notebooks. The platform also supports modern development practices through features such as version control integration.

In addition, Workbench enables direct access to external data sources and scalable compute resources, allowing analytic workflows to operate efficiently on cloud-based infrastructure. Together, these capabilities create an environment that supports both familiar SAS programming practices and modern, multi-language analytics in a unified workspace.

REFERENCES

SAS. (n.d.). *Getting started with SAS Viya Workbench* [Video]. YouTube.
<https://www.youtube.com/watch?v=BkjmQMXfpJU>

ACKNOWLEDGMENTS

Thank you to Peter Styliadis for the original creation of the SAS Viya Workbench overview, and Nicolas Robert and Peter Christie for their contributions.

ADDITIONAL RESOURCES

Modern Data Science with SAS® Viya® Workbench: Unified Development with SAS®, Python, and R.

- Free course available at <https://learn.sas.com/course/view.php?id=7274#section-7-tab>

Workbench for Learners: https://www.sas.com/en_us/software/viya-workbench-for-learners.html

- Free access to SAS Viya Workbench for students and educators.

CONTACT INFORMATION

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

Shelby Lynne Taylor
SAS Institute
shelby.taylor@sas.com