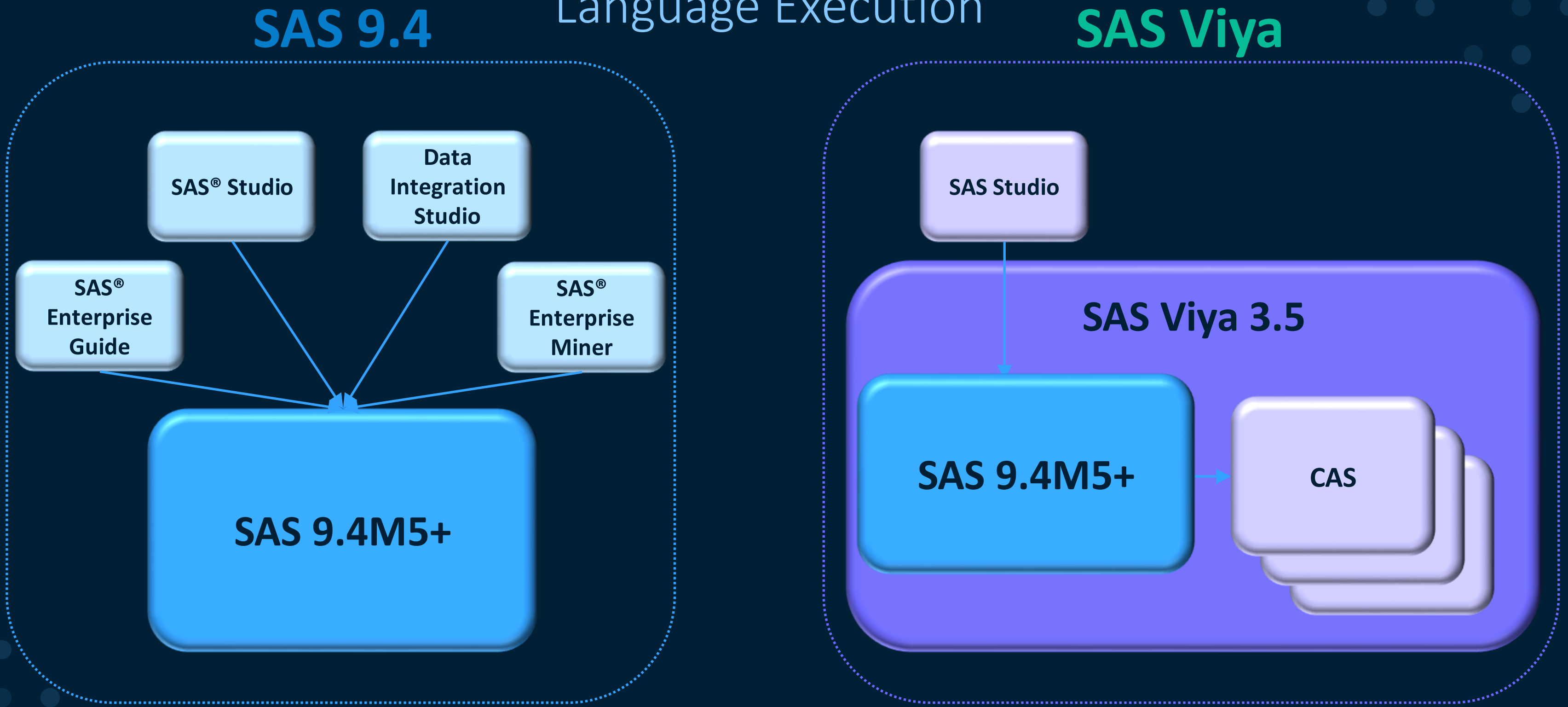
Viya Programming HOW

Hands On Workshop – Section #1

The SAS Platform

Language Execution



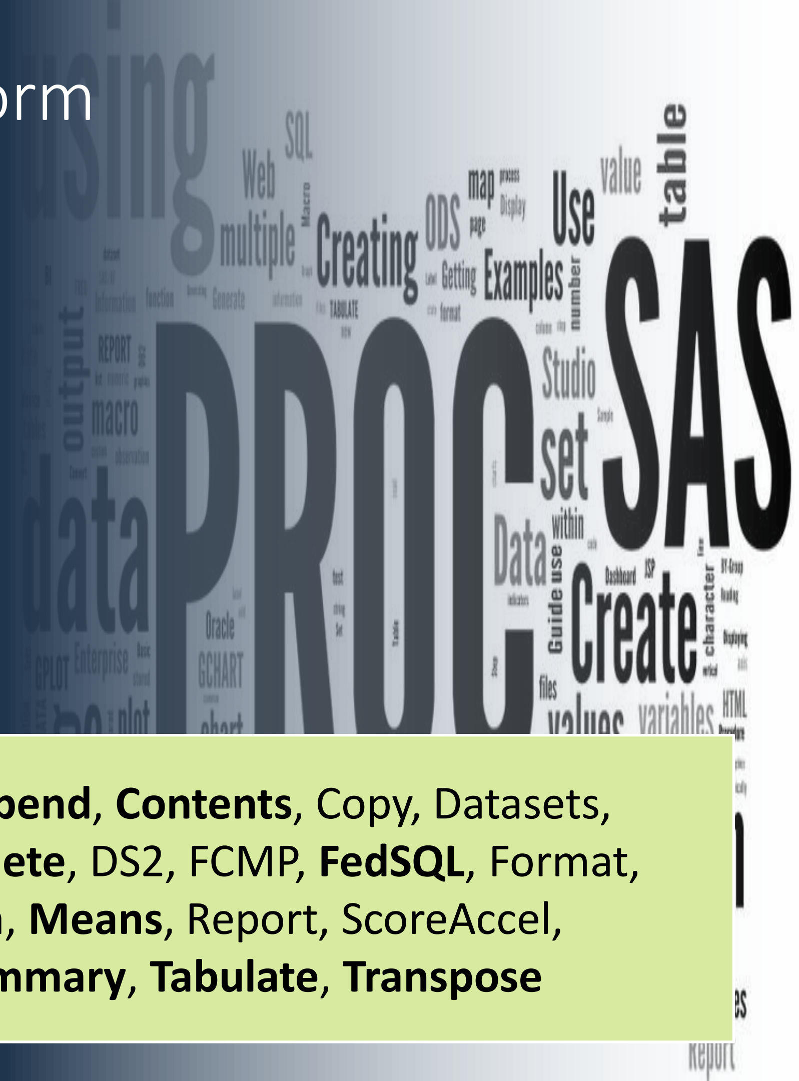
SAS Viya Procedures

The SAS Platform

Base SAS

- All 9.4 Base in Viya 3.3+
- SAS Viya = *speed!*
  - Multi-threaded DATA step
    - Rework code to leverage
    - Steven Sober's SGF Paper #1710-2018
      - SAS Viya Readiness Utility
  - Procedures That Use CAS Actions (16)

Append, **Contents**, Copy, Datasets, **Delete**, DS2, FCMP, **FedSQL**, Format, Lua, **Means**, Report, ScoreAccel, **Summary**, **Tabulate**, **Transpose**



SAS Viya Data Processing

General information

- Most analytics run in memory
  - Visual Statistics, Visual Forecasting, VDMML, Optimization, Econometrics, Visual Text Analytics
- SAS Foundation PROCs
  - CAS-enabled
  - **Not** CAS-enabled
- CAS Actions
  - PROC CAS (CASL)
  - Python
  - Lua…

[Procedures That Use CAS Actions]

[CAS Processing of Base SAS Procedures]

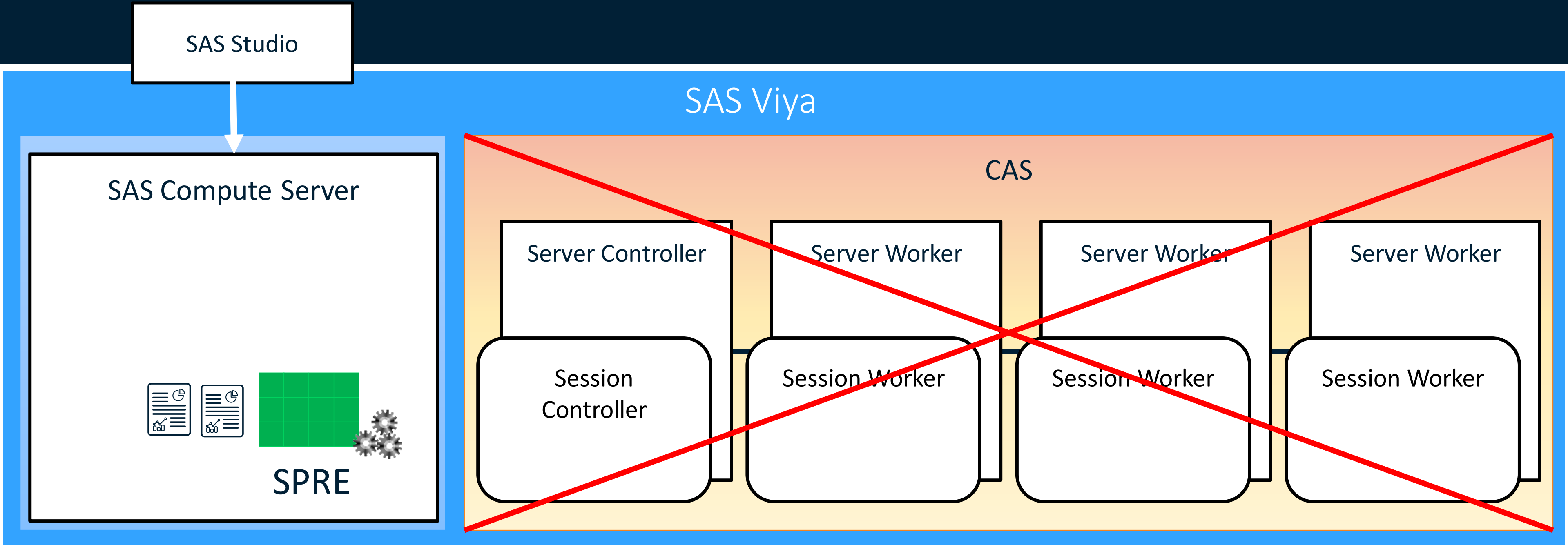
Append, Contents, Copy, Datasets, Delete, DS2, FCMP, FedSQL, Format, Lua, Means, Report, ScoreAccel, Summary, Tabulate, Transpose

Catalog, Compare, Download, DSTODS2, Export, FMTC2ITM, Hadoop, HDMD, HTTP, Import, JavaInfo, JSON, MapImport, Options, $\boxed{\text{Print}}$, PrintTo, Product_Status, PWEncode, Registry, S3, SGPanel, $\boxed{\text{SGPlot}}$, SGRender, SGScatter, $\boxed{\text{Sort}}$, $\boxed{\text{SQL}}$, Stream, Template

SAS Viya Data Processing

Local Data & Local Processing

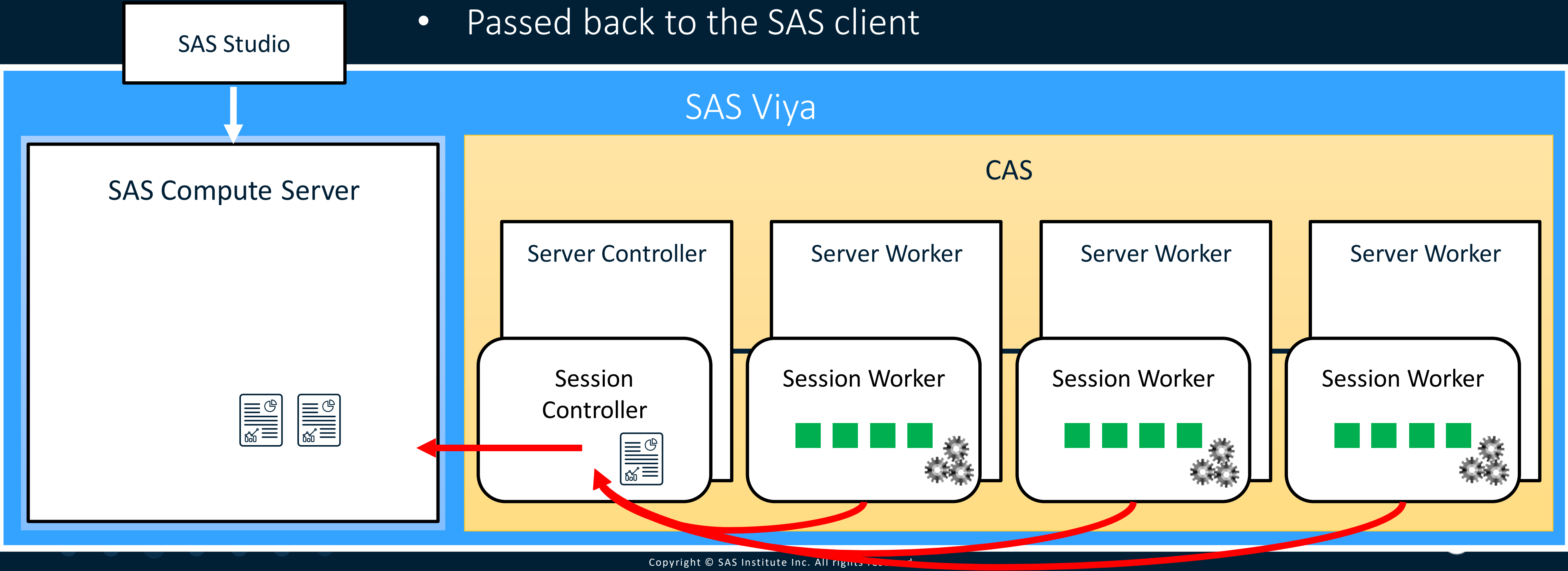
- Data in Viya Compute Server (SPRE)
- Work is done by the Compute Server single-threaded



SAS Viya Data Processing

CAS-enabled data processing

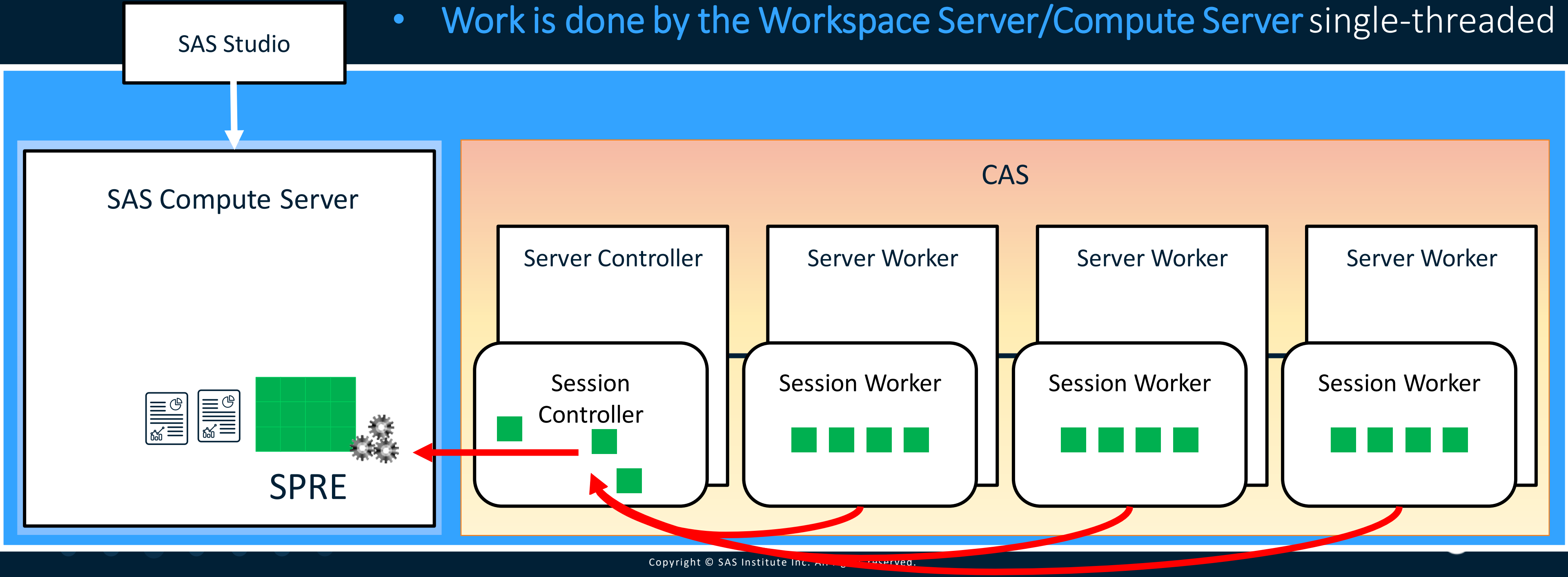
- Work is done by the CAS Session Workers
- Results passed to the CAS Session Controller for consolidation
- Passed back to the SAS client



SAS Viya Data Processing

NOT CAS-enabled data processing

- Data is retrieved from CAS
- Passed to the SAS 9 Workspace Server or Compute Server (SPRE)
- Work is done by the Workspace Server/Compute Server single-threaded



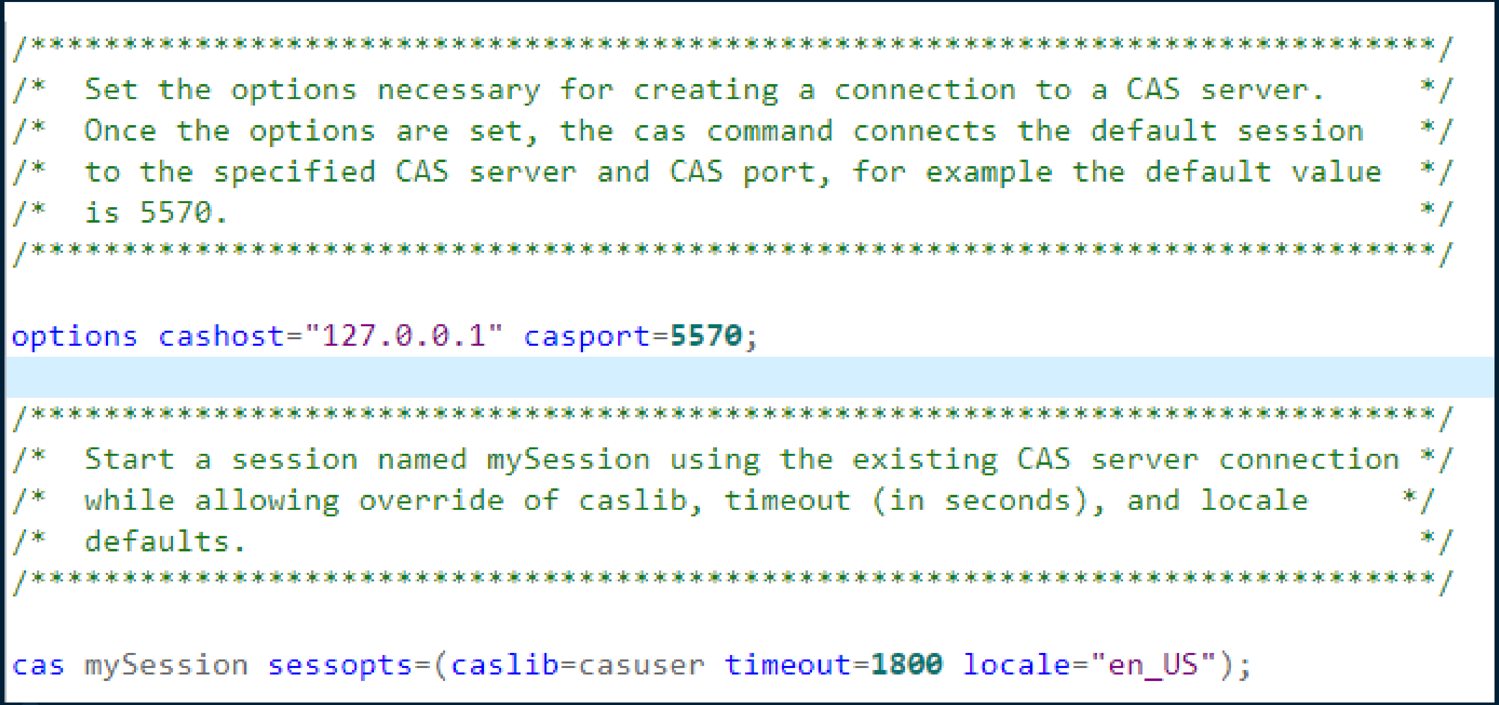
Viya Programming

In Eight Easy Steps

- Start a New CAS Session
- Create CASLIB(s) and Assign SAS Librefs to Access CASLIB(s)
- Load Data into CAS (and List CAS In-Memory Tables)
- Use DATA Step to Process CAS Tables
- Analyze Data Using SAS 9 Procedures
- Analyze Data Using Viya (CAS Enabled) Procedures
- Query Data Using PROC SQL and PROC FedSQL
- Format Your Results in CAS Using SAS Formats
- Clean Up After Yourself

Start a New CAS Session

Code

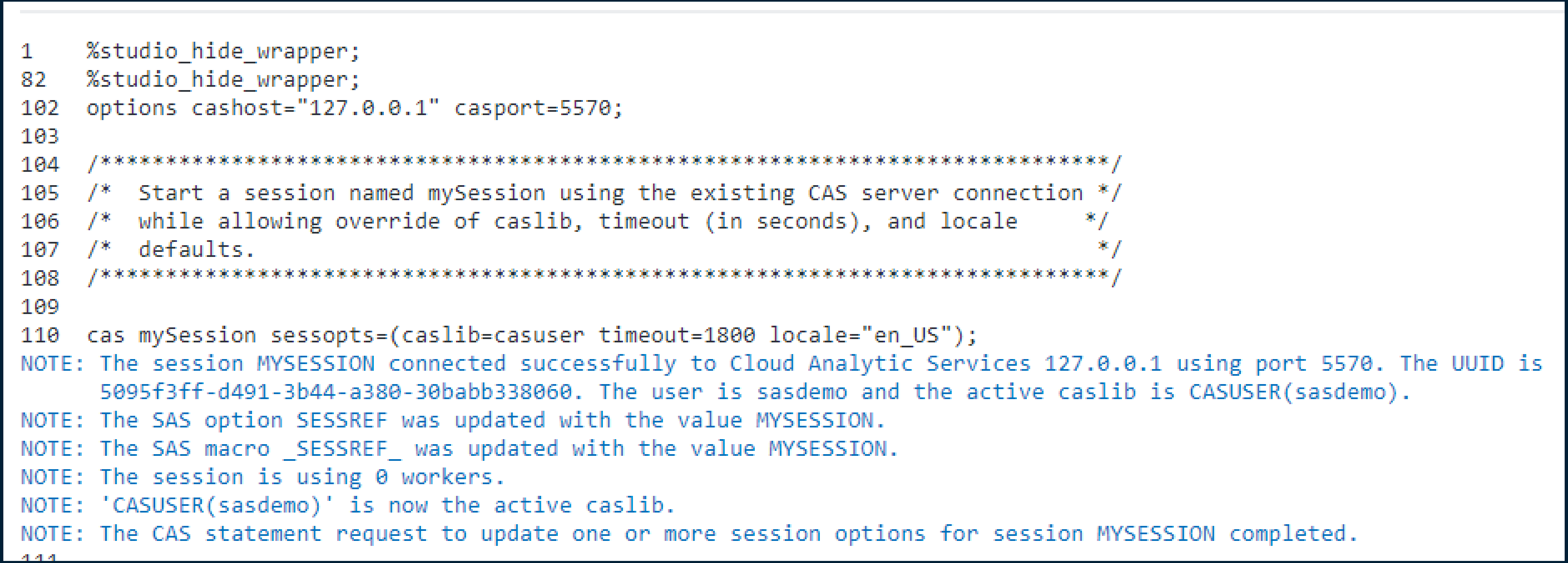


```
/*****************************************************************************/
/*  Set the options necessary for creating a connection to a CAS server.    */
/*  Once the options are set, the cas command connects the default session  */
/*  to the specified CAS server and CAS port, for example the default value */
/*  is 5570.                                                                 */
/*****************************************************************************/

options cashost="127.0.0.1" casport=5570;

/*****************************************************************************/
/*  Start a session named mySession using the existing CAS server connection */
/*  while allowing override of caslib, timeout (in seconds), and locale     */
/*  defaults.                                                                */
/*****************************************************************************/

cas mySession sessopts=(caslib=casuser timeout=1800 locale="en_US");
```

ViyaPgm_01 – Start CAS Session.sas

Start a New CAS Session

Log

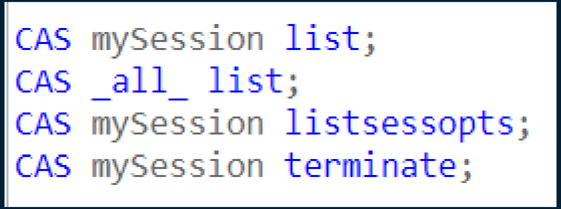


```
%studio_hide_wrapper;
%studio_hide_wrapper;
options cashost="127.0.0.1" casport=5570;

/****************************************************************************/
/*  Start a session named mySession using the existing CAS server connection */
/*  while allowing override of caslib, timeout (in seconds), and locale     */
/*  defaults.                                                                */
/****************************************************************************/

cas mySession sessopts=(caslib=casuser timeout=1800 locale="en_US");
NOTE: The session MYSESSION connected successfully to Cloud Analytic Services 127.0.0.1 using port 5570. The UUID is
      5095f3ff-d491-3b44-a380-30babb338060. The user is sasdemo and the active caslib is CASUSER(sasdemo).
NOTE: The SAS option SESSREF was updated with the value MYSESSION.
NOTE: The SAS macro _SESSREF_ was updated with the value MYSESSION.
NOTE: The session is using 0 workers.
NOTE: 'CASUSER(sasdemo)' is now the active caslib.
NOTE: The CAS statement request to update one or more session options for session MYSESSION completed.
```

ViyaPgm_01 – Start CAS Session.sas

CAS Statement

More Examples of Usage

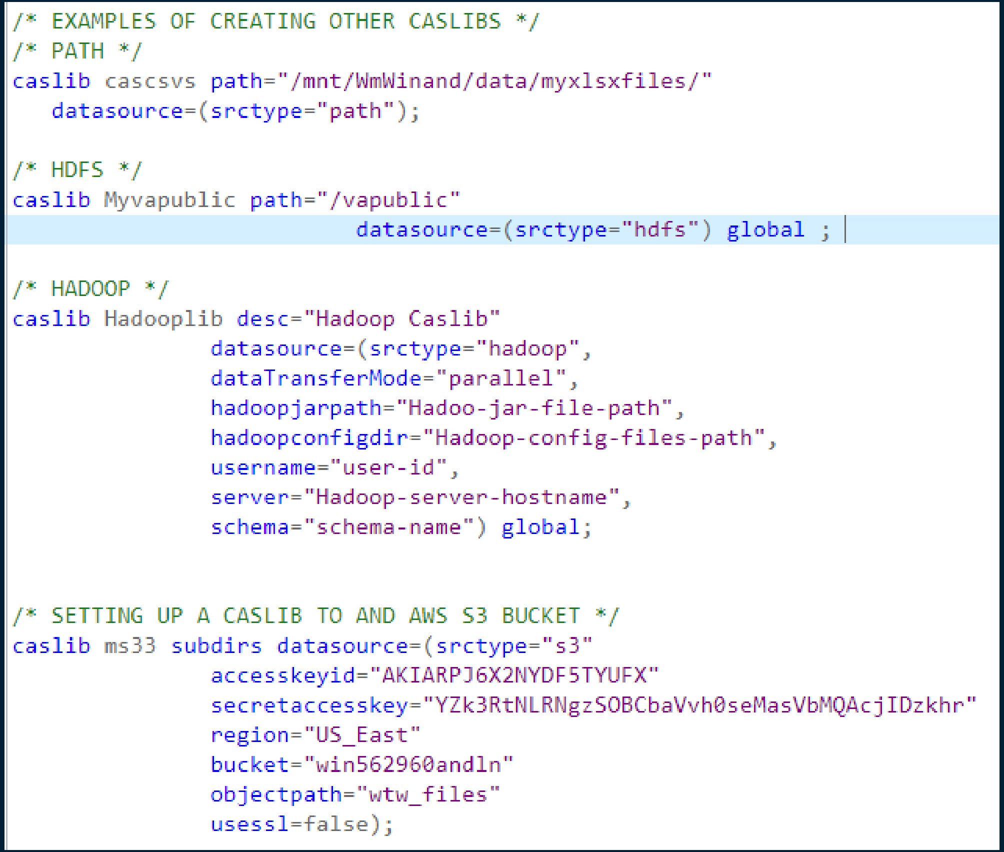


```
CAS mySession list;
CAS _all_ list;
CAS mySession listsessopts;
CAS mySession terminate;
```

CAS Statement

Create CASLIB(s) and Assign SAS Librefs to Access CASLIB(s)

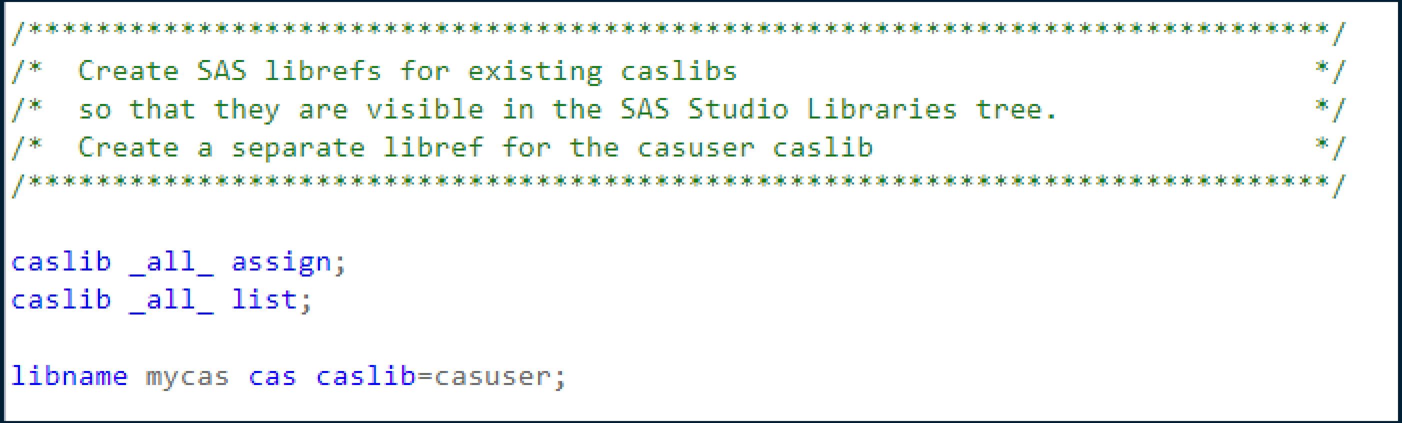
Creating CASLIB(s) - Examples



```
/* EXAMPLES OF CREATING OTHER CASLIBS */
/* PATH */
caslib cascsvs path="/mnt/WmWinand/data/myxlsxfiles/"
   datasource=(srctype="path");

/* HDFS */
caslib Myvapublic path="/vapublic"
                          datasource=(srctype="hdfs") global ;

/* HADOOP */
caslib Hadooplib desc="Hadoop Caslib"
               datasource=(srctype="hadoop",
               dataTransferMode="parallel",
               hadoopjarpath="Hadoo-jar-file-path",
               hadoopconfigdir="Hadoop-config-files-path",
               username="user-id",
               server="Hadoop-server-hostname",
               schema="schema-name") global;


/* SETTING UP A CASLIB TO AND AWS S3 BUCKET */
caslib ms33 subdirs datasource=(srctype="s3"
               accesskeyid="AKIARPJ6X2NYDF5TYUFX"
               secretaccesskey="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"
               region="US_East"
               bucket="win562960andln"
               objectpath="wtw_files"
               usessl=false);
```

Create CASLIB(s) and Assign SAS Librefs to Access CASLIB(s)

Assign SAS Librefs - Code

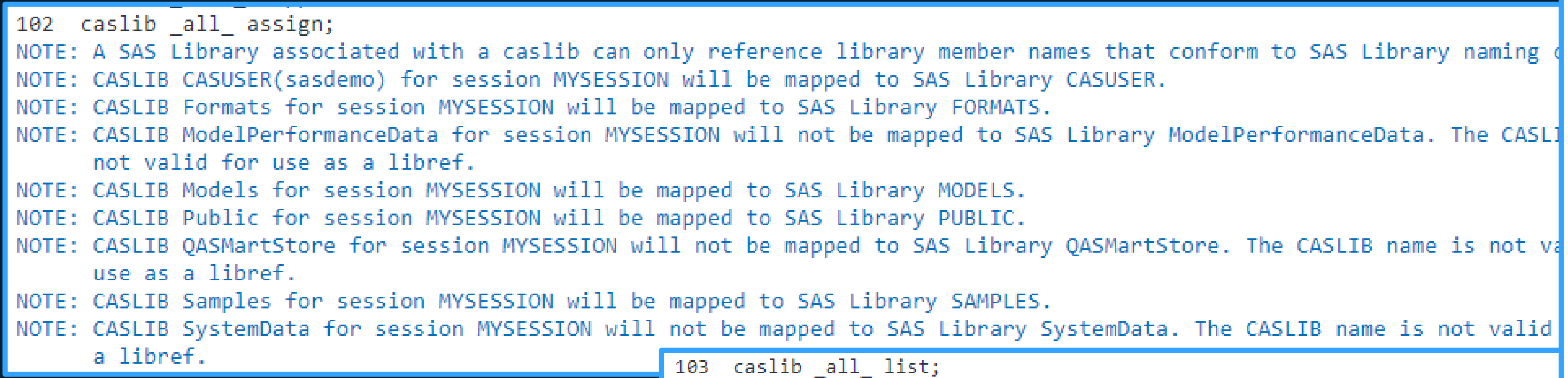


```
/******************************************************************************/
/*  Create SAS librefs for existing caslibs                                   */
/*  so that they are visible in the SAS Studio Libraries tree.                */
/*  Create a separate libref for the casuser caslib                           */
/******************************************************************************/

caslib _all_ assign;
caslib _all_ list;

libname mycas cas caslib=casuser;
```

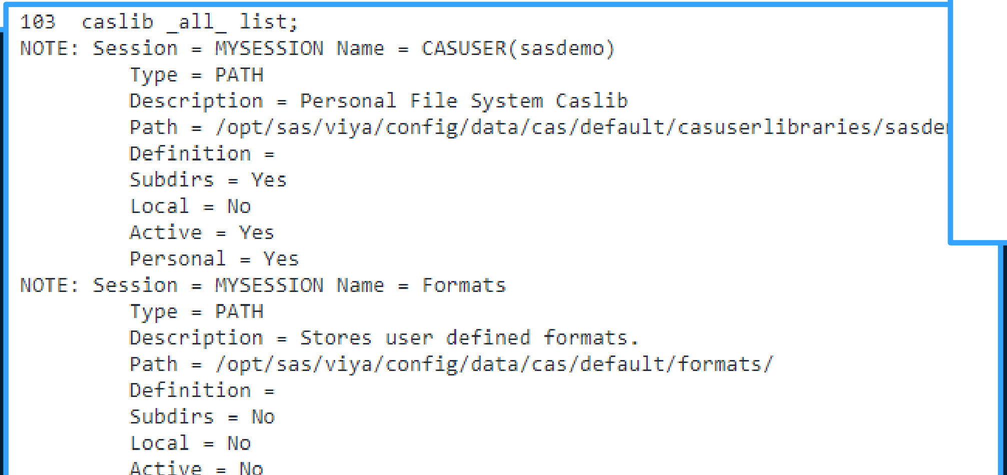
ViyaPgm_02 – Assign Librefs.sas

Create CASLIB(s) and Assign SAS Librefs to Access CASLIB(s)

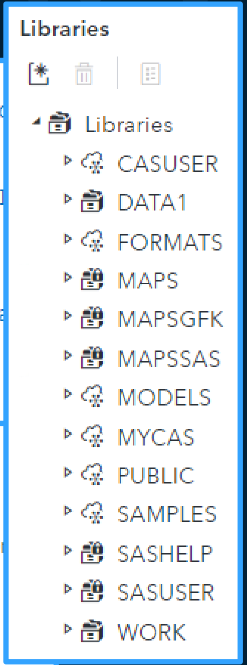
Assign SAS Librefs - Log & Libraries



```
102  caslib _all_ assign;
NOTE: A SAS Library associated with a caslib can only reference library member names that conform to SAS Library naming c
NOTE: CASLIB CASUSER(sasdemo) for session MYSESSION will be mapped to SAS Library CASUSER.
NOTE: CASLIB Formats for session MYSESSION will be mapped to SAS Library FORMATS.
NOTE: CASLIB ModelPerformanceData for session MYSESSION will not be mapped to SAS Library ModelPerformanceData. The CASLI
      not valid for use as a libref.
NOTE: CASLIB Models for session MYSESSION will be mapped to SAS Library MODELS.
NOTE: CASLIB Public for session MYSESSION will be mapped to SAS Library PUBLIC.
NOTE: CASLIB QASMartStore for session MYSESSION will not be mapped to SAS Library QASMartStore. The CASLIB name is not va
      use as a libref.
NOTE: CASLIB Samples for session MYSESSION will be mapped to SAS Library SAMPLES.
NOTE: CASLIB SystemData for session MYSESSION will not be mapped to SAS Library SystemData. The CASLIB name is not valid
      a libref.
```



```
103  caslib _all_ list;
NOTE: Session = MYSESSION Name = CASUSER(sasdemo)
          Type = PATH
          Description = Personal File System Caslib
          Path = /opt/sas/viya/config/data/cas/default/casuserlibraries/sasde
          Definition =
          Subdirs = Yes
          Local = No
          Active = Yes
          Personal = Yes
NOTE: Session = MYSESSION Name = Formats
          Type = PATH
          Description = Stores user defined formats.
          Path = /opt/sas/viya/config/data/cas/default/formats/
          Definition =
          Subdirs = No
          Local = No
          Active = No
```

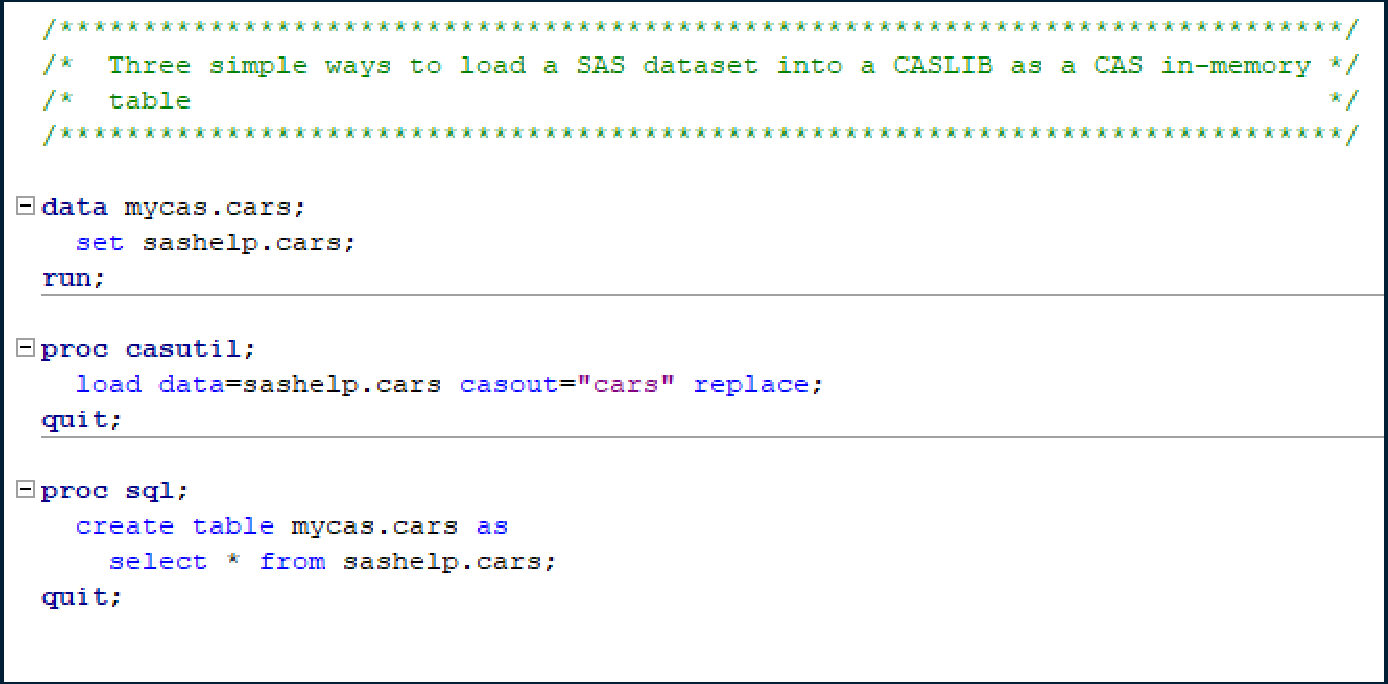


CASLIB Statement

ViyaPgm_02 – Assign Librefs.sas

Load Data into CAS

SAS Data Sets - Code

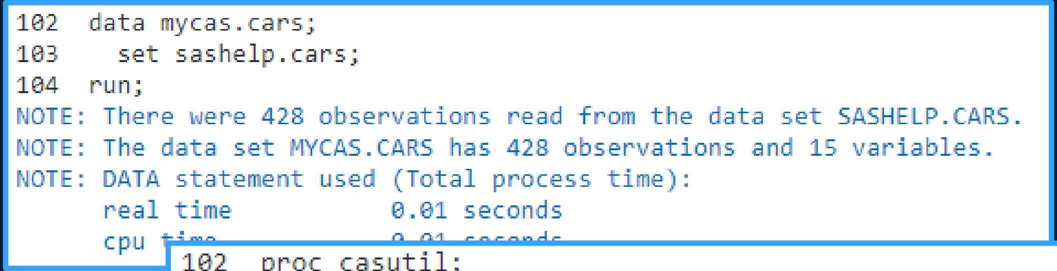


```
/*****************************************************************************/
/*  Three simple ways to load a SAS dataset into a CASLIB as a CAS in-memory */
/*  table                                                                    */
/*****************************************************************************/

data mycas.cars;
  set sashelp.cars;
run;

proc casutil;
  load data=sashelp.cars casout="cars" replace;
quit;

proc sql;
  create table mycas.cars as
    select * from sashelp.cars;
quit;
```

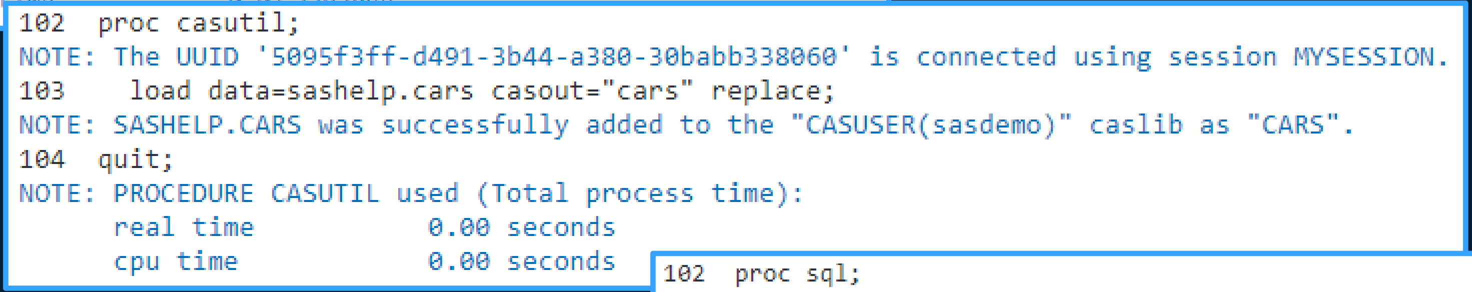
ViyaPgm_03 – Load Data into CAS

Load Data into CAS

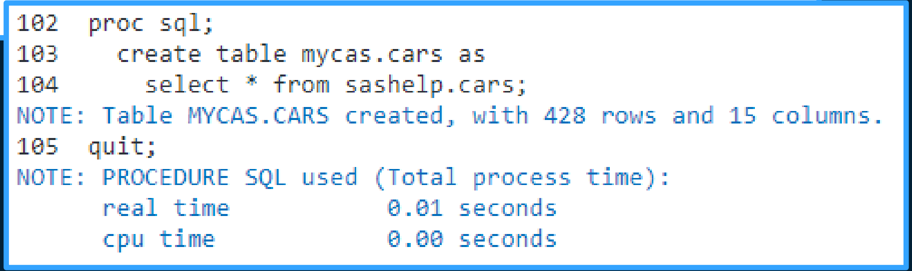
SAS Data Sets - Logs



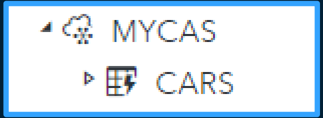
```
102  data mycas.cars;
103    set sashelp.cars;
104  run;
NOTE: There were 428 observations read from the data set SASHELP.CARS.
NOTE: The data set MYCAS.CARS has 428 observations and 15 variables.
NOTE: DATA statement used (Total process time):
      real time           0.01 seconds
      cpu time            0.01 seconds
```



```
102  proc casutil;
NOTE: The UUID '5095f3ff-d491-3b44-a380-30babb338060' is connected using session MYSESSION.
103    load data=sashelp.cars casout="cars" replace;
NOTE: SASHELP.CARS was successfully added to the "CASUSER(sasdemo)" caslib as "CARS".
104  quit;
NOTE: PROCEDURE CASUTIL used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds
```



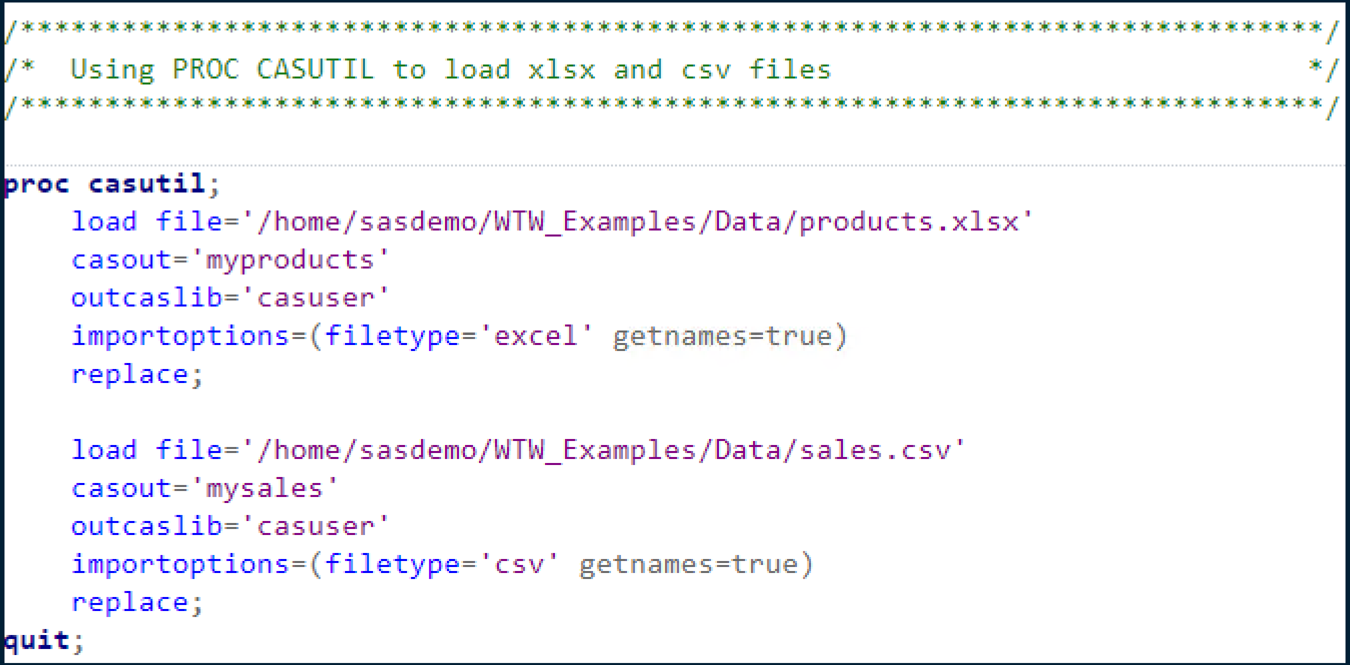
```
102  proc sql;
103    create table mycas.cars as
104      select * from sashelp.cars;
NOTE: Table MYCAS.CARS created, with 428 rows and 15 columns.
105  quit;
NOTE: PROCEDURE SQL used (Total process time):
      real time           0.01 seconds
      cpu time            0.00 seconds
```



ViyaPgm_03 – Load Data into CAS

Load Data into CAS

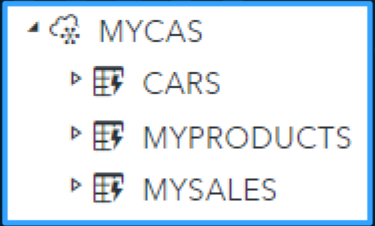
Excel and CSV Files - Code

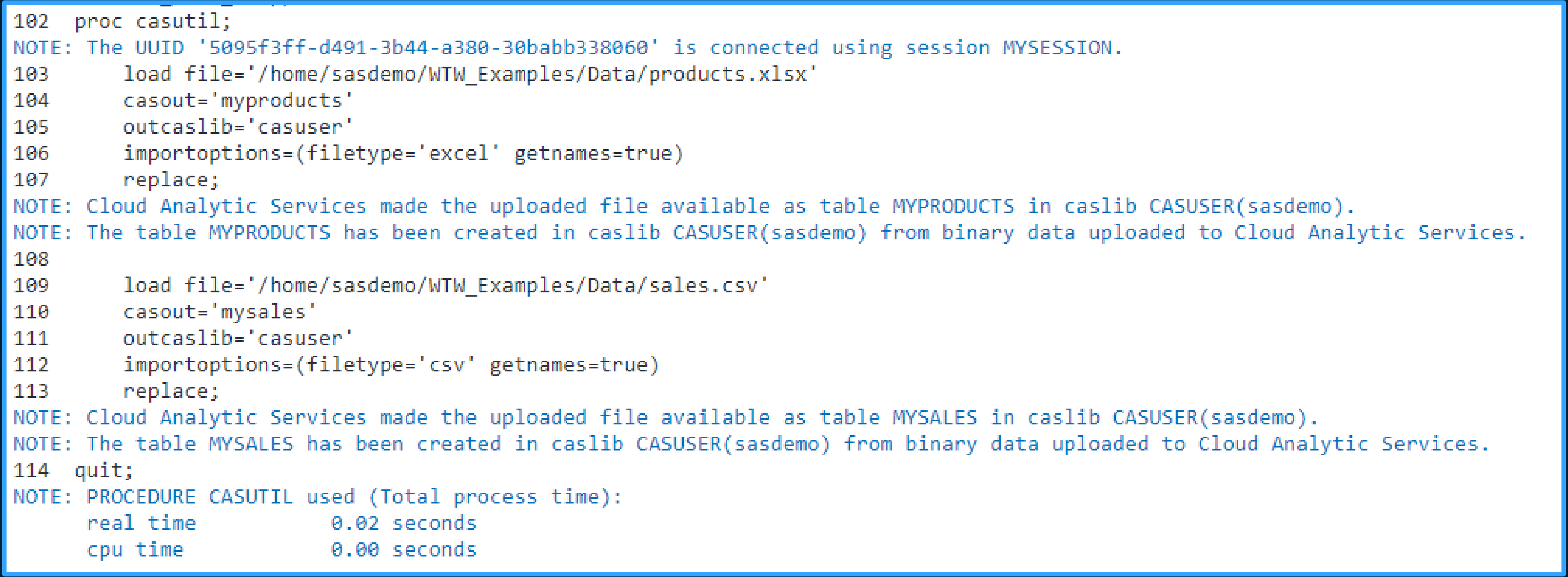


```
/******************************************************************************/
/*  Using PROC CASUTIL to load xlsx and csv files                           */
/******************************************************************************/

proc casutil;
    load file='/home/sasdemo/WTW_Examples/Data/products.xlsx'
    casout='myproducts'
    outcaslib='casuser'
    importoptions=(filetype='excel' getnames=true)
    replace;

    load file='/home/sasdemo/WTW_Examples/Data/sales.csv'
    casout='mysales'
    outcaslib='casuser'
    importoptions=(filetype='csv' getnames=true)
    replace;
quit;
```

Load Data into CAS

Excel and CSV Files - Log

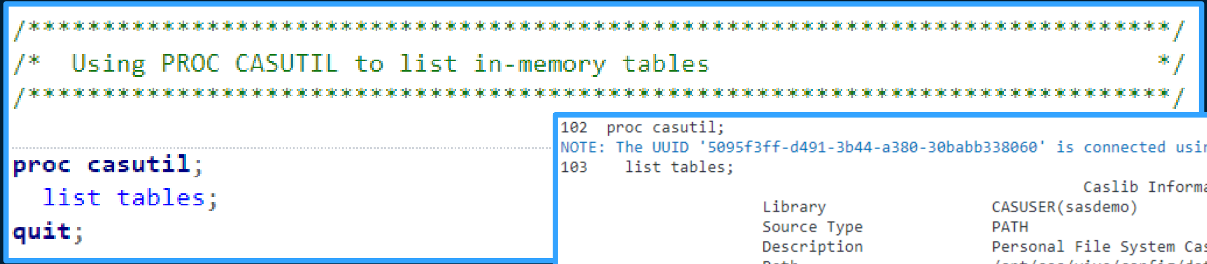




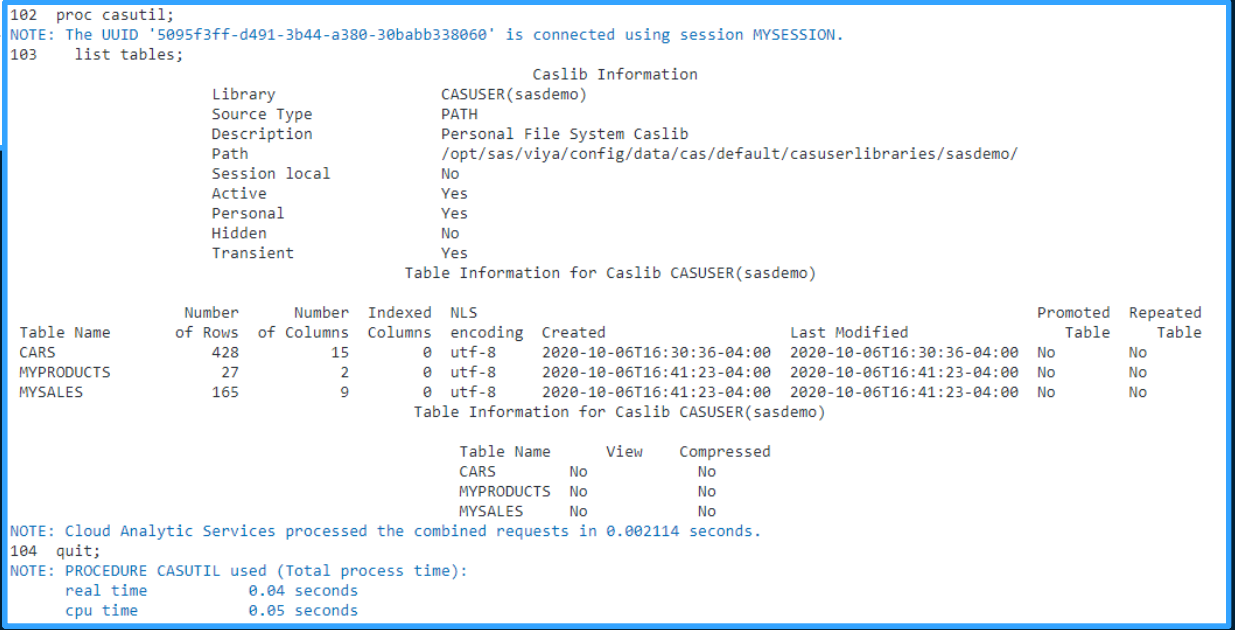
```
102  proc casutil;
NOTE: The UUID '5095f3ff-d491-3b44-a380-30babb338060' is connected using session MYSESSION.
103      load file='/home/sasdemo/WTW_Examples/Data/products.xlsx'
104      casout='myproducts'
105      outcaslib='casuser'
106      importoptions=(filetype='excel' getnames=true)
107      replace;
NOTE: Cloud Analytic Services made the uploaded file available as table MYPRODUCTS in caslib CASUSER(sasdemo).
NOTE: The table MYPRODUCTS has been created in caslib CASUSER(sasdemo) from binary data uploaded to Cloud Analytic Services.
108
109      load file='/home/sasdemo/WTW_Examples/Data/sales.csv'
110      casout='mysales'
111      outcaslib='casuser'
112      importoptions=(filetype='csv' getnames=true)
113      replace;
NOTE: Cloud Analytic Services made the uploaded file available as table MYSALES in caslib CASUSER(sasdemo).
NOTE: The table MYSALES has been created in caslib CASUSER(sasdemo) from binary data uploaded to Cloud Analytic Services.
114  quit;
NOTE: PROCEDURE CASUTIL used (Total process time):
      real time           0.02 seconds
      cpu time            0.00 seconds
```

List CAS In-Memory Tables

Code & Log



```
/*****************************************************************/
/*  Using PROC CASUTIL to list in-memory tables                 */
/*****************************************************************/

proc casutil;
  list tables;
quit;
```

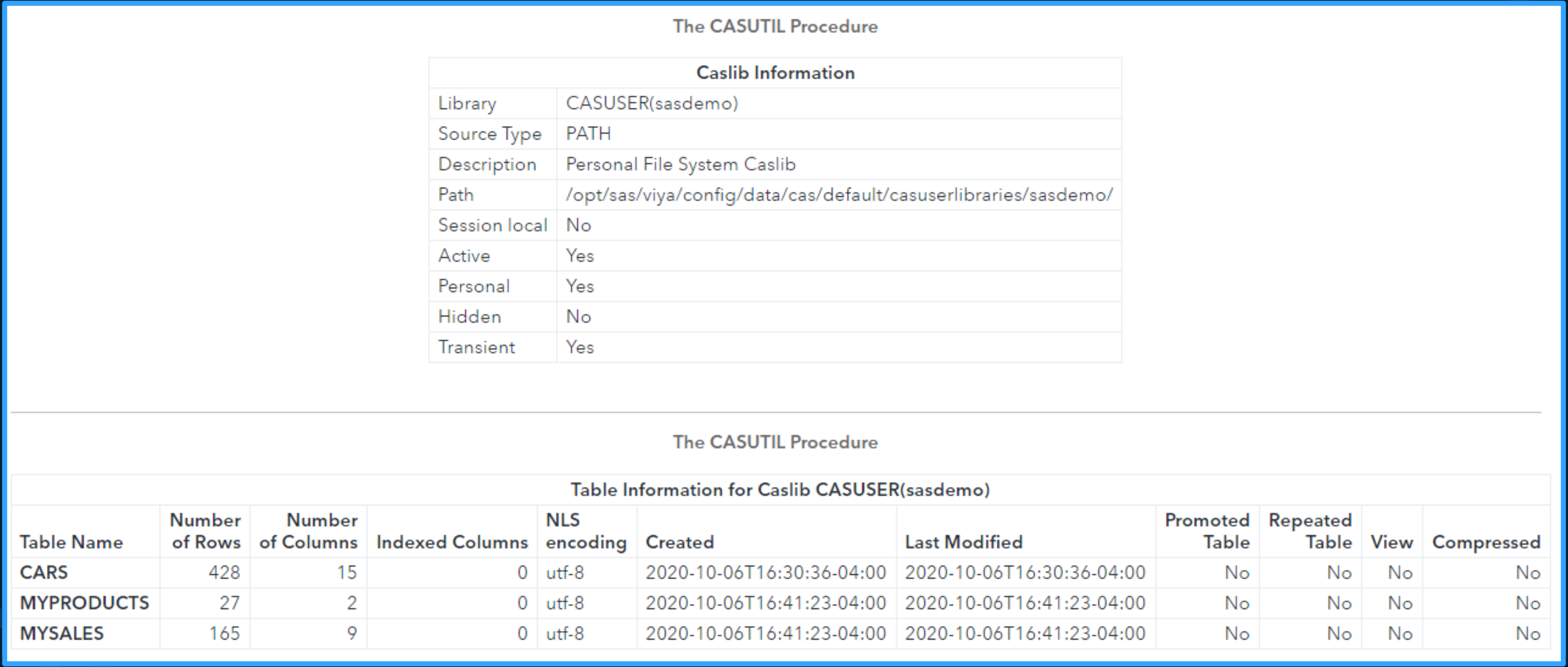


```
102  proc casutil;
NOTE: The UUID '5095f3ff-d491-3b44-a380-30babb338060' is connected using session MYSESSION.
103    list tables;
                                              Caslib Information
                      Library                    CASUSER(sasdemo)
                      Source Type                PATH
                      Description                Personal File System Caslib
                      Path                       /opt/sas/viya/config/data/cas/default/casuserlibraries/sasdemo/
                      Session local              No
                      Active                     Yes
                      Personal                   Yes
                      Hidden                     No
                      Transient                  Yes
                                        Table Information for Caslib CASUSER(sasdemo)

                  Number      Number  Indexed  NLS                                                         Promoted  Repeated
 Table Name      of Rows  of Columns  Columns  encoding  Created                    Last Modified              Table     Table
 CARS                428          15        0  utf-8     2020-10-06T16:30:36-04:00  2020-10-06T16:30:36-04:00  No        No
 MYPRODUCTS           27           2        0  utf-8     2020-10-06T16:41:23-04:00  2020-10-06T16:41:23-04:00  No        No
 MYSALES             165           9        0  utf-8     2020-10-06T16:41:23-04:00  2020-10-06T16:41:23-04:00  No        No
                                         Table Information for Caslib CASUSER(sasdemo)

                                                Table Name      View    Compressed
                                                CARS        No            No
                                                MYPRODUCTS  No            No
                                                MYSALES     No            No
NOTE: Cloud Analytic Services processed the combined requests in 0.002114 seconds.
104  quit;
NOTE: PROCEDURE CASUTIL used (Total process time):
      real time           0.04 seconds
      cpu time            0.05 seconds
```

ViyaPgm_04 – List CAS Table Info.sas

List CAS In-Memory Tables

Results



The CASUTIL Procedure

Caslib Information	
Library	CASUSER(sasdemo)
Source Type	PATH
Description	Personal File System Caslib
Path	/opt/sas/viya/config/data/cas/default/casuserlibraries/sasdemo/
Session local	No
Active	Yes
Personal	Yes
Hidden	No
Transient	Yes

The CASUTIL Procedure

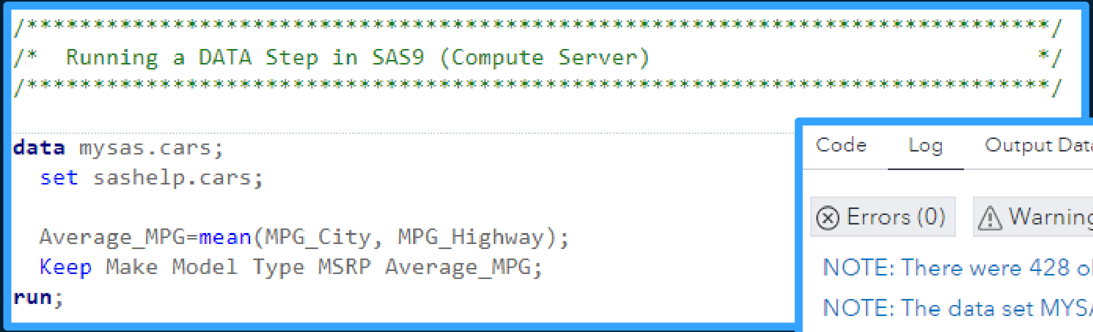
Table Information for Caslib CASUSER(sasdemo)

Table Name	Number of Rows	Number of Columns	Indexed Columns	NLS encoding	Created	Last Modified	Promoted Table	Repeated Table	View	Compressed
CARS	428	15	0	utf-8	2020-10-06T16:30:36-04:00	2020-10-06T16:30:36-04:00	No	No	No	No
MYPRODUCTS	27	2	0	utf-8	2020-10-06T16:41:23-04:00	2020-10-06T16:41:23-04:00	No	No	No	No
MYSALES	165	9	0	utf-8	2020-10-06T16:41:23-04:00	2020-10-06T16:41:23-04:00	No	No	No	No

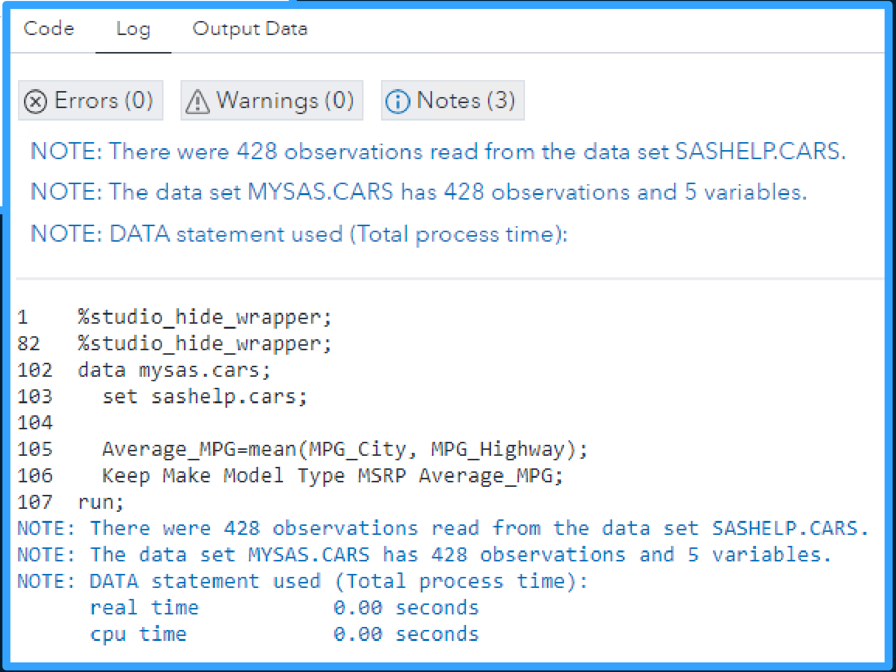
ViyaPgm_04 – List CAS Table Info.sas

Use DATA Step on Compute Server

Code & Log



```
/*****************************************************************************/
/*  Running a DATA Step in SAS9 (Compute Server)                            */
/*****************************************************************************/

data mysas.cars;
  set sashelp.cars;

  Average_MPG=mean(MPG_City, MPG_Highway);
  Keep Make Model Type MSRP Average_MPG;
run;
```



NOTE: There were 428 observations read from the data set SASHELP.CARS.

NOTE: The data set MYSAS.CARS has 428 observations and 5 variables.

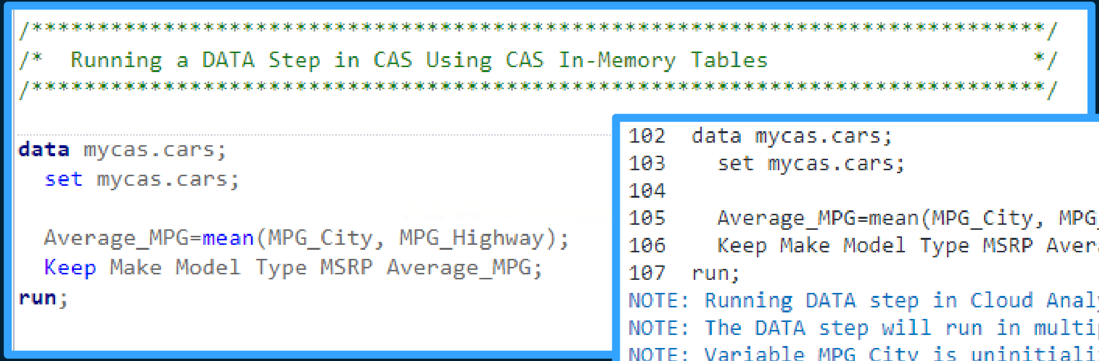
NOTE: DATA statement used (Total process time):

```
1    %studio_hide_wrapper;
82   %studio_hide_wrapper;
102  data mysas.cars;
103    set sashelp.cars;
104
105    Average_MPG=mean(MPG_City, MPG_Highway);
106    Keep Make Model Type MSRP Average_MPG;
107  run;
NOTE: There were 428 observations read from the data set SASHELP.CARS.
NOTE: The data set MYSAS.CARS has 428 observations and 5 variables.
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds
```

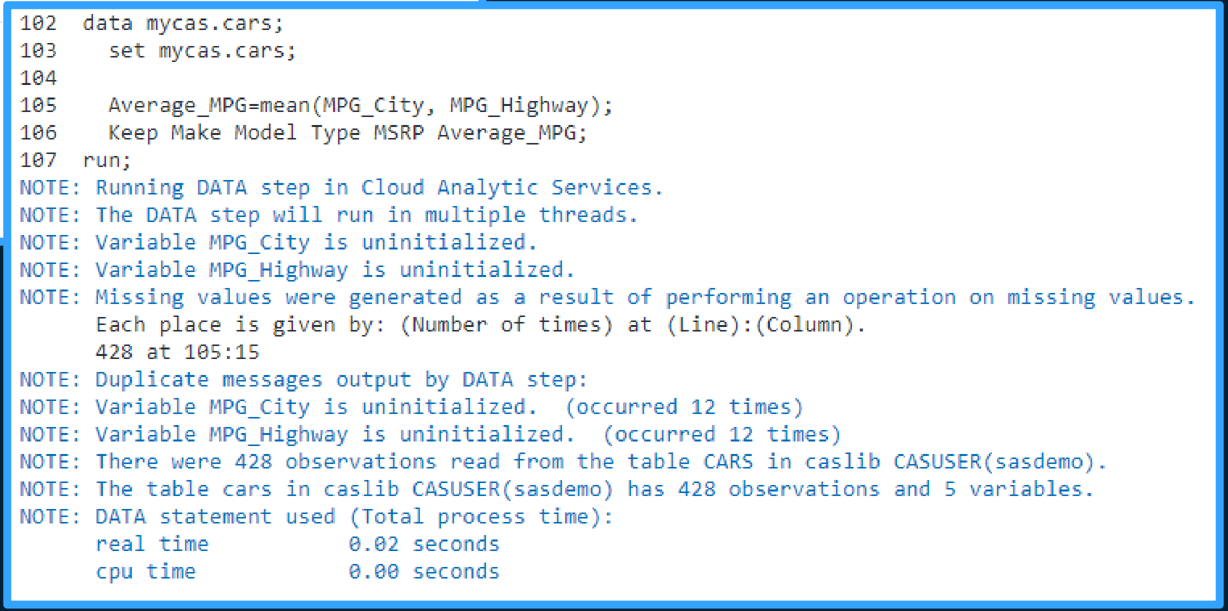
ViyaPgm_05 – DATA Step in CAS.sas

Use DATA Step in CAS

Code & Log



```
/***************************************************************************/
/*  Running a DATA Step in CAS Using CAS In-Memory Tables                  */
/***************************************************************************/

data mycas.cars;
  set mycas.cars;

  Average_MPG=mean(MPG_City, MPG_Highway);
  Keep Make Model Type MSRP Average_MPG;
run;
```

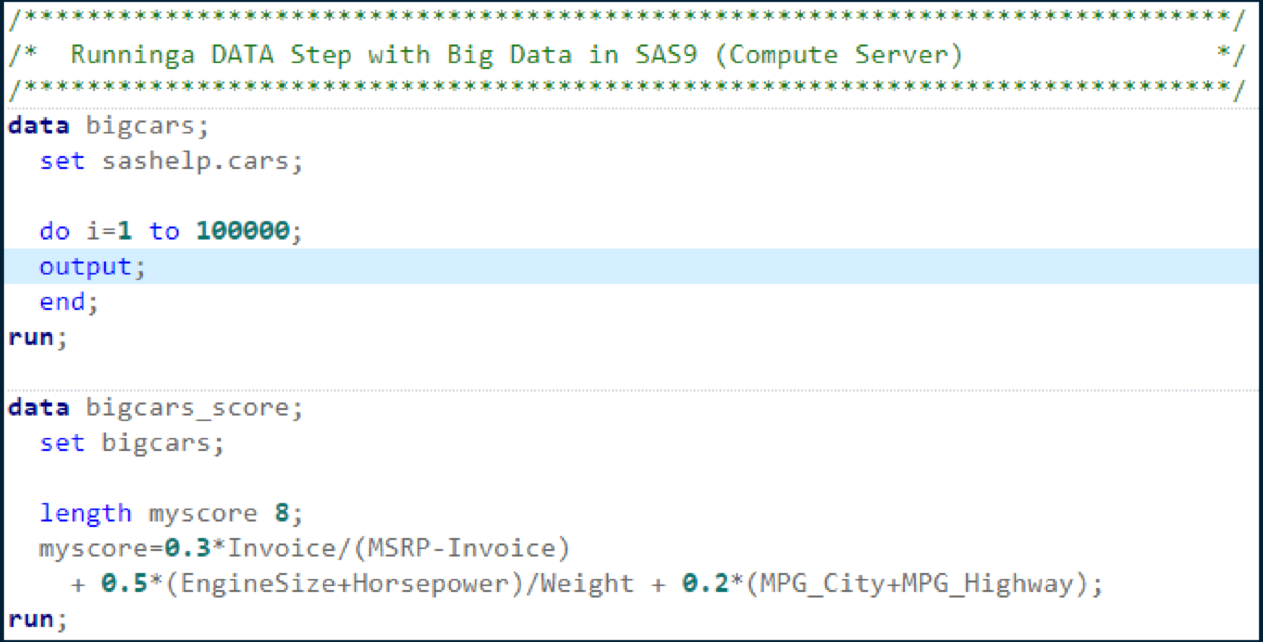


```
102  data mycas.cars;
103    set mycas.cars;
104
105    Average_MPG=mean(MPG_City, MPG_Highway);
106    Keep Make Model Type MSRP Average_MPG;
107  run;
NOTE: Running DATA step in Cloud Analytic Services.
NOTE: The DATA step will run in multiple threads.
NOTE: Variable MPG_City is uninitialized.
NOTE: Variable MPG_Highway is uninitialized.
NOTE: Missing values were generated as a result of performing an operation on missing values.
      Each place is given by: (Number of times) at (Line):(Column).
      428 at 105:15
NOTE: Duplicate messages output by DATA step:
NOTE: Variable MPG_City is uninitialized.  (occurred 12 times)
NOTE: Variable MPG_Highway is uninitialized.  (occurred 12 times)
NOTE: There were 428 observations read from the table CARS in caslib CASUSER(sasdemo).
NOTE: The table cars in caslib CASUSER(sasdemo) has 428 observations and 5 variables.
NOTE: DATA statement used (Total process time):
      real time           0.02 seconds
      cpu time            0.00 seconds
```

ViyaPgm_05 – DATA Step in CAS.sas

Use DATA Step with “BIG” Data on Compute Server

Code

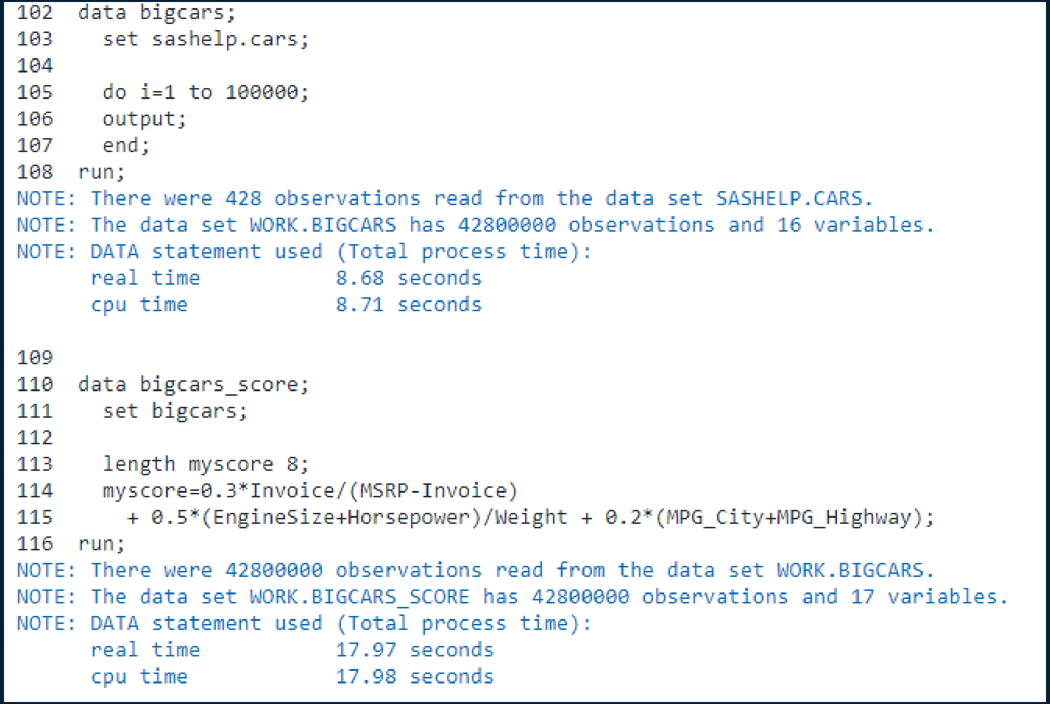


```
/***************************************************************************/
/*   Runninga DATA Step with Big Data in SAS9 (Compute Server)            */
/***************************************************************************/
data bigcars;
  set sashelp.cars;

  do i=1 to 100000;
  output;
  end;
run;

data bigcars_score;
  set bigcars;

  length myscore 8;
  myscore=0.3*Invoice/(MSRP-Invoice)
    + 0.5*(EngineSize+Horsepower)/Weight + 0.2*(MPG_City+MPG_Highway);
run;
```

ViyaPgm_06 – DATA Step with Big Data in CAS.sas

Use DATA Step with “BIG” Data on Compute Server

Log

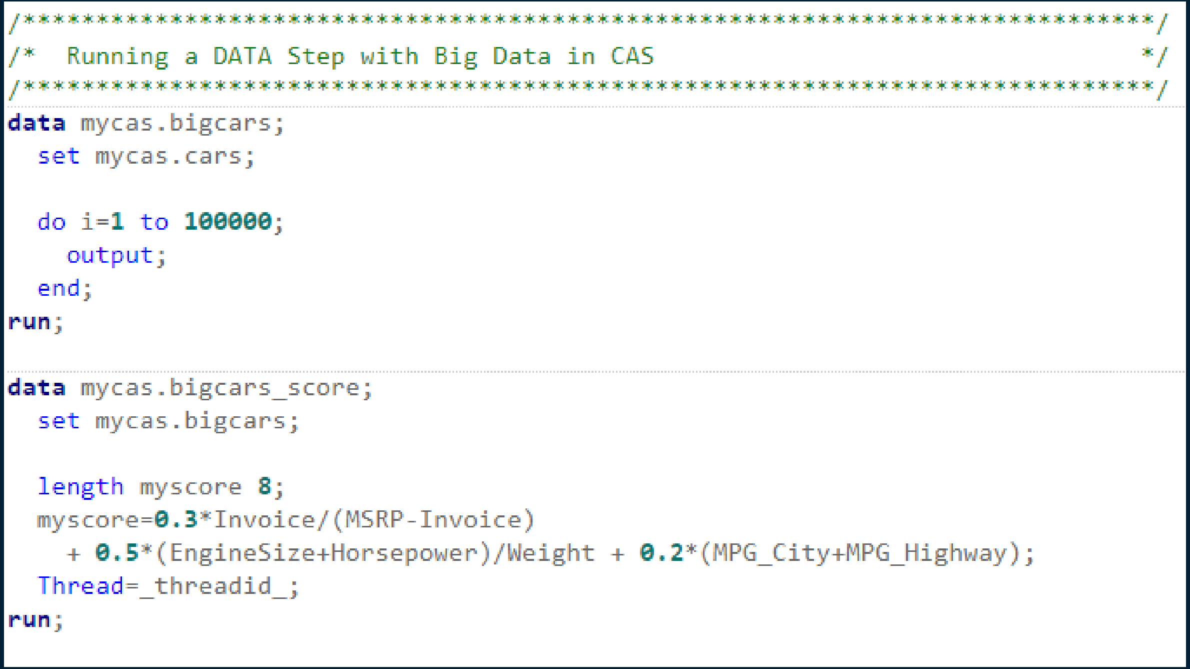


```
data bigcars;
  set sashelp.cars;

  do i=1 to 100000;
  output;
  end;
run;
NOTE: There were 428 observations read from the data set SASHELP.CARS.
NOTE: The data set WORK.BIGCARS has 42800000 observations and 16 variables.
NOTE: DATA statement used (Total process time):
      real time           8.68 seconds
      cpu time            8.71 seconds


data bigcars_score;
  set bigcars;

  length myscore 8;
  myscore=0.3*Invoice/(MSRP-Invoice)
    + 0.5*(EngineSize+Horsepower)/Weight + 0.2*(MPG_City+MPG_Highway);
run;
NOTE: There were 42800000 observations read from the data set WORK.BIGCARS.
NOTE: The data set WORK.BIGCARS_SCORE has 42800000 observations and 17 variables.
NOTE: DATA statement used (Total process time):
      real time           17.97 seconds
      cpu time            17.98 seconds
```

ViyaPgm_06 – DATA Step with Big Data in CAS.sas

Use DATA Step with “BIG” Data in CAS

Code

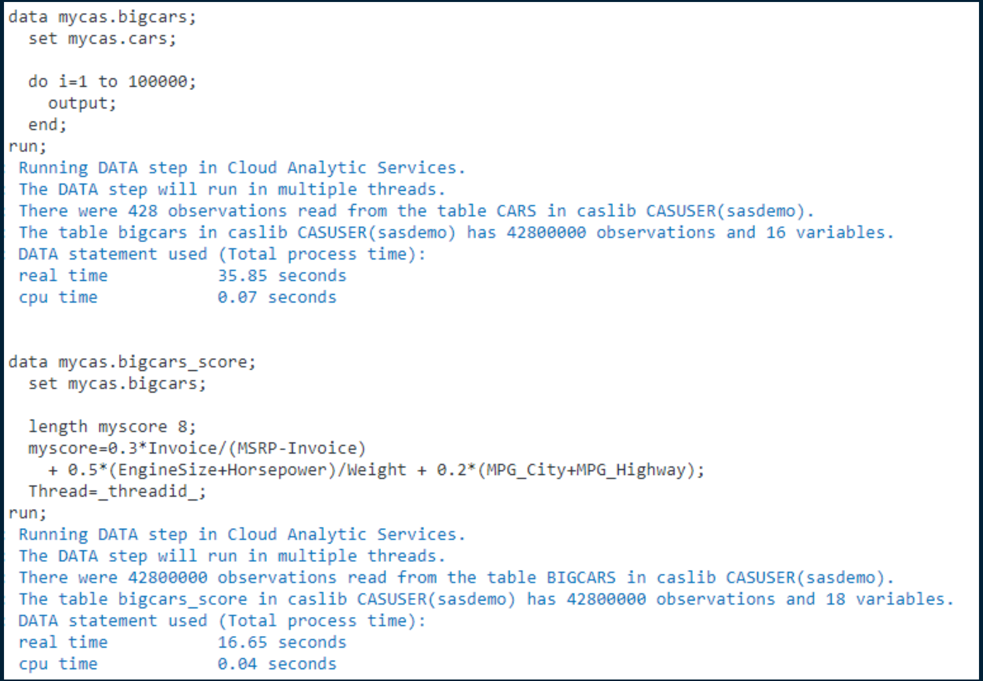


```
/*****************************************************************************/
/*  Running a DATA Step with Big Data in CAS                                 */
/*****************************************************************************/
data mycas.bigcars;
  set mycas.cars;

  do i=1 to 100000;
    output;
  end;
run;

data mycas.bigcars_score;
  set mycas.bigcars;

  length myscore 8;
  myscore=0.3*Invoice/(MSRP-Invoice)
    + 0.5*(EngineSize+Horsepower)/Weight + 0.2*(MPG_City+MPG_Highway);
  Thread=_threadid_;
run;
```

ViyaPgm_06 – DATA Step with Big Data in CAS.sas

Use DATA Step with “BIG” Data in CAS

Log

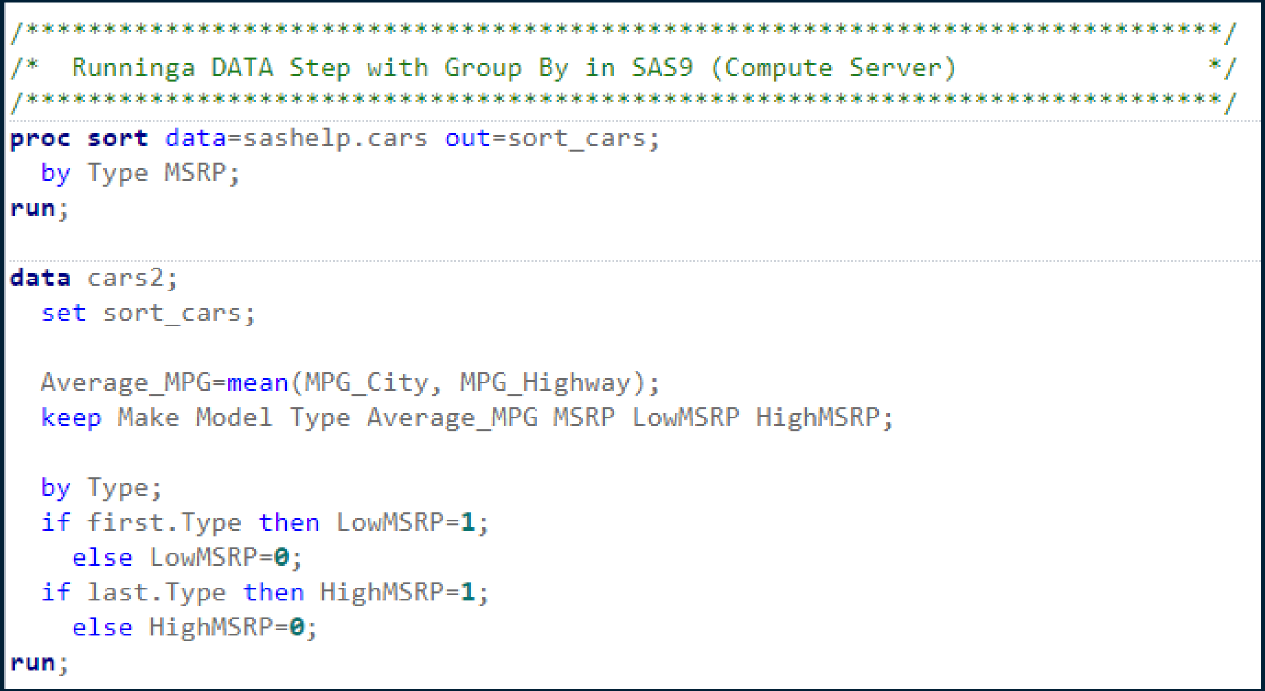


```
data mycas.bigcars;
  set mycas.cars;

  do i=1 to 100000;
    output;
  end;
run;
 Running DATA step in Cloud Analytic Services.
 The DATA step will run in multiple threads.
 There were 428 observations read from the table CARS in caslib CASUSER(sasdemo).
 The table bigcars in caslib CASUSER(sasdemo) has 42800000 observations and 16 variables.
 DATA statement used (Total process time):
 real time           35.85 seconds
 cpu time            0.07 seconds


data mycas.bigcars_score;
  set mycas.bigcars;

  length myscore 8;
  myscore=0.3*Invoice/(MSRP-Invoice)
    + 0.5*(EngineSize+Horsepower)/Weight + 0.2*(MPG_City+MPG_Highway);
  Thread=_threadid_;
run;
 Running DATA step in Cloud Analytic Services.
 The DATA step will run in multiple threads.
 There were 42800000 observations read from the table BIGCARS in caslib CASUSER(sasdemo).
 The table bigcars_score in caslib CASUSER(sasdemo) has 42800000 observations and 18 variables.
 DATA statement used (Total process time):
 real time           16.65 seconds
 cpu time            0.04 seconds
```

ViyaPgm_06 – DATA Step with Big Data in CAS.sas

Use DATA Step with By Group Processing – Compute Server

Code

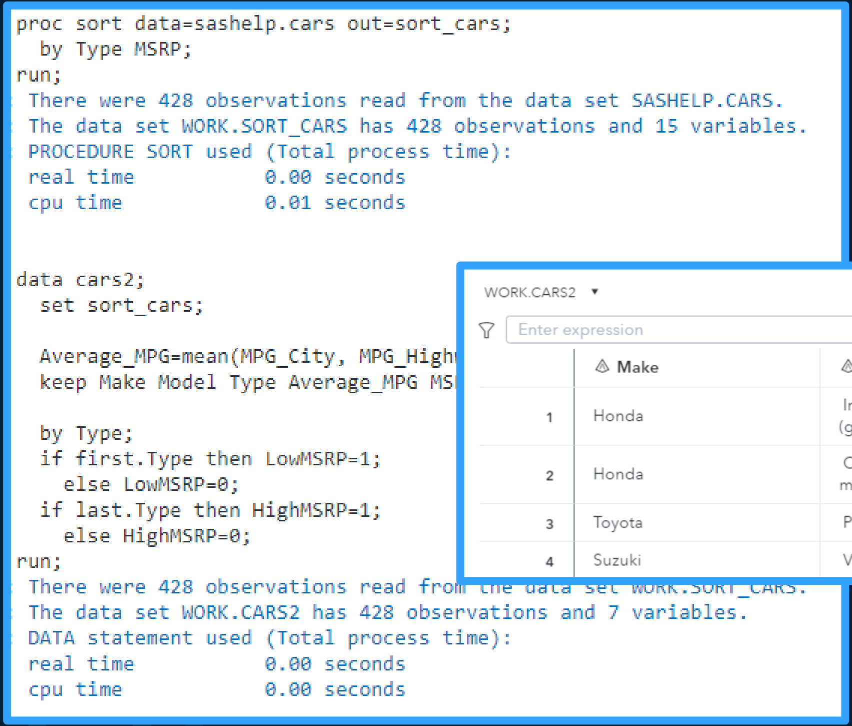


```
/***************************************************************************/
/*  Runninga DATA Step with Group By in SAS9 (Compute Server)             */
/***************************************************************************/
proc sort data=sashelp.cars out=sort_cars;
  by Type MSRP;
run;

data cars2;
  set sort_cars;

  Average_MPG=mean(MPG_City, MPG_Highway);
  keep Make Model Type Average_MPG MSRP LowMSRP HighMSRP;

  by Type;
  if first.Type then LowMSRP=1;
    else LowMSRP=0;
  if last.Type then HighMSRP=1;
    else HighMSRP=0;
run;
```

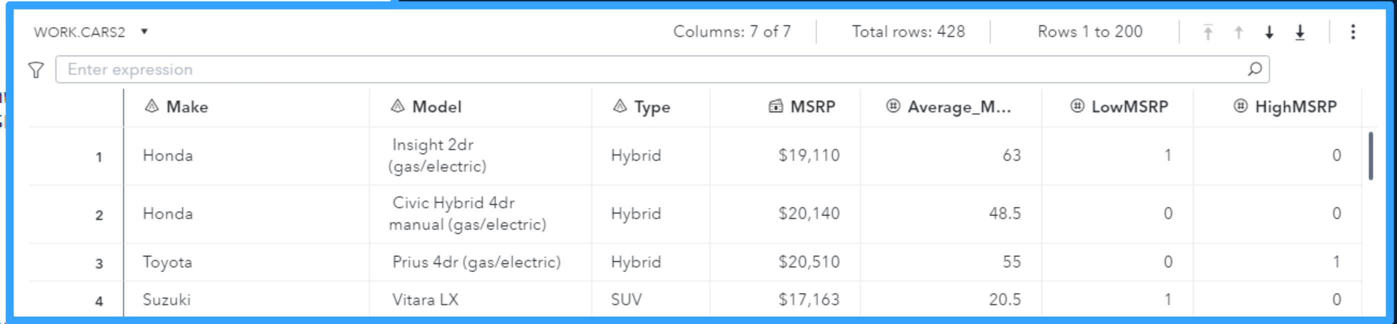
ViyaPgm_07 – DATA Step with Group By in CAS.sas

Use DATA Step with By Group Processing – Compute Server

Log & Output



```
proc sort data=sashelp.cars out=sort_cars;
  by Type MSRP;
run;
 There were 428 observations read from the data set SASHELP.CARS.
 The data set WORK.SORT_CARS has 428 observations and 15 variables.
 PROCEDURE SORT used (Total process time):
 real time           0.00 seconds
 cpu time            0.01 seconds


data cars2;
  set sort_cars;

  Average_MPG=mean(MPG_City, MPG_High
  keep Make Model Type Average_MPG MS

  by Type;
  if first.Type then LowMSRP=1;
    else LowMSRP=0;
  if last.Type then HighMSRP=1;
    else HighMSRP=0;
run;
 There were 428 observations read from the data set WORK.SORT_CARS.
 The data set WORK.CARS2 has 428 observations and 7 variables.
 DATA statement used (Total process time):
 real time           0.00 seconds
 cpu time            0.00 seconds
```

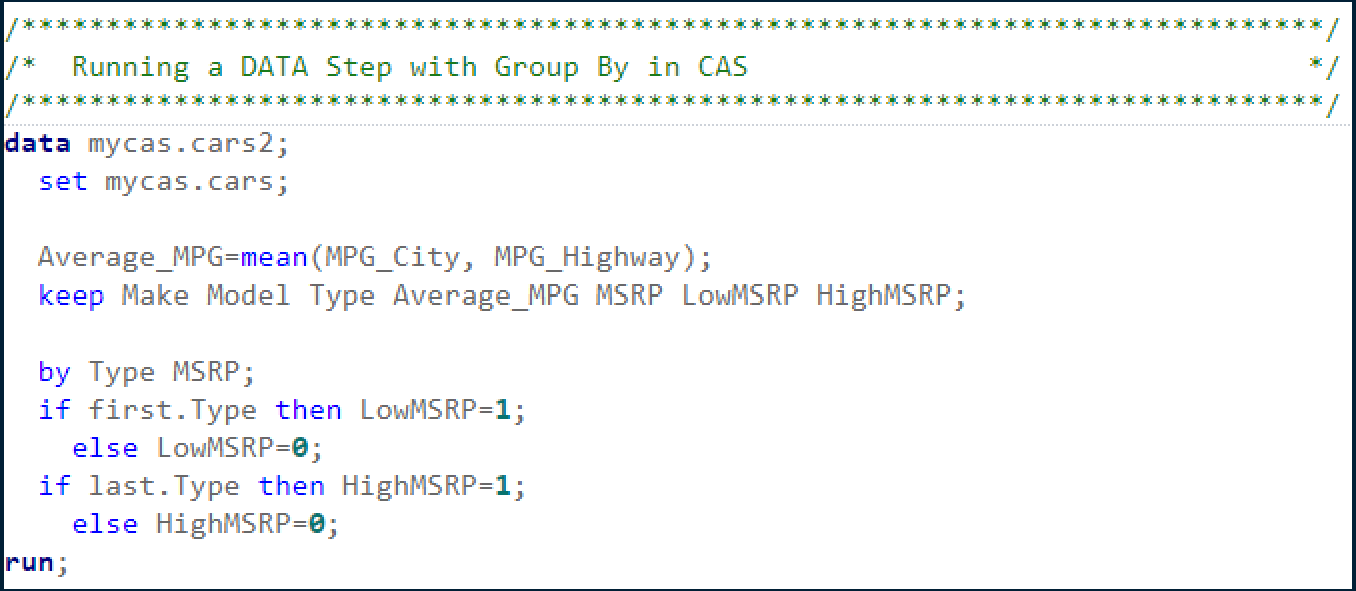


	Make	Model	Type	MSRP	Average_M...	LowMSRP	HighMSRP
1	Honda	Insight 2dr (gas/electric)	Hybrid	$19,110	63	1	0
2	Honda	Civic Hybrid 4dr manual (gas/electric)	Hybrid	$20,140	48.5	0	0
3	Toyota	Prius 4dr (gas/electric)	Hybrid	$20,510	55	0	1
4	Suzuki	Vitara LX	SUV	$17,163	20.5	1	0

ViyaPgm_07 – DATA Step with Group By in CAS.sas

Use DATA Step with By Group Processing - CAS

Code

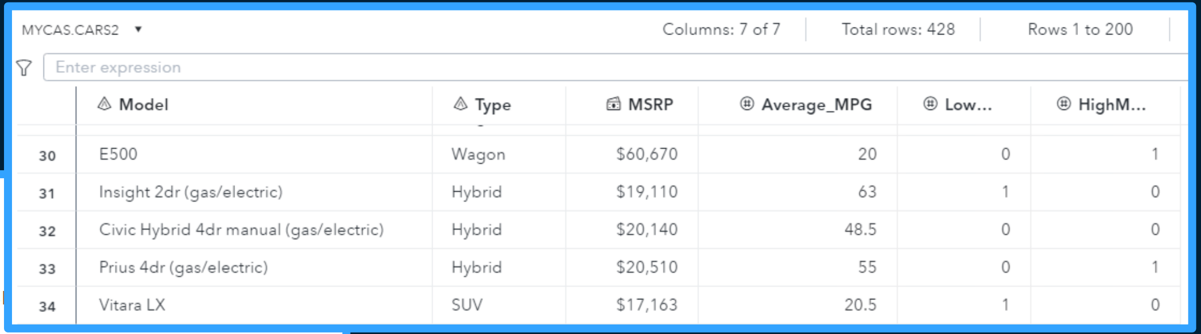


```
/******************************************************************************/
/*  Running a DATA Step with Group By in CAS                                  */
/******************************************************************************/
data mycas.cars2;
  set mycas.cars;

  Average_MPG=mean(MPG_City, MPG_Highway);
  keep Make Model Type Average_MPG MSRP LowMSRP HighMSRP;

  by Type MSRP;
  if first.Type then LowMSRP=1;
    else LowMSRP=0;
  if last.Type then HighMSRP=1;
    else HighMSRP=0;
run;
```

ViyaPgm_07 – DATA Step with Group By in CAS.sas

Use DATA Step with By Group Processing - CAS

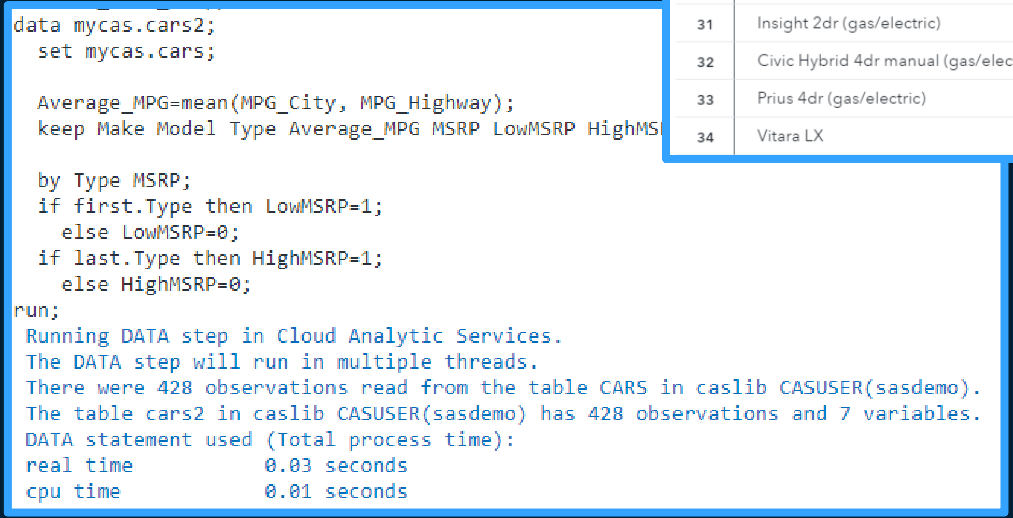
Log & Output



MYCAS.CARS2 ▾ Columns: 7 of 7 | Total rows: 428 | Rows 1 to 200

Enter expression

	Model	Type	MSRP	Average_MPG	Low...	HighM...
30	E500	Wagon	$60,670	20	0	1
31	Insight 2dr (gas/electric)	Hybrid	$19,110	63	1	0
32	Civic Hybrid 4dr manual (gas/electric)	Hybrid	$20,140	48.5	0	0
33	Prius 4dr (gas/electric)	Hybrid	$20,510	55	0	1
34	Vitara LX	SUV	$17,163	20.5	1	0

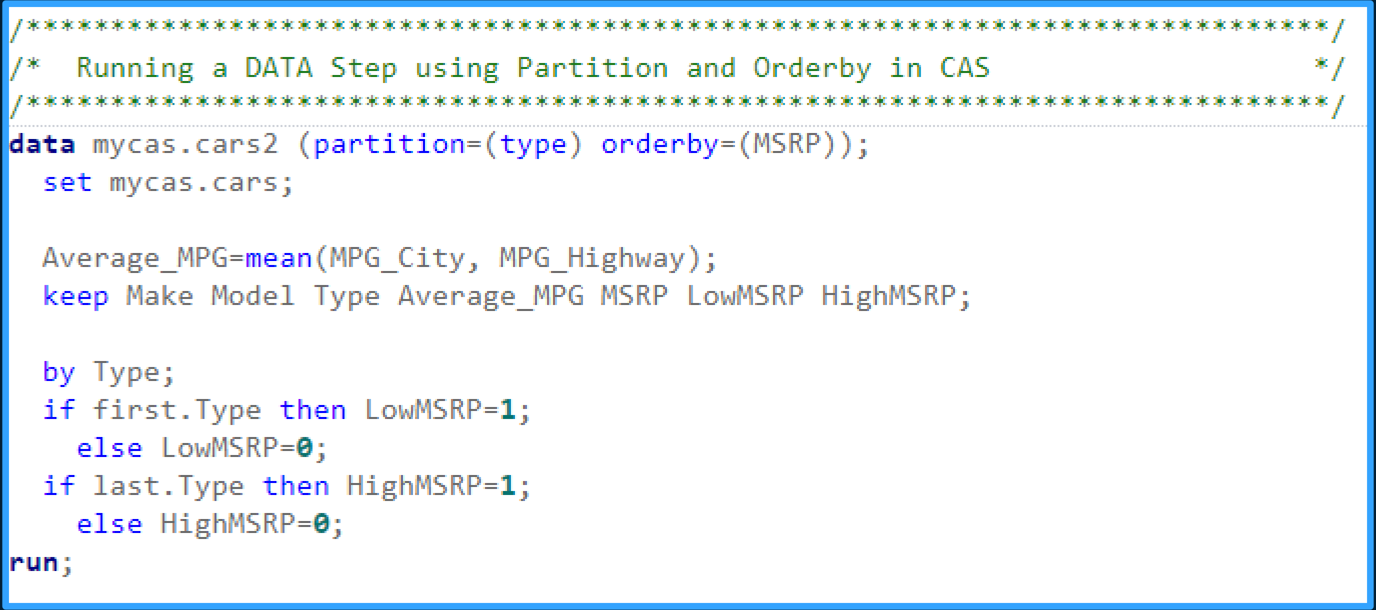


```
data mycas.cars2;
  set mycas.cars;

  Average_MPG=mean(MPG_City, MPG_Highway);
  keep Make Model Type Average_MPG MSRP LowMSRP HighMS

  by Type MSRP;
  if first.Type then LowMSRP=1;
    else LowMSRP=0;
  if last.Type then HighMSRP=1;
    else HighMSRP=0;
run;
 Running DATA step in Cloud Analytic Services.
 The DATA step will run in multiple threads.
 There were 428 observations read from the table CARS in caslib CASUSER(sasdemo).
 The table cars2 in caslib CASUSER(sasdemo) has 428 observations and 7 variables.
 DATA statement used (Total process time):
 real time           0.03 seconds
 cpu time            0.01 seconds
```

ViyaPgm_07 – DATA Step with Group By in CAS.sas

Use DATA Step using Partition & Orderby in CAS

Code

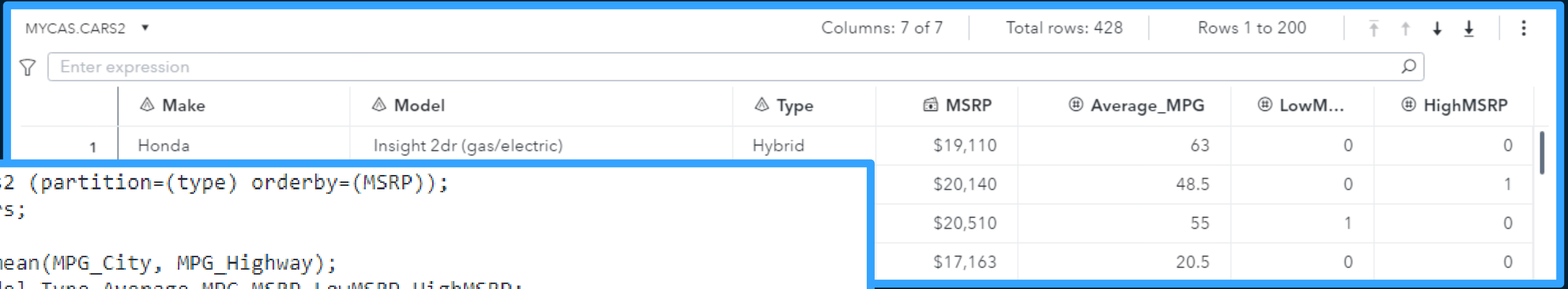


```
/*****************************************************************************/
/*  Running a DATA Step using Partition and Orderby in CAS                   */
/*****************************************************************************/
data mycas.cars2 (partition=(type) orderby=(MSRP));
  set mycas.cars;

  Average_MPG=mean(MPG_City, MPG_Highway);
  keep Make Model Type Average_MPG MSRP LowMSRP HighMSRP;

  by Type;
  if first.Type then LowMSRP=1;
    else LowMSRP=0;
  if last.Type then HighMSRP=1;
    else HighMSRP=0;
run;
```

ViyaPgm_08 – DATA Step with Partition and Order By.sas

Use DATA Step using Partition & Orderby in CAS

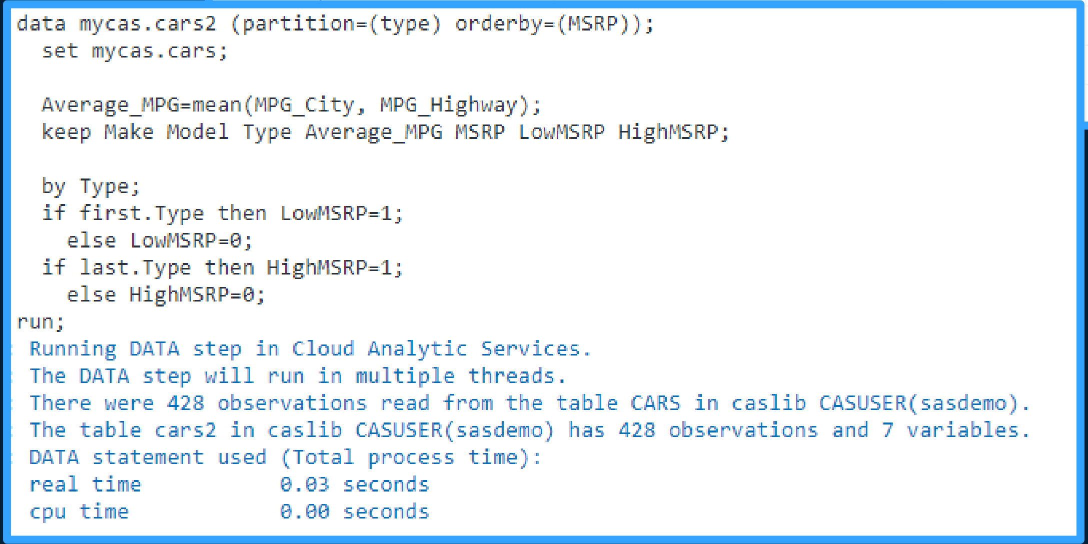
Log & Output



MYCAS.CARS2 ▾ | Columns: 7 of 7 | Total rows: 428 | Rows 1 to 200

Enter expression

	Make	Model	Type	MSRP	Average_MPG	LowM...	HighMSRP
1	Honda	Insight 2dr (gas/electric)	Hybrid	$19,110	63	0	0
				$20,140	48.5	0	1
				$20,510	55	1	0
				$17,163	20.5	0	0

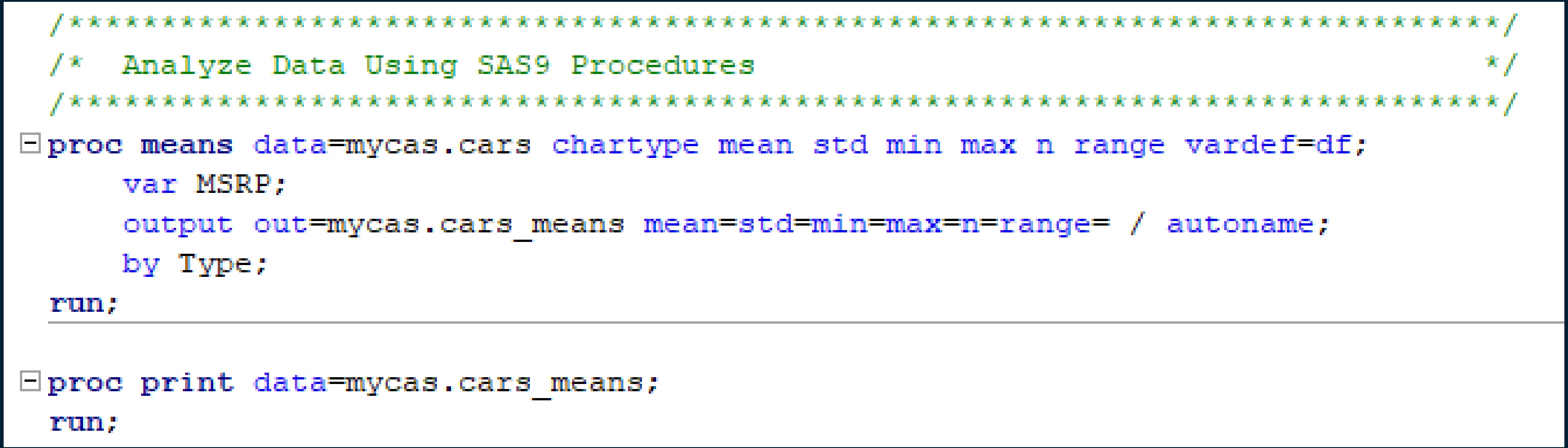


```
data mycas.cars2 (partition=(type) orderby=(MSRP));
  set mycas.cars;

  Average_MPG=mean(MPG_City, MPG_Highway);
  keep Make Model Type Average_MPG MSRP LowMSRP HighMSRP;

  by Type;
  if first.Type then LowMSRP=1;
    else LowMSRP=0;
  if last.Type then HighMSRP=1;
    else HighMSRP=0;
run;
 Running DATA step in Cloud Analytic Services.
 The DATA step will run in multiple threads.
 There were 428 observations read from the table CARS in caslib CASUSER(sasdemo).
 The table cars2 in caslib CASUSER(sasdemo) has 428 observations and 7 variables.
 DATA statement used (Total process time):
 real time           0.03 seconds
 cpu time            0.00 seconds
```

ViyaPgm_08 – DATA Step with Partition and Order By.sas

Analyze Data Using SAS 9 Procedures

Code

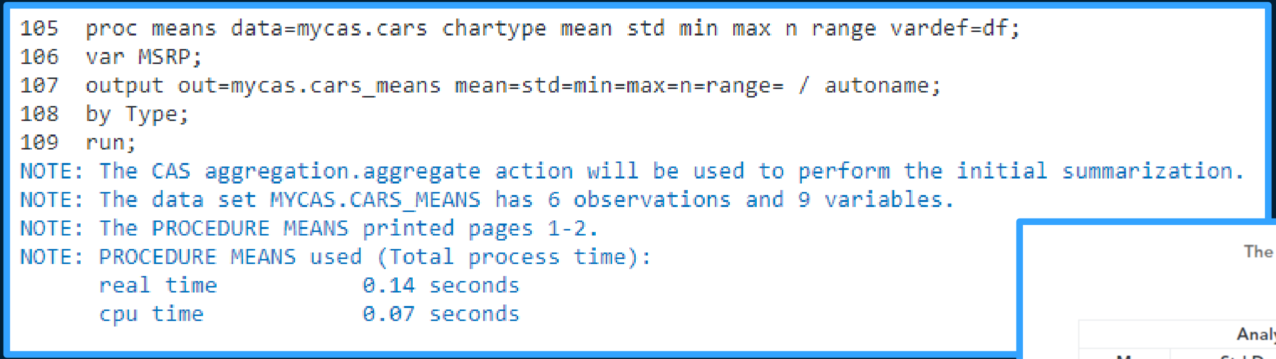


```
/*************************************************************************/
/*  Analyze Data Using SAS9 Procedures                                   */
/*************************************************************************/
proc means data=mycas.cars chartype mean std min max n range vardef=df;
    var MSRP;
    output out=mycas.cars_means mean=std=min=max=n=range= / autoname;
    by Type;
run;

proc print data=mycas.cars_means;
run;
```

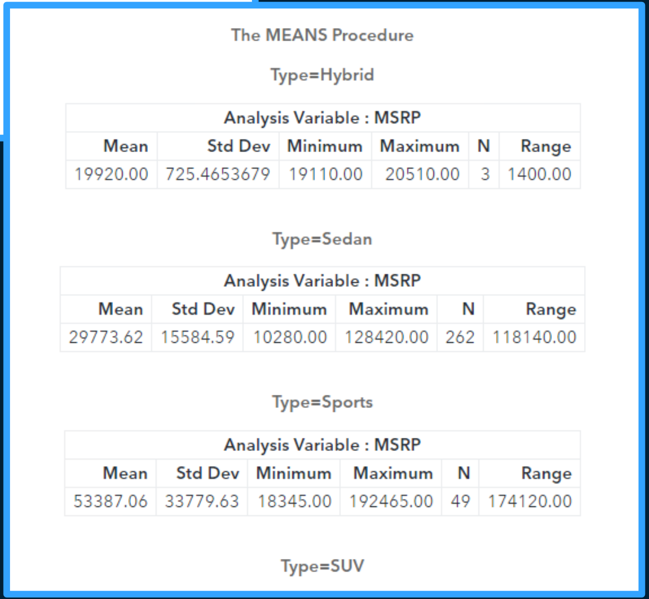
ViyaPgm_09 – Using SAS9 Procs in CAS.sas

Analyze Data Using SAS 9 Procedures

Log & Results



```
proc means data=mycas.cars chartype mean std min max n range vardef=df;
var MSRP;
output out=mycas.cars_means mean=std=min=max=n=range= / autoname;
by Type;
run;
NOTE: The CAS aggregation.aggregate action will be used to perform the initial summarization.
NOTE: The data set MYCAS.CARS_MEANS has 6 observations and 9 variables.
NOTE: The PROCEDURE MEANS printed pages 1-2.
NOTE: PROCEDURE MEANS used (Total process time):
      real time           0.14 seconds
      cpu time            0.07 seconds
```



The MEANS Procedure

Type=Hybrid

Analysis Variable : MSRP					
Mean	Std Dev	Minimum	Maximum	N	Range
19920.00	725.4653679	19110.00	20510.00	3	1400.00

Type=Sedan

Analysis Variable : MSRP					
Mean	Std Dev	Minimum	Maximum	N	Range
29773.62	15584.59	10280.00	128420.00	262	118140.00

Type=Sports

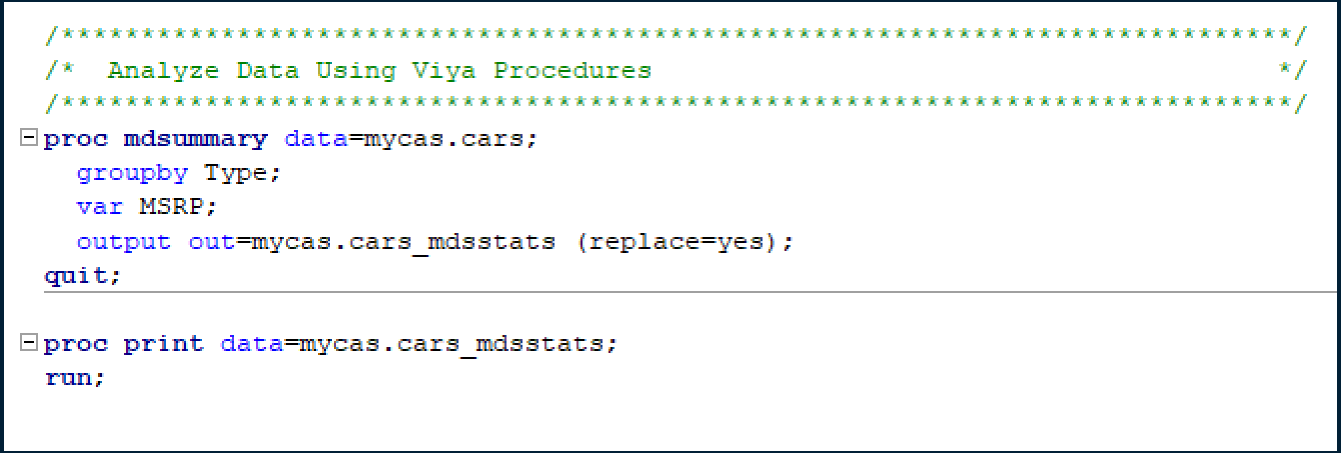
Analysis Variable : MSRP					
Mean	Std Dev	Minimum	Maximum	N	Range
53387.06	33779.63	18345.00	192465.00	49	174120.00

Type=SUV

ViyaPgm_09 – Using SAS9 Procs in CAS.sas

Analyze Data Using Viya Procedures

Code

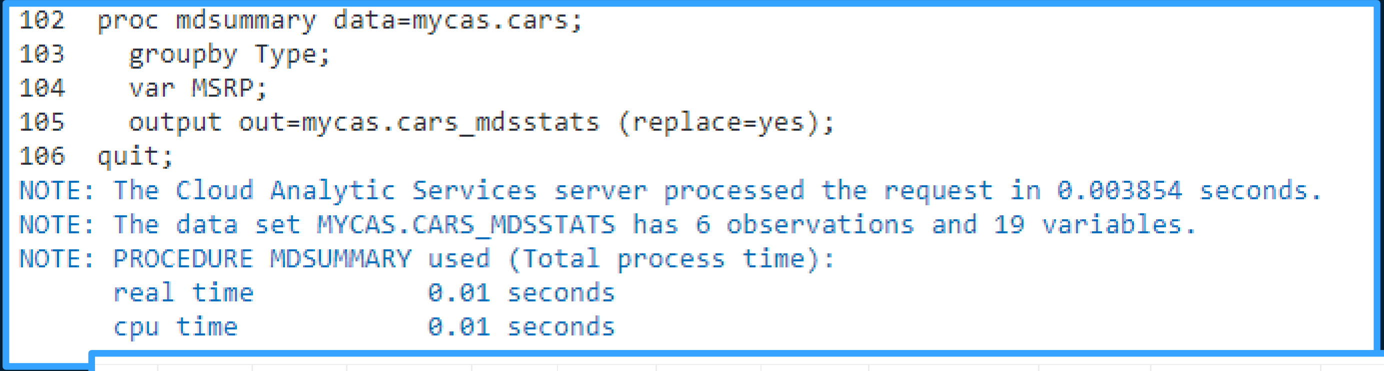


```
/*********************************************************************************/
/*  Analyze Data Using Viya Procedures                                          */
/*********************************************************************************/
proc mdsummary data=mycas.cars;
  groupby Type;
  var MSRP;
  output out=mycas.cars_mdsstats (replace=yes);
quit;

proc print data=mycas.cars_mdsstats;
run;
```

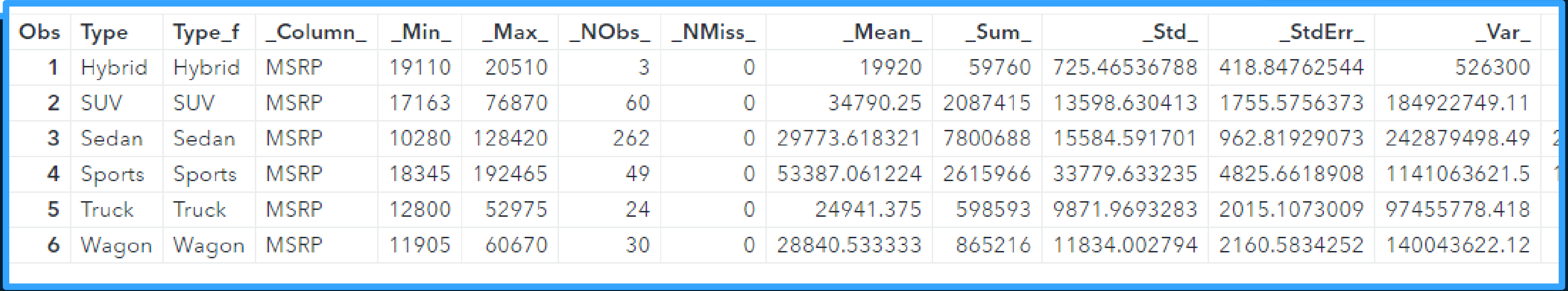
ViyaPgm_10 – Using Viya Procs.sas

Analyze Data Using Viya Procedures

Log & Results



```
proc mdsummary data=mycas.cars;
   groupby Type;
   var MSRP;
   output out=mycas.cars_mdsstats (replace=yes);
quit;
NOTE: The Cloud Analytic Services server processed the request in 0.003854 seconds.
NOTE: The data set MYCAS.CARS_MDSSTATS has 6 observations and 19 variables.
NOTE: PROCEDURE MDSUMMARY used (Total process time):
      real time           0.01 seconds
      cpu time            0.01 seconds
```

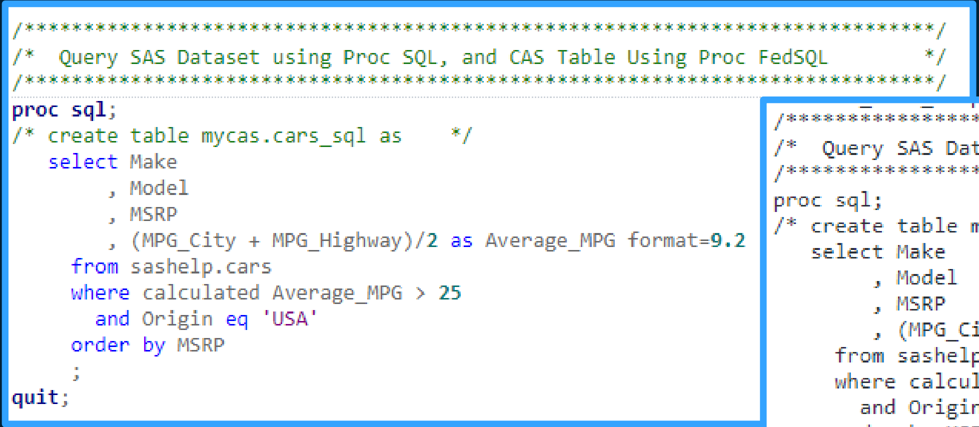


Obs	Type	Type_f	_Column_	_Min_	_Max_	_NObs_	_NMiss_	_Mean_	_Sum_	_Std_	_StdErr_	_Var_
1	Hybrid	Hybrid	MSRP	19110	20510	3	0	19920	59760	725.46536788	418.84762544	526300
2	SUV	SUV	MSRP	17163	76870	60	0	34790.25	2087415	13598.630413	1755.5756373	184922749.11
3	Sedan	Sedan	MSRP	10280	128420	262	0	29773.618321	7800688	15584.591701	962.81929073	242879498.49
4	Sports	Sports	MSRP	18345	192465	49	0	53387.061224	2615966	33779.633235	4825.6618908	1141063621.5
5	Truck	Truck	MSRP	12800	52975	24	0	24941.375	598593	9871.9693283	2015.1073009	97455778.418
6	Wagon	Wagon	MSRP	11905	60670	30	0	28840.533333	865216	11834.002794	2160.5834252	140043622.12

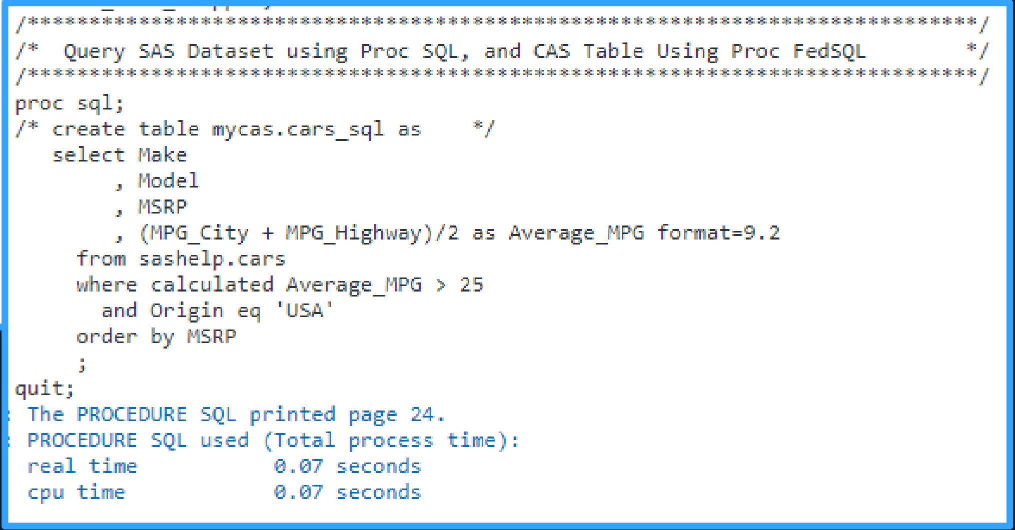
ViyaPgm_10 – Using Viya Procs.sas

Query a SAS Dataset Using Proc SQL – Compute Server

Code & Log



```
/*************************************************************************/
/*  Query SAS Dataset using Proc SQL, and CAS Table Using Proc FedSQL      */
/*************************************************************************/
proc sql;
/* create table mycas.cars_sql as    */
   select Make
        , Model
        , MSRP
        , (MPG_City + MPG_Highway)/2 as Average_MPG format=9.2
     from sashelp.cars
     where calculated Average_MPG > 25
       and Origin eq 'USA'
     order by MSRP
     ;
quit;
```

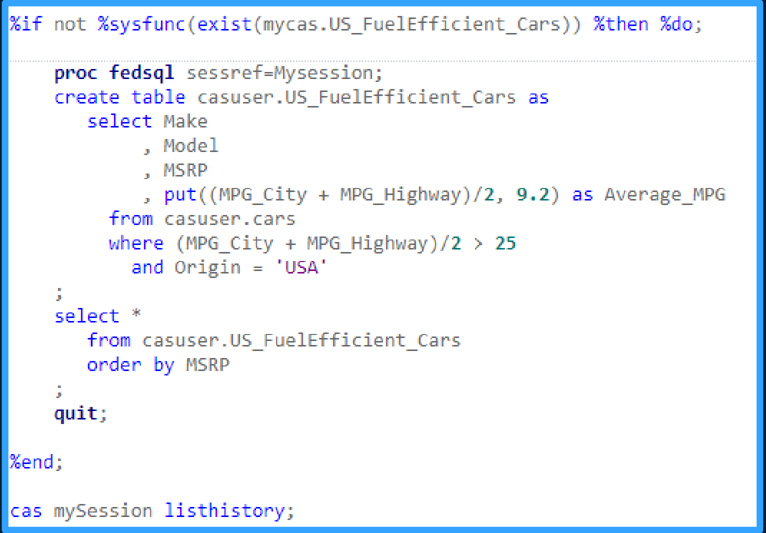


```
/*************************************************************************/
/*  Query SAS Dataset using Proc SQL, and CAS Table Using Proc FedSQL      */
/*************************************************************************/
proc sql;
/* create table mycas.cars_sql as    */
   select Make
        , Model
        , MSRP
        , (MPG_City + MPG_Highway)/2 as Average_MPG format=9.2
     from sashelp.cars
     where calculated Average_MPG > 25
       and Origin eq 'USA'
     order by MSRP
     ;
quit;
 The PROCEDURE SQL printed page 24.
 PROCEDURE SQL used (Total process time):
 real time           0.07 seconds
 cpu time            0.07 seconds
```

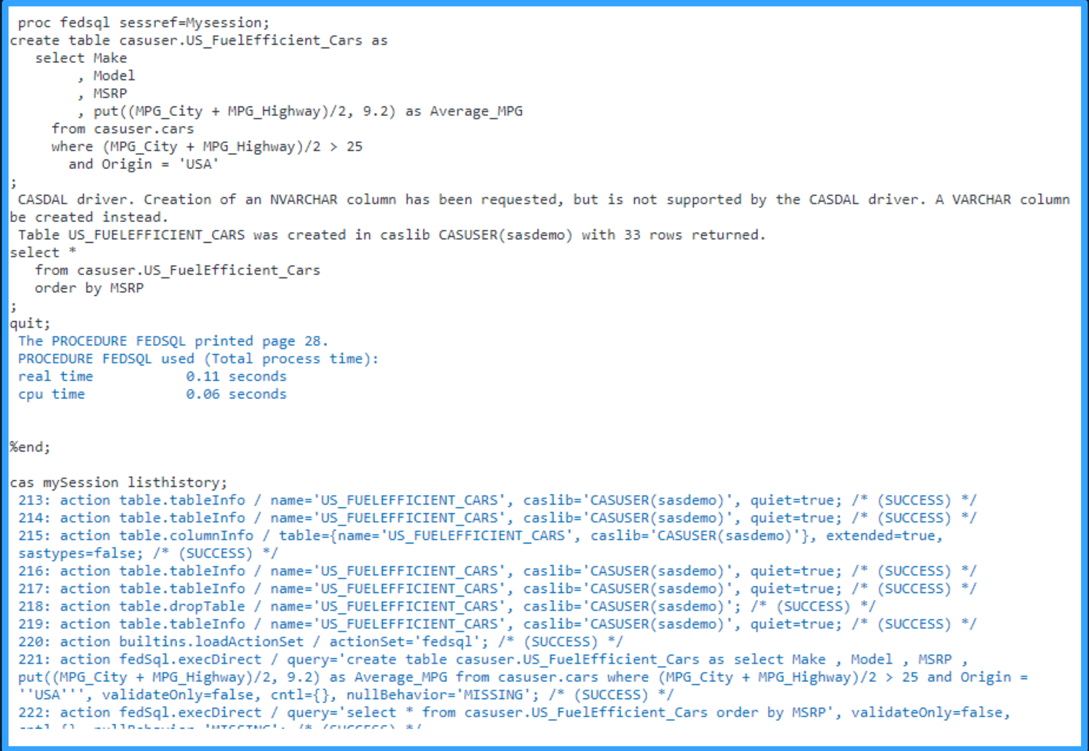
ViyaPgm_11 – Proc SQL & FedSQL.sas

Query a CAS Table Using Proc FedSQL – CAS

Code & Log (with List History)



```
%if not %sysfunc(exist(mycas.US_FuelEfficient_Cars)) %then %do;

    proc fedsql sessref=Mysession;
    create table casuser.US_FuelEfficient_Cars as
       select Make
            , Model
            , MSRP
            , put((MPG_City + MPG_Highway)/2, 9.2) as Average_MPG
         from casuser.cars
         where (MPG_City + MPG_Highway)/2 > 25
           and Origin = 'USA'
    ;
    select *
       from casuser.US_FuelEfficient_Cars
       order by MSRP
    ;
    quit;

%end;

cas mySession listhistory;
```

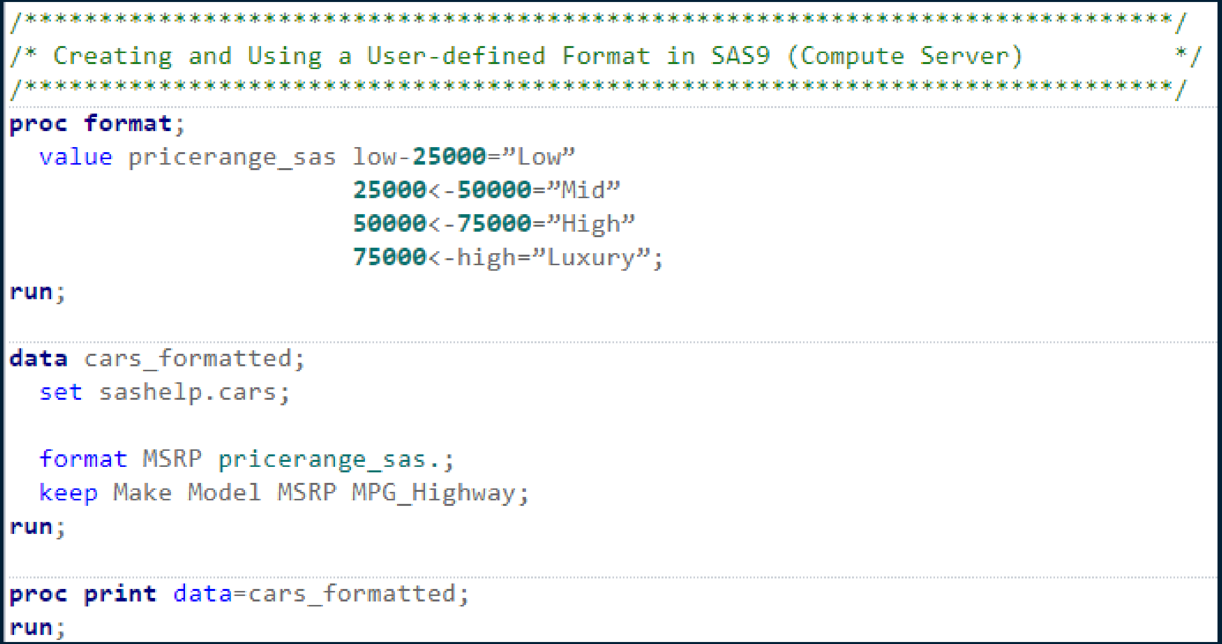


```
 proc fedsql sessref=Mysession;
create table casuser.US_FuelEfficient_Cars as
   select Make
        , Model
        , MSRP
        , put((MPG_City + MPG_Highway)/2, 9.2) as Average_MPG
     from casuser.cars
     where (MPG_City + MPG_Highway)/2 > 25
       and Origin = 'USA'
;
 CASDAL driver. Creation of an NVARCHAR column has been requested, but is not supported by the CASDAL driver. A VARCHAR column
be created instead.
 Table US_FUELEFFICIENT_CARS was created in caslib CASUSER(sasdemo) with 33 rows returned.
select *
   from casuser.US_FuelEfficient_Cars
   order by MSRP
;
quit;
 The PROCEDURE FEDSQL printed page 28.
 PROCEDURE FEDSQL used (Total process time):
 real time           0.11 seconds
 cpu time            0.06 seconds


%end;

cas mySession listhistory;
 213: action table.tableInfo / name='US_FUELEFFICIENT_CARS', caslib='CASUSER(sasdemo)', quiet=true; /* (SUCCESS) */
 214: action table.tableInfo / name='US_FUELEFFICIENT_CARS', caslib='CASUSER(sasdemo)', quiet=true; /* (SUCCESS) */
 215: action table.columnInfo / table={name='US_FUELEFFICIENT_CARS', caslib='CASUSER(sasdemo)'}, extended=true,
 sastypes=false; /* (SUCCESS) */
 216: action table.tableInfo / name='US_FUELEFFICIENT_CARS', caslib='CASUSER(sasdemo)', quiet=true; /* (SUCCESS) */
 217: action table.tableInfo / name='US_FUELEFFICIENT_CARS', caslib='CASUSER(sasdemo)', quiet=true; /* (SUCCESS) */
 218: action table.dropTable / name='US_FUELEFFICIENT_CARS', caslib='CASUSER(sasdemo)'; /* (SUCCESS) */
 219: action table.tableInfo / name='US_FUELEFFICIENT_CARS', caslib='CASUSER(sasdemo)', quiet=true; /* (SUCCESS) */
 220: action builtins.loadActionSet / actionSet='fedsql'; /* (SUCCESS) */
 221: action fedSql.execDirect / query='create table casuser.US_FuelEfficient_Cars as select Make , Model , MSRP ,
 put((MPG_City + MPG_Highway)/2, 9.2) as Average_MPG from casuser.cars where (MPG_City + MPG_Highway)/2 > 25 and Origin =
 ''USA''', validateOnly=false, cntl={}, nullBehavior='MISSING'; /* (SUCCESS) */
 222: action fedSql.execDirect / query='select * from casuser.US_FuelEfficient_Cars order by MSRP', validateOnly=false,
```

ViyaPgm_11 – Proc SQL & FedSQL.sas

Formats on the Compute Server

Code

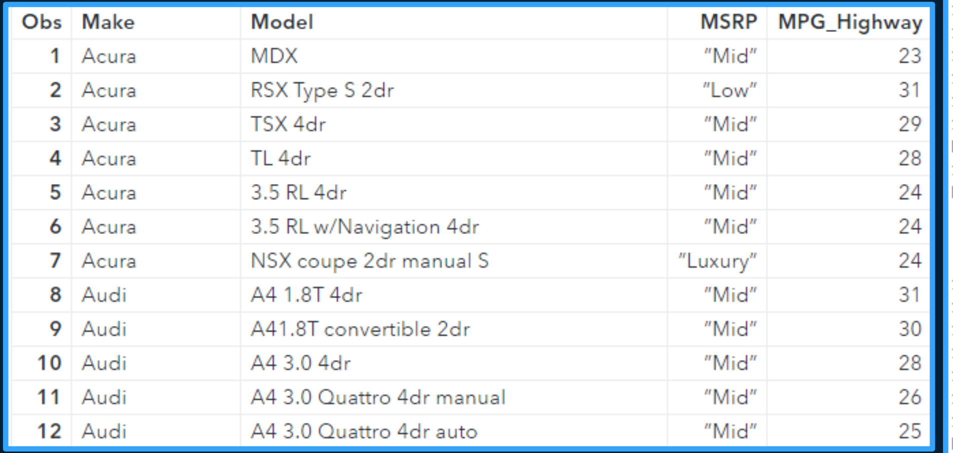


```
/*************************************************************************/
/* Creating and Using a User-defined Format in SAS9 (Compute Server)    */
/*************************************************************************/
proc format;
  value pricerange_sas low-25000="Low"
                       25000<-50000="Mid"
                       50000<-75000="High"
                       75000<-high="Luxury";
run;

data cars_formatted;
  set sashelp.cars;

  format MSRP pricerange_sas.;
  keep Make Model MSRP MPG_Highway;
run;

proc print data=cars_formatted;
run;
```

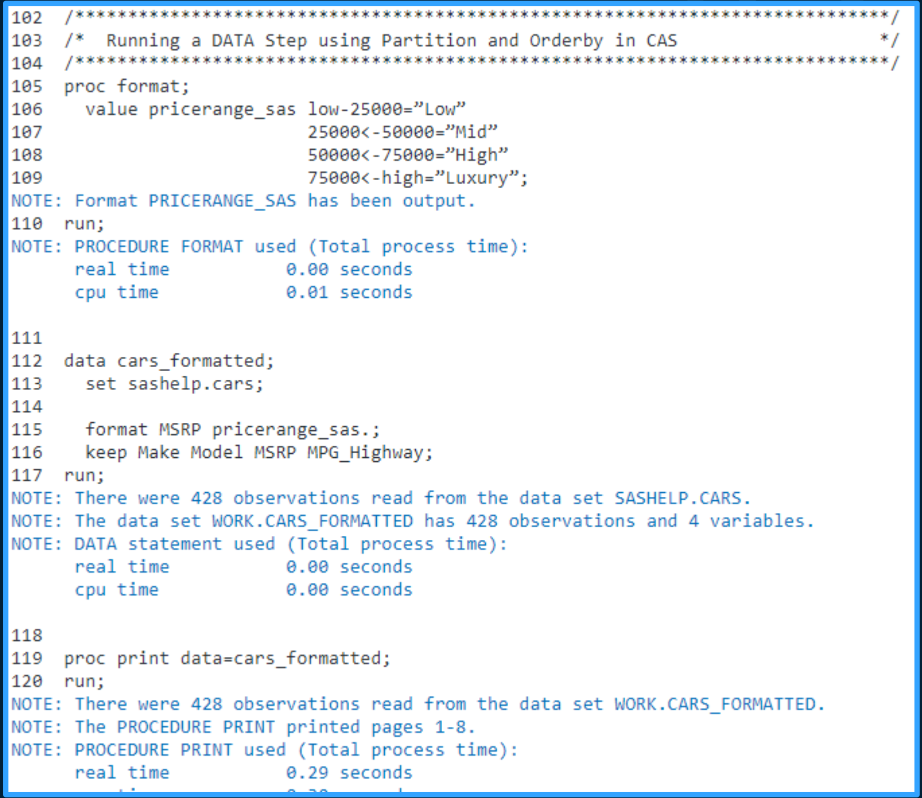
ViyaPgm_12 – User-Defined Formats in CAS.sas

Formats on the Compute Server

Log & Results



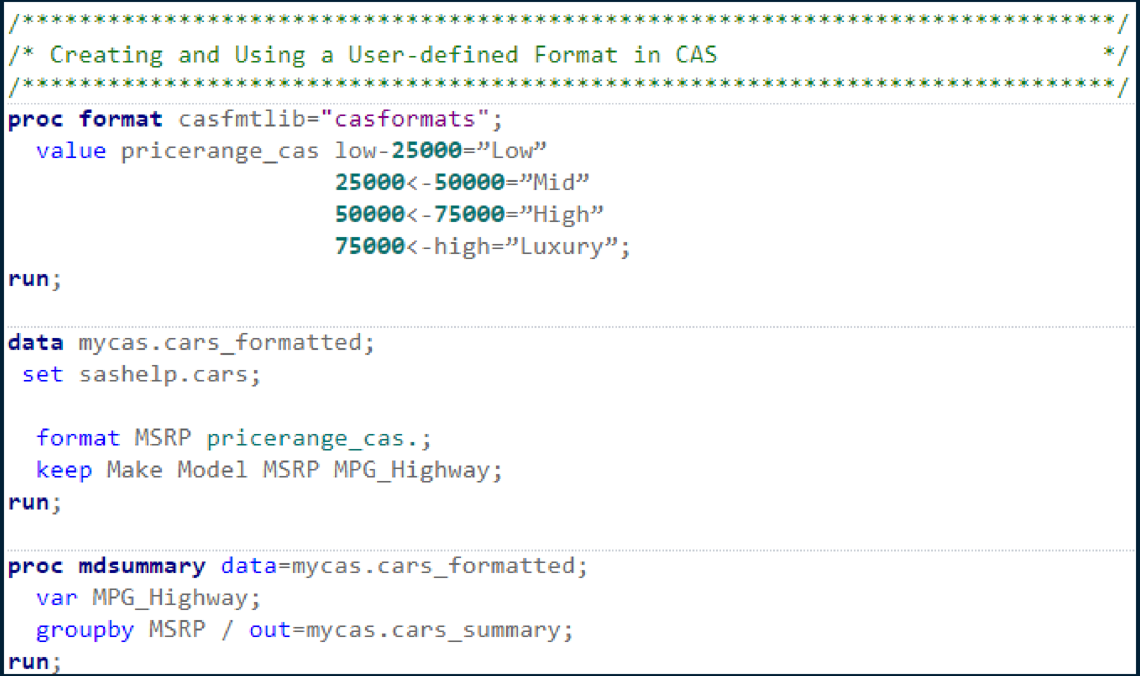
Obs	Make	Model	MSRP	MPG_Highway
1	Acura	MDX	"Mid"	23
2	Acura	RSX Type S 2dr	"Low"	31
3	Acura	TSX 4dr	"Mid"	29
4	Acura	TL 4dr	"Mid"	28
5	Acura	3.5 RL 4dr	"Mid"	24
6	Acura	3.5 RL w/Navigation 4dr	"Mid"	24
7	Acura	NSX coupe 2dr manual S	"Luxury"	24
8	Audi	A4 1.8T 4dr	"Mid"	31
9	Audi	A41.8T convertible 2dr	"Mid"	30
10	Audi	A4 3.0 4dr	"Mid"	28
11	Audi	A4 3.0 Quattro 4dr manual	"Mid"	26
12	Audi	A4 3.0 Quattro 4dr auto	"Mid"	25



```
102  /*************************************************************************/
103  /*  Running a DATA Step using Partition and Orderby in CAS               */
104  /*************************************************************************/
105  proc format;
106    value pricerange_sas low-25000="Low"
107                         25000<-50000="Mid"
108                         50000<-75000="High"
109                         75000<-high="Luxury";
NOTE: Format PRICERANGE_SAS has been output.
110  run;
NOTE: PROCEDURE FORMAT used (Total process time):
      real time           0.00 seconds
      cpu time            0.01 seconds


111
112  data cars_formatted;
113    set sashelp.cars;
114
115    format MSRP pricerange_sas.;
116    keep Make Model MSRP MPG_Highway;
117  run;
NOTE: There were 428 observations read from the data set SASHELP.CARS.
NOTE: The data set WORK.CARS_FORMATTED has 428 observations and 4 variables.
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds


118
119  proc print data=cars_formatted;
120  run;
NOTE: There were 428 observations read from the data set WORK.CARS_FORMATTED.
NOTE: The PROCEDURE PRINT printed pages 1-8.
NOTE: PROCEDURE PRINT used (Total process time):
      real time           0.29 seconds
```

ViyaPgm_12 – User-Defined Formats in CAS.sas

Formats in CAS
Code

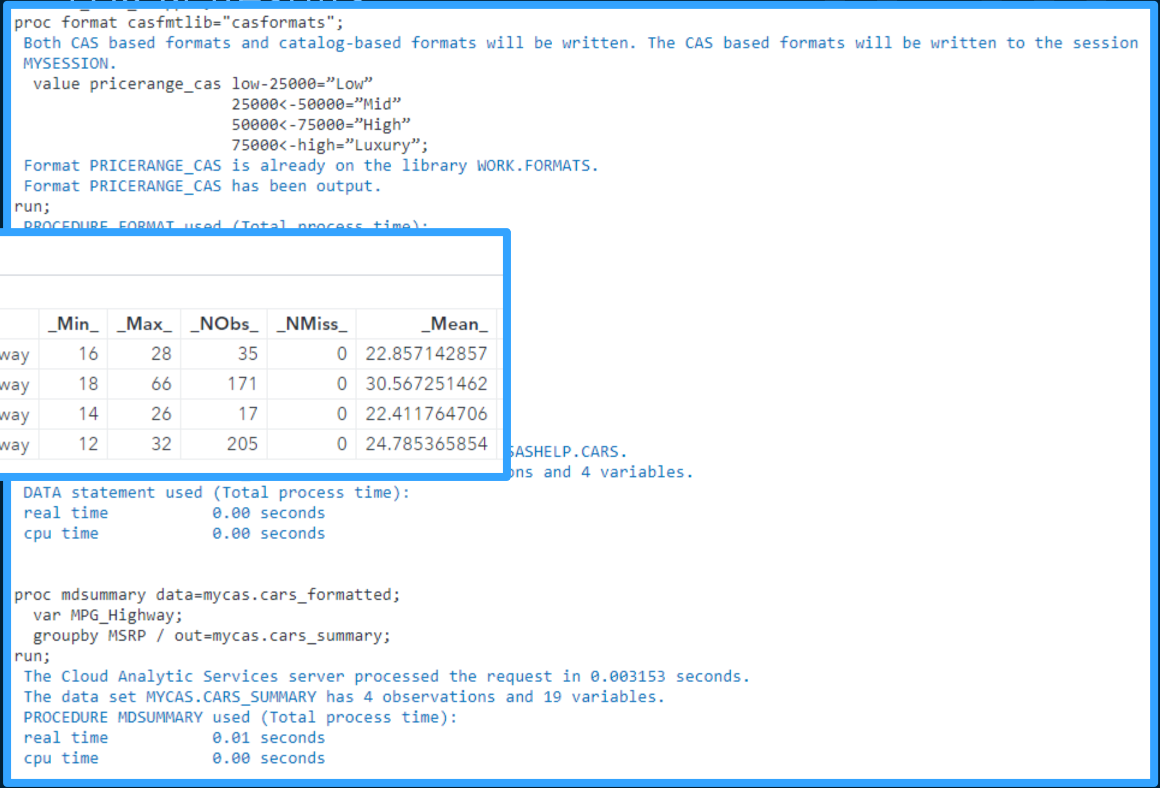


```
/******************************************************************************/
/* Creating and Using a User-defined Format in CAS                            */
/******************************************************************************/
proc format casfmtlib="casformats";
  value pricerange_cas low-25000=”Low”
                       25000<-50000=”Mid”
                       50000<-75000=”High”
                       75000<-high=”Luxury”;
run;

data mycas.cars_formatted;
 set sashelp.cars;

  format MSRP pricerange_cas.;
  keep Make Model MSRP MPG_Highway;
run;

proc mdsummary data=mycas.cars_formatted;
  var MPG_Highway;
  groupby MSRP / out=mycas.cars_summary;
run;
```

ViyaPgm_12 – User-Defined Formats in CAS.sas

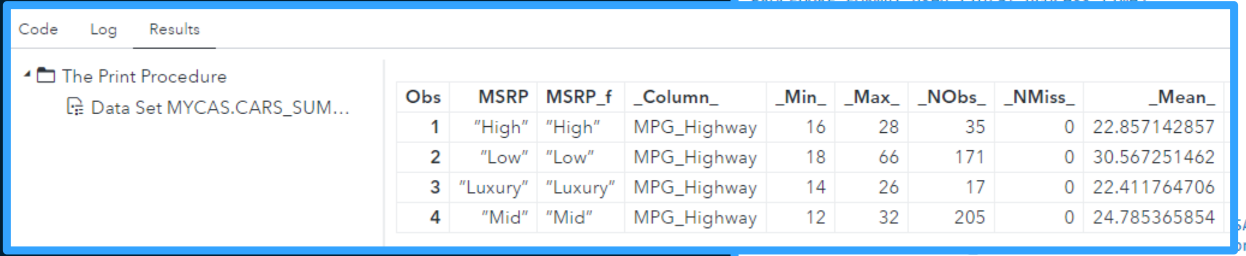
Formats in CAS

Log & Results



```
proc format casfmtlib="casformats";
 Both CAS based formats and catalog-based formats will be written. The CAS based formats will be written to the session
 MYSESSION.
  value pricerange_cas low-25000="Low"
                       25000<-50000="Mid"
                       50000<-75000="High"
                       75000<-high="Luxury";
 Format PRICERANGE_CAS is already on the library WORK.FORMATS.
 Format PRICERANGE_CAS has been output.
run;
 PROCEDURE FORMAT used (Total process time):
```

```
                                                     SASHELP.CARS.
                                                     ons and 4 variables.
 DATA statement used (Total process time):
 real time           0.00 seconds
 cpu time            0.00 seconds


proc mdsummary data=mycas.cars_formatted;
  var MPG_Highway;
  groupby MSRP / out=mycas.cars_summary;
run;
 The Cloud Analytic Services server processed the request in 0.003153 seconds.
 The data set MYCAS.CARS_SUMMARY has 4 observations and 19 variables.
 PROCEDURE MDSUMMARY used (Total process time):
 real time           0.01 seconds
 cpu time            0.00 seconds
```

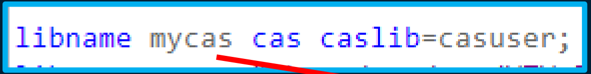


Obs	MSRP	MSRP_f	_Column_	_Min_	_Max_	_NObs_	_NMiss_	_Mean_
1	"High"	"High"	MPG_Highway	16	28	35	0	22.857142857
2	"Low"	"Low"	MPG_Highway	18	66	171	0	30.567251462
3	"Luxury"	"Luxury"	MPG_Highway	14	26	17	0	22.411764706
4	"Mid"	"Mid"	MPG_Highway	12	32	205	0	24.785365854

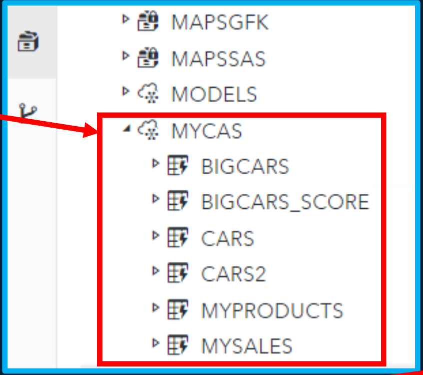
ViyaPgm_12 – User-Defined Formats in CAS.sas

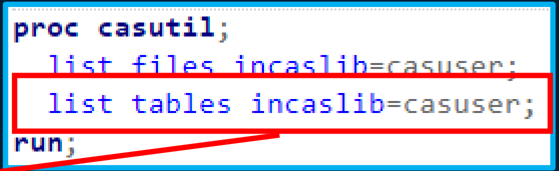
Clean Up

List Files and in-Memory Tables Associated with CASLIB = CASUSER

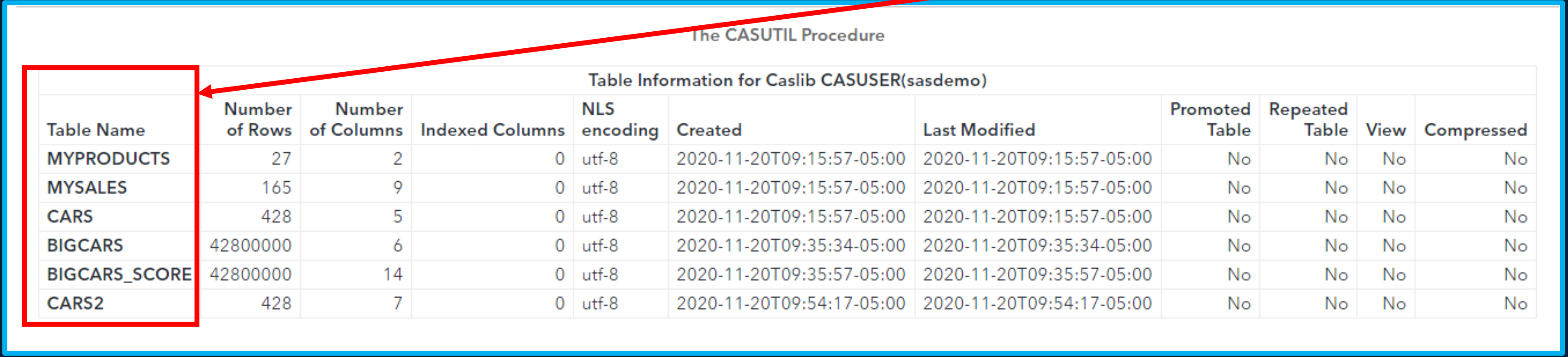


```
libname mycas cas caslib=casuser;
```





```
proc casutil;
   list files incaslib=casuser;
   list tables incaslib=casuser;
run;
```



The CASUTIL Procedure

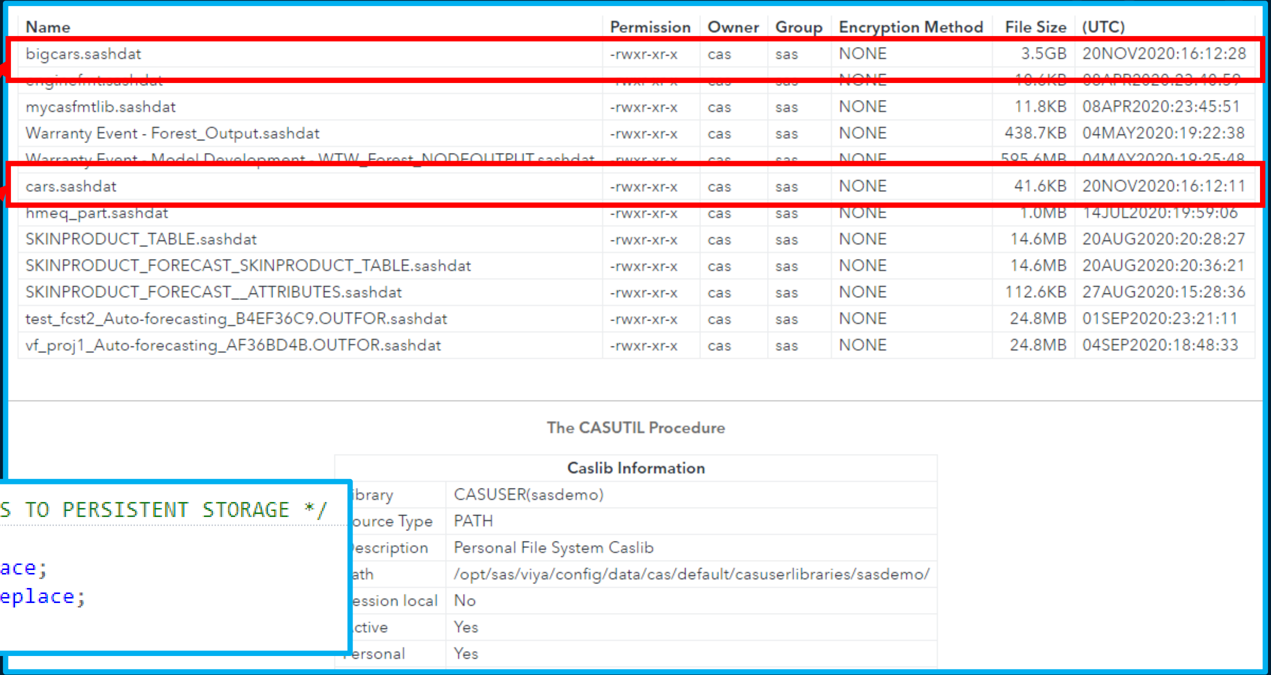
Table Information for Caslib CASUSER(sasdemo)

Table Name	Number of Rows	Number of Columns	Indexed Columns	NLS encoding	Created	Last Modified	Promoted Table	Repeated Table	View	Compressed
MYPRODUCTS	27	2	0	utf-8	2020-11-20T09:15:57-05:00	2020-11-20T09:15:57-05:00	No	No	No	No
MYSALES	165	9	0	utf-8	2020-11-20T09:15:57-05:00	2020-11-20T09:15:57-05:00	No	No	No	No
CARS	428	5	0	utf-8	2020-11-20T09:15:57-05:00	2020-11-20T09:15:57-05:00	No	No	No	No
BIGCARS	42800000	6	0	utf-8	2020-11-20T09:35:34-05:00	2020-11-20T09:35:34-05:00	No	No	No	No
BIGCARS_SCORE	42800000	14	0	utf-8	2020-11-20T09:35:57-05:00	2020-11-20T09:35:57-05:00	No	No	No	No
CARS2	428	7	0	utf-8	2020-11-20T09:54:17-05:00	2020-11-20T09:54:17-05:00	No	No	No	No

ViyaPgm_13 – Cleaning Up CAS.sas

Clean Up

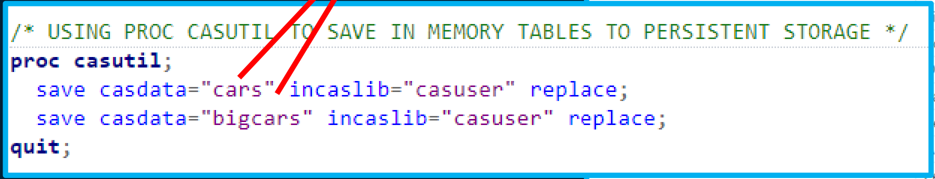
Save In-Memory Tables in CASLIB = CASUSER



Name	Permission	Owner	Group	Encryption Method	File Size	(UTC)
bigcars.sashdat	-rwxr-xr-x	cas	sas	NONE	3.5GB	20NOV2020:16:12:28
enginefmt.sashdat	-rwxr-xr-x	cas	sas	NONE	10.6KB	08APR2020:23:40:59
mycasfmtlib.sashdat	-rwxr-xr-x	cas	sas	NONE	11.8KB	08APR2020:23:45:51
Warranty Event - Forest_Output.sashdat	-rwxr-xr-x	cas	sas	NONE	438.7KB	04MAY2020:19:22:38
Warranty Event - Model Development - WTW_Forest_NODEOUTPUT.sashdat	-rwxr-xr-x	cas	sas	NONE	595.6MB	04MAY2020:19:25:48
cars.sashdat	-rwxr-xr-x	cas	sas	NONE	41.6KB	20NOV2020:16:12:11
hmeq_part.sashdat	-rwxr-xr-x	cas	sas	NONE	1.0MB	14JUL2020:19:59:06
SKINPRODUCT_TABLE.sashdat	-rwxr-xr-x	cas	sas	NONE	14.6MB	20AUG2020:20:28:27
SKINPRODUCT_FORECAST_SKINPRODUCT_TABLE.sashdat	-rwxr-xr-x	cas	sas	NONE	14.6MB	20AUG2020:20:36:21
SKINPRODUCT_FORECAST__ATTRIBUTES.sashdat	-rwxr-xr-x	cas	sas	NONE	112.6KB	27AUG2020:15:28:36
test_fcst2_Auto-forecasting_B4EF36C9.OUTFOR.sashdat	-rwxr-xr-x	cas	sas	NONE	24.8MB	01SEP2020:23:21:11
vf_proj1_Auto-forecasting_AF36BD4B.OUTFOR.sashdat	-rwxr-xr-x	cas	sas	NONE	24.8MB	04SEP2020:18:48:33

The CASUTIL Procedure

Caslib Information	
ibrary	CASUSER(sasdemo)
ource Type	PATH
escription	Personal File System Caslib
ath	/opt/sas/viya/config/data/cas/default/casuserlibraries/sasdemo/
ession local	No
ctive	Yes
ersonal	Yes

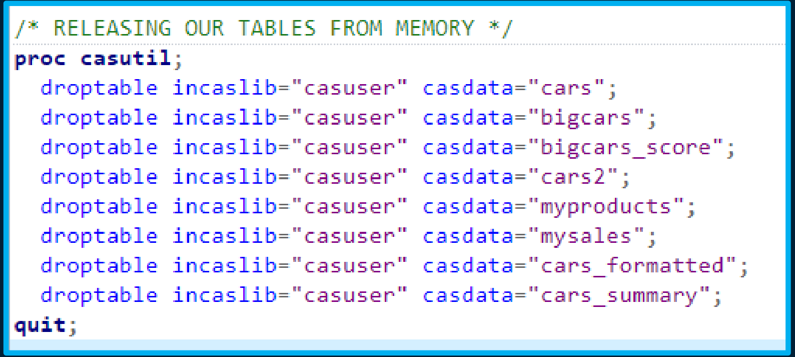


```
/* USING PROC CASUTIL TO SAVE IN MEMORY TABLES TO PERSISTENT STORAGE */
proc casutil;
  save casdata="cars" incaslib="casuser" replace;
  save casdata="bigcars" incaslib="casuser" replace;
quit;
```

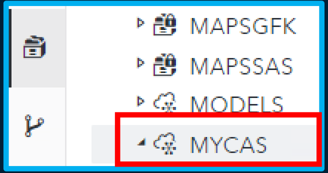
ViyaPgm_13 – Cleaning Up CAS.sas

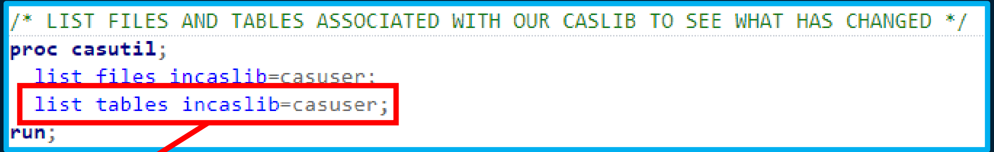
Clean Up

Drop Tables from CAS Memory Associated with CASLIB = CASUSER

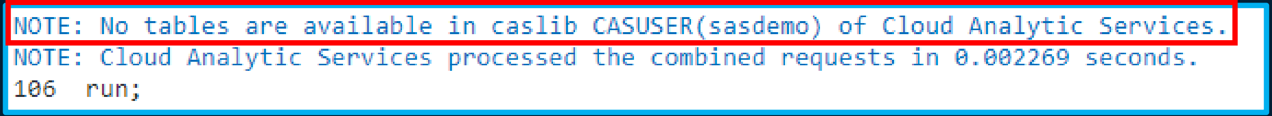


```
/* RELEASING OUR TABLES FROM MEMORY */
proc casutil;
  droptable incaslib="casuser" casdata="cars";
  droptable incaslib="casuser" casdata="bigcars";
  droptable incaslib="casuser" casdata="bigcars_score";
  droptable incaslib="casuser" casdata="cars2";
  droptable incaslib="casuser" casdata="myproducts";
  droptable incaslib="casuser" casdata="mysales";
  droptable incaslib="casuser" casdata="cars_formatted";
  droptable incaslib="casuser" casdata="cars_summary";
quit;
```





```
/* LIST FILES AND TABLES ASSOCIATED WITH OUR CASLIB TO SEE WHAT HAS CHANGED */
proc casutil;
  list files incaslib=casuser;
  list tables incaslib=casuser;
run;
```

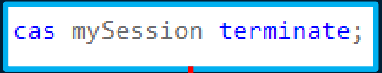


```
NOTE: No tables are available in caslib CASUSER(sasdemo) of Cloud Analytic Services.
NOTE: Cloud Analytic Services processed the combined requests in 0.002269 seconds.
106  run;
```

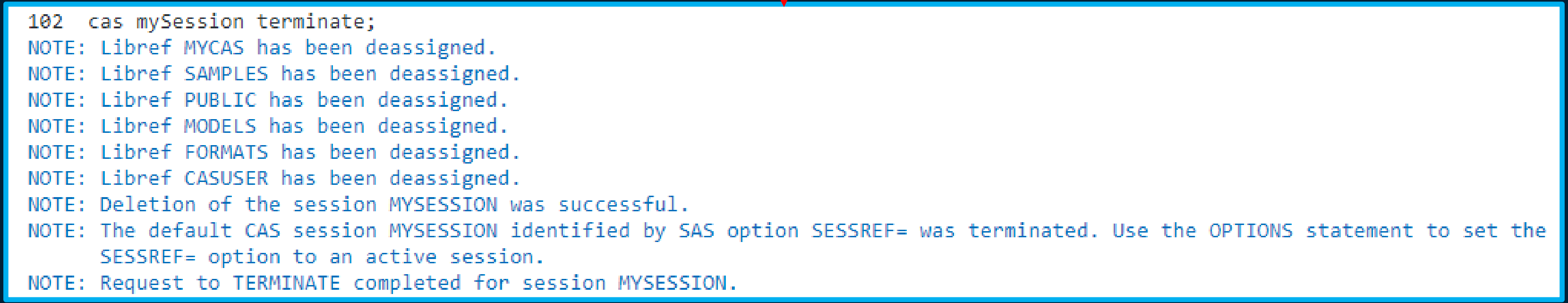
ViyaPgm_13 – Cleaning Up CAS.sas

Clean Up

Terminate CAS Session – Code & Log



```
cas mySession terminate;
```



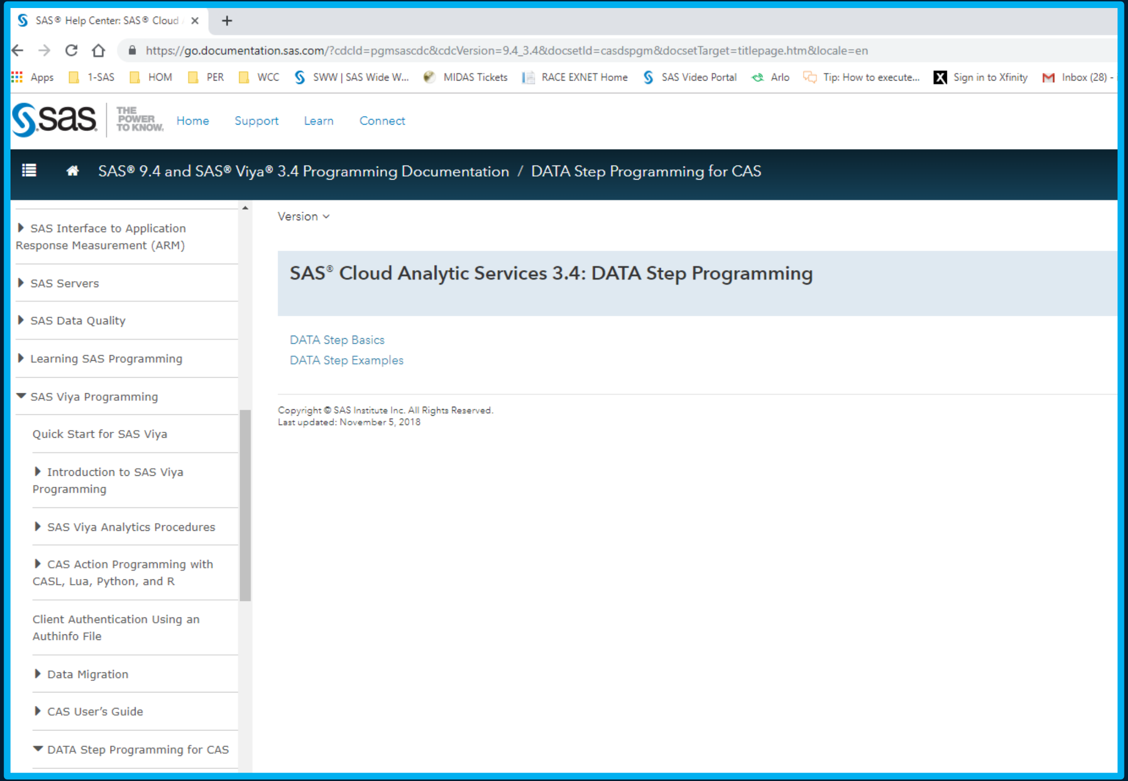
```
cas mySession terminate;
NOTE: Libref MYCAS has been deassigned.
NOTE: Libref SAMPLES has been deassigned.
NOTE: Libref PUBLIC has been deassigned.
NOTE: Libref MODELS has been deassigned.
NOTE: Libref FORMATS has been deassigned.
NOTE: Libref CASUSER has been deassigned.
NOTE: Deletion of the session MYSESSION was successful.
NOTE: The default CAS session MYSESSION identified by SAS option SESSREF= was terminated. Use the OPTIONS statement to set the
      SESSREF= option to an active session.
NOTE: Request to TERMINATE completed for session MYSESSION.
```

ViyaPgm_13 – Cleaning Up CAS.sas

References and Resources

sas.com

Documentation



SAS® Help Center: SAS® Cloud

https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.4&docsetId=casdspgm&docsetTarget=titlepage.htm&locale=en

SAS® 9.4 and SAS® Viya® 3.4 Programming Documentation / DATA Step Programming for CAS

- SAS Interface to Application Response Measurement (ARM)
- SAS Servers
- SAS Data Quality
- Learning SAS Programming
- SAS Viya Programming
  - Quick Start for SAS Viya
  - Introduction to SAS Viya Programming
  - SAS Viya Analytics Procedures
  - CAS Action Programming with CASL, Lua, Python, and R
  - Client Authentication Using an Authinfo File
  - Data Migration
  - CAS User's Guide
  - DATA Step Programming for CAS

Version

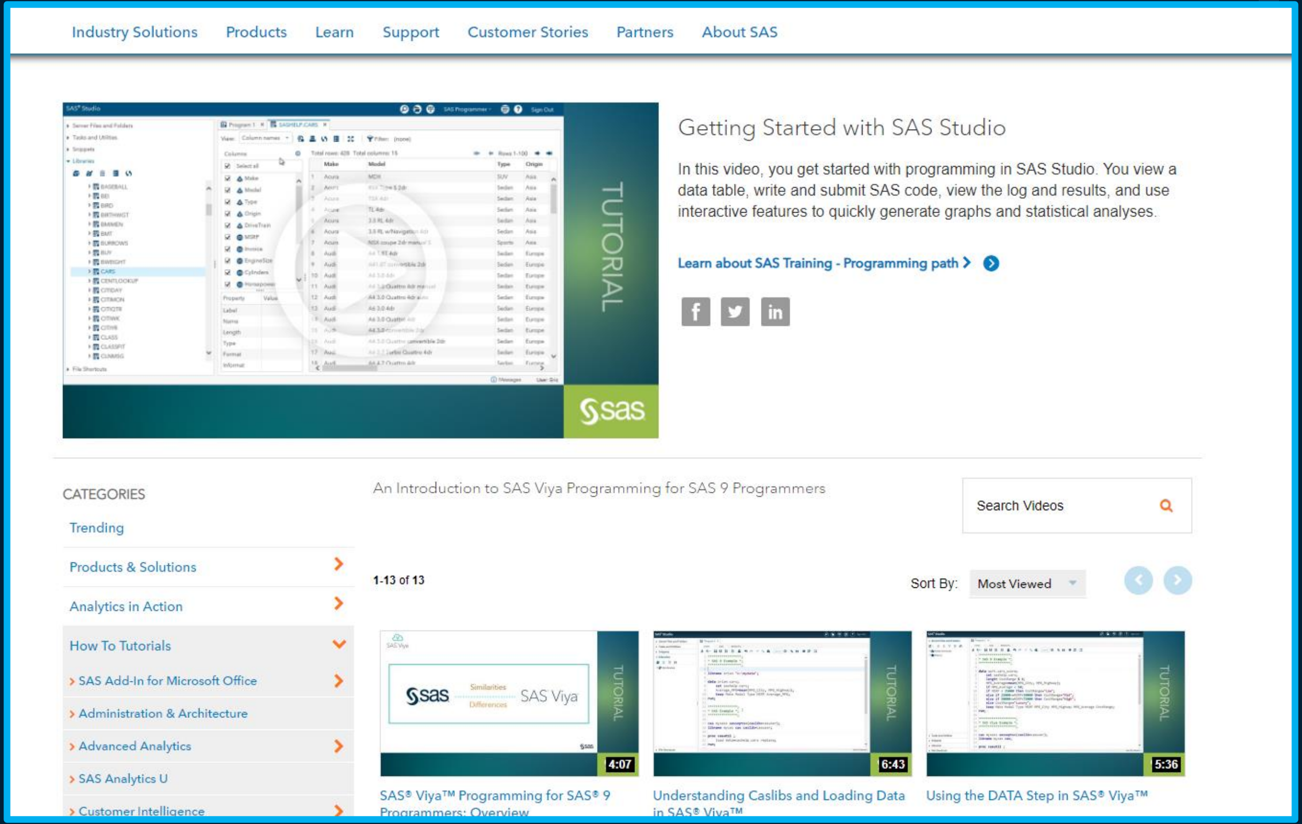
SAS® Cloud Analytic Services 3.4: DATA Step Programming

DATA Step Basics
DATA Step Examples

Copyright © SAS Institute Inc. All Rights Reserved.
Last updated: November 5, 2018

[Viya Programming Documentation]

Video Tutorials



Industry Solutions | Products | Learn | Support | Customer Stories | Partners | About SAS

Getting Started with SAS Studio

In this video, you get started with programming in SAS Studio. You view a data table, write and submit SAS code, view the log and results, and use interactive features to quickly generate graphs and statistical analyses.

Learn about SAS Training - Programming path >

CATEGORIES

- Trending
- Products & Solutions
- Analytics in Action
- How To Tutorials
  - SAS Add-In for Microsoft Office
  - Administration & Architecture
  - Advanced Analytics
  - SAS Analytics U
  - Customer Intelligence

An Introduction to SAS Viya Programming for SAS 9 Programmers

Search Videos

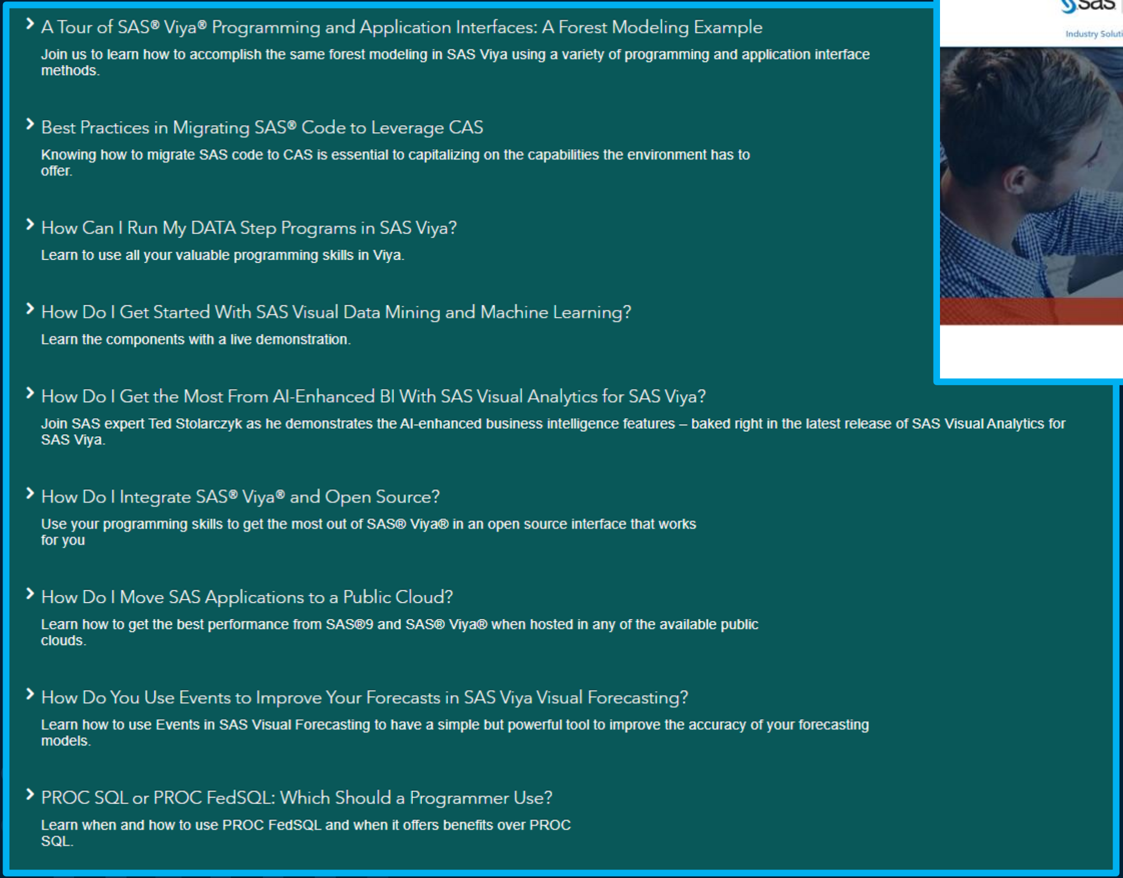
1-13 of 13

Sort By: Most Viewed

- 4:07 SAS® Viya™ Programming for SAS® 9 Programmers: Overview
- 6:43 Understanding Caslibs and Loading Data in SAS® Viya™
- 5:36 Using the DATA Step in SAS® Viya™

[Viya Programming for SAS 9 Programmers]

Ask The Expert Webinars



› A Tour of SAS® Viya® Programming and Application Interfaces: A Forest Modeling Example

Join us to learn how to accomplish the same forest modeling in SAS Viya using a variety of programming and application interface methods.

› Best Practices in Migrating SAS® Code to Leverage CAS

Knowing how to migrate SAS code to CAS is essential to capitalizing on the capabilities the environment has to offer.

› How Can I Run My DATA Step Programs in SAS Viya?

Learn to use all your valuable programming skills in Viya.

› How Do I Get Started With SAS Visual Data Mining and Machine Learning?

Learn the components with a live demonstration.

› How Do I Get the Most From AI-Enhanced BI With SAS Visual Analytics for SAS Viya?

Join SAS expert Ted Stolarczyk as he demonstrates the AI-enhanced business intelligence features – baked right in the latest release of SAS Visual Analytics for SAS Viya.

› How Do I Integrate SAS® Viya® and Open Source?

Use your programming skills to get the most out of SAS® Viya® in an open source interface that works for you

› How Do I Move SAS Applications to a Public Cloud?

Learn how to get the best performance from SAS®9 and SAS® Viya® when hosted in any of the available public clouds.

› How Do You Use Events to Improve Your Forecasts in SAS Viya Visual Forecasting?

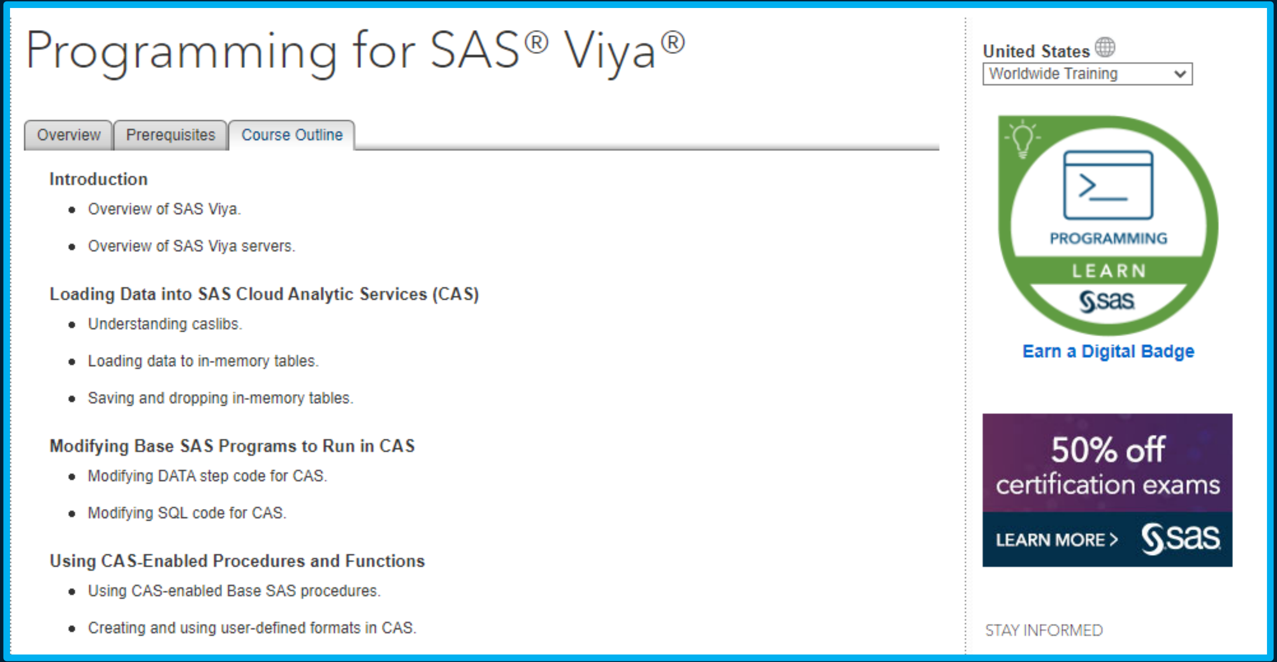
Learn how to use Events in SAS Visual Forecasting to have a simple but powerful tool to improve the accuracy of your forecasting models.

› PROC SQL or PROC FedSQL: Which Should a Programmer Use?

Learn when and how to use PROC FedSQL and when it offers benefits over PROC SQL.

Ask The Expert Webinars

Training



Programming for SAS® Viya®

Introduction

- Overview of SAS Viya.
- Overview of SAS Viya servers.

Loading Data into SAS Cloud Analytic Services (CAS)

- Understanding caslibs.
- Loading data to in-memory tables.
- Saving and dropping in-memory tables.

Modifying Base SAS Programs to Run in CAS

- Modifying DATA step code for CAS.
- Modifying SQL code for CAS.

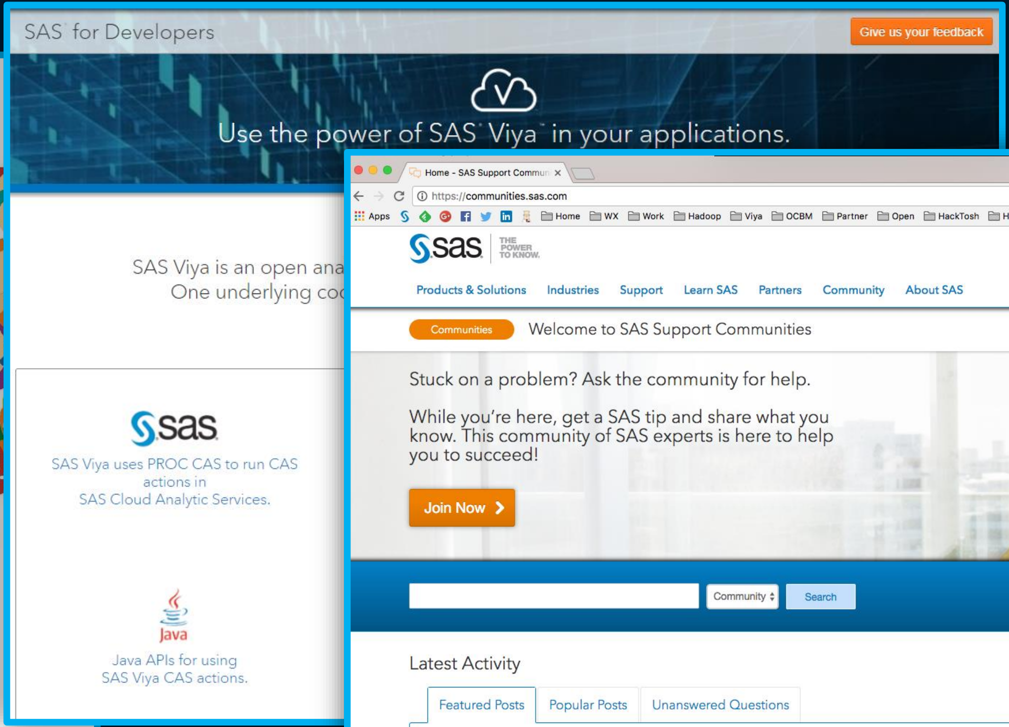
Using CAS-Enabled Procedures and Functions

- Using CAS-enabled Base SAS procedures.
- Creating and using user-defined formats in CAS.

[Programming for SAS Viya]

Useful Websites

Developer.sas.com, Communities.sas.com



SAS for Developers

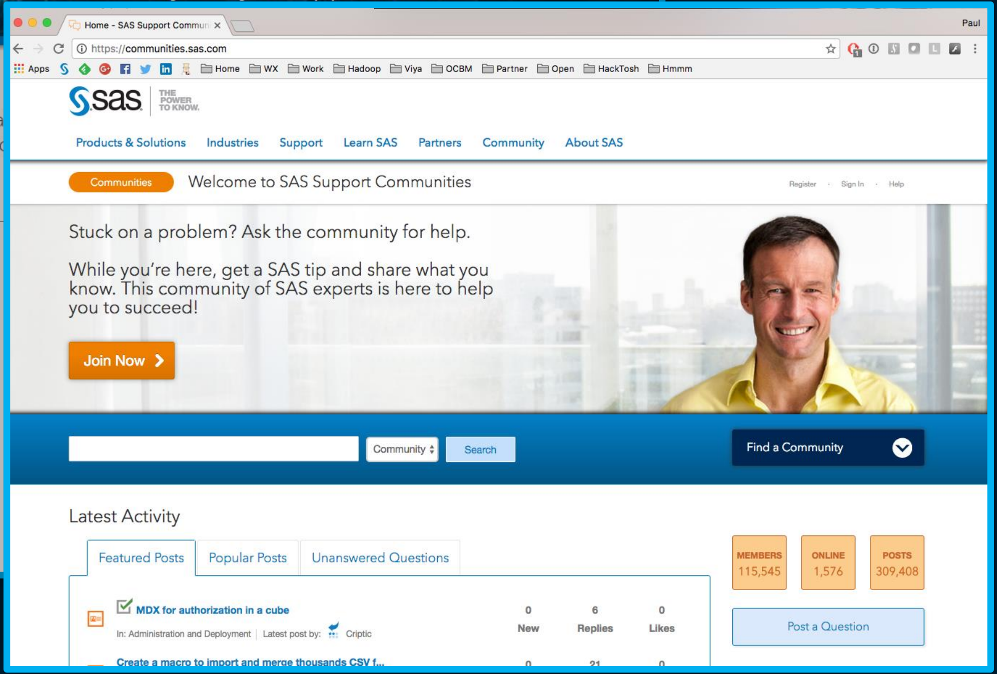
Give us your feedback

Use the power of SAS Viya in your applications.

SAS Viya is an open ana
One underlying coc

SAS Viya uses PROC CAS to run CAS actions in SAS Cloud Analytic Services.

Java APIs for using SAS Viya CAS actions.



Home - SAS Support Commun

https://communities.sas.com

Apps · Home · WX · Work · Hadoop · Viya · OCBM · Partner · Open · HackTosh · Hmmm

Paul

THE POWER TO KNOW.

Products & Solutions · Industries · Support · Learn SAS · Partners · Community · About SAS

Communities · Welcome to SAS Support Communities · Register · Sign In · Help

Stuck on a problem? Ask the community for help.

While you're here, get a SAS tip and share what you know. This community of SAS experts is here to help you to succeed!

Join Now

Community · Search · Find a Community

Latest Activity

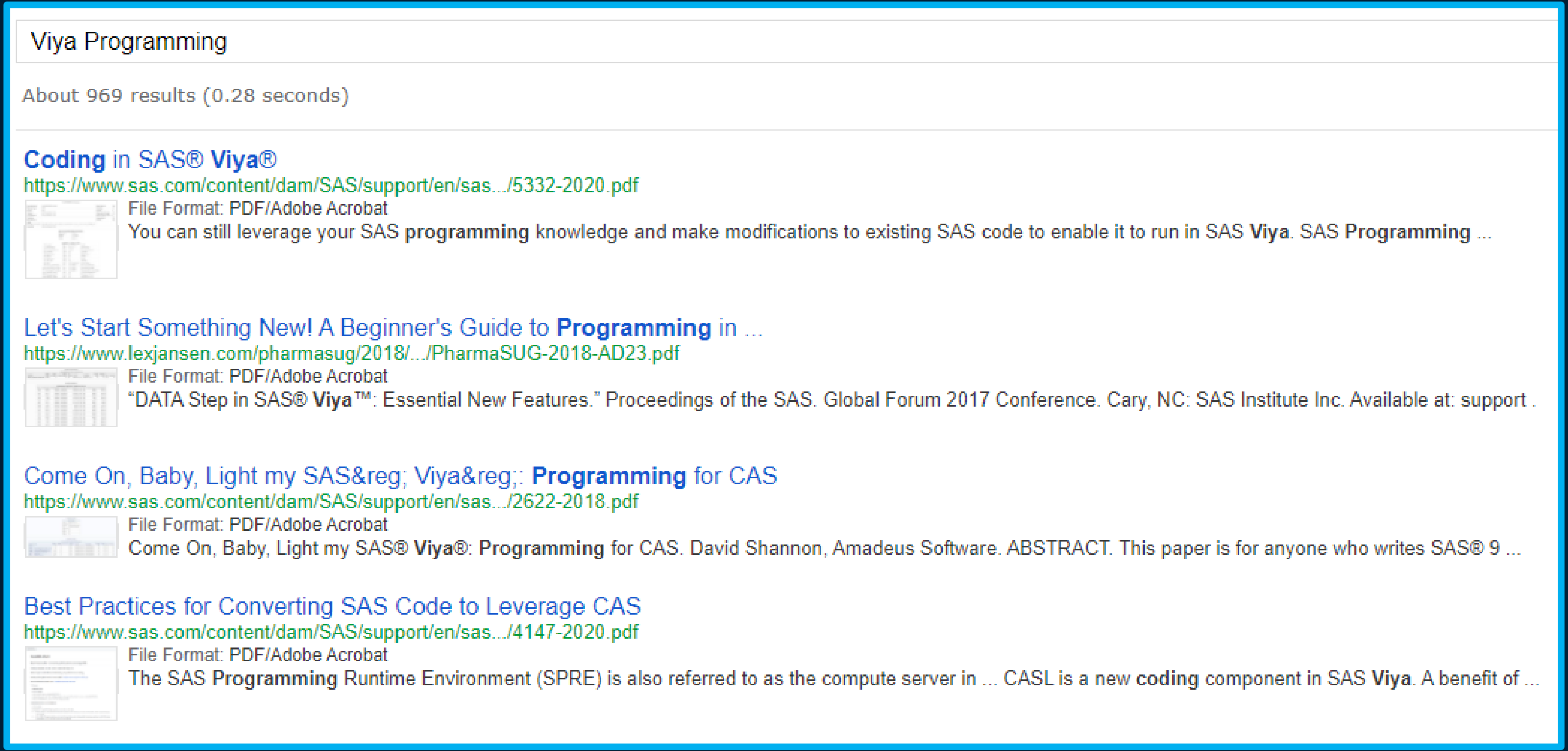
Featured Posts · Popular Posts · Unanswered Questions

	New	Replies	Likes
MDX for authorization in a cube In: Administration and Deployment | Latest post by: Criptic	0	6	0
Create a macro to import and merge thousands CSV f...	0	21	0

MEMBERS	ONLINE	POSTS
115,545	1,576	309,408

Post a Question

Papers



Viya Programming

About 969 results (0.28 seconds)

Coding in SAS® **Viya**®
https://www.sas.com/content/dam/SAS/support/en/sas.../5332-2020.pdf
File Format: PDF/Adobe Acrobat
You can still leverage your SAS **programming** knowledge and make modifications to existing SAS code to enable **it** to run in SAS **Viya**. SAS **Programming** ...

Let's Start Something New! A Beginner's Guide to **Programming** in ...
https://www.lexjansen.com/pharmasug/2018/.../PharmaSUG-2018-AD23.pdf
File Format: PDF/Adobe Acrobat
"DATA Step in SAS® **Viya**™: Essential New Features." Proceedings of the SAS. Global Forum 2017 Conference. Cary, NC: SAS Institute Inc. Available at: support .

Come On, Baby, Light my SAS&reg; Viya&reg;: **Programming** for CAS
https://www.sas.com/content/dam/SAS/support/en/sas.../2622-2018.pdf
File Format: PDF/Adobe Acrobat
Come On, Baby, Light my SAS® **Viya**®: **Programming** for CAS. David Shannon, Amadeus Software. ABSTRACT. This paper is for anyone who writes SAS® 9 ...

Best Practices for Converting SAS Code to Leverage CAS
https://www.sas.com/content/dam/SAS/support/en/sas.../4147-2020.pdf
File Format: PDF/Adobe Acrobat
The SAS **Programming** Runtime Environment (SPRE) is also referred to as the compute server in ... CASL is a new **coding** component in SAS **Viya**. A benefit of ...

Lex Jansen Search

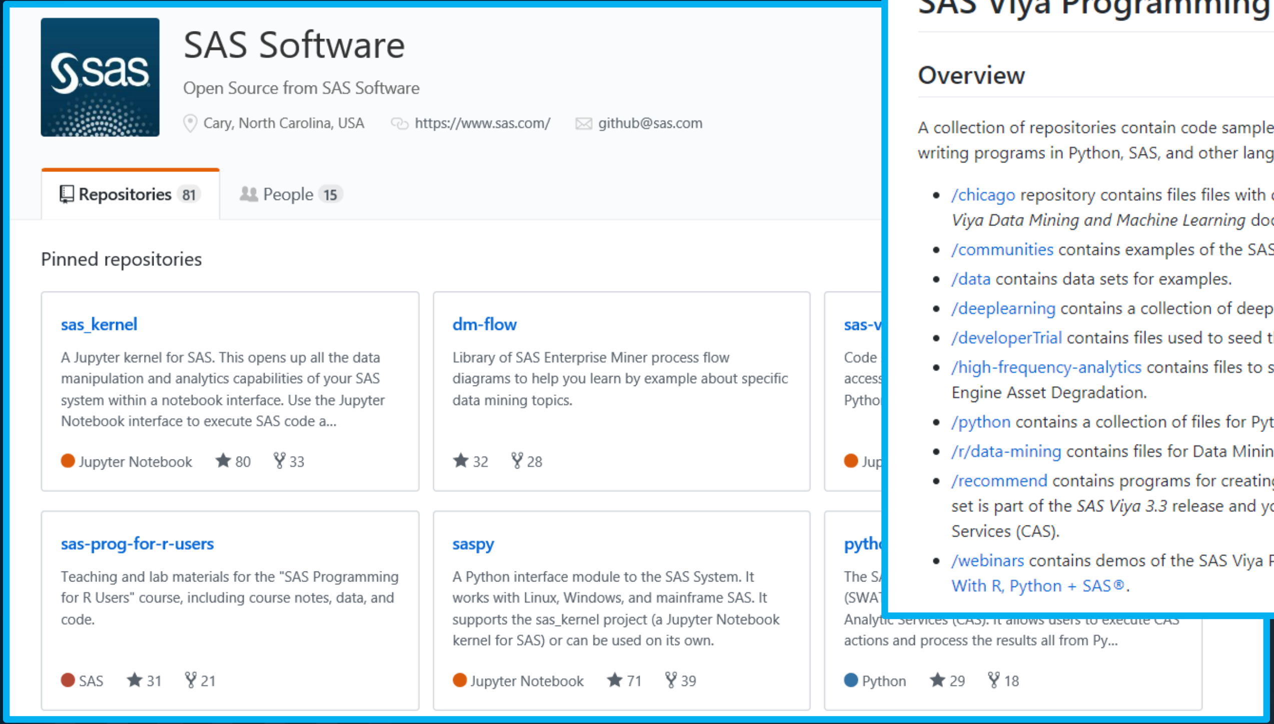
Best Practices – Paper, Super Demo, Repository

Paper: SAS4147-2020 Best Practices for Converting SAS® Code to Leverage SAS® Cloud Analytic Services Steven Sober, Brian Kinnebrew, SAS Institute Inc.

Super Demo: SAS4147 Best Practices for Converting SAS® Code to Leverage SAS® Cloud Analytic Services

GitHub Repository for SAS4147. https://github.com/sascommunities/sas-global-forum-2020/tree/master/papers/4147-2020-Sober

https://github.com/sassoftware



SAS Software

Open Source from SAS Software

Cary, North Carolina, USA   https://www.sas.com/   github@sas.com

Repositories 81   People 15

Pinned repositories

sas_kernel
A Jupyter kernel for SAS. This opens up all the data manipulation and analytics capabilities of your SAS system within a notebook interface. Use the Jupyter Notebook interface to execute SAS code a...
● Jupyter Notebook ★ 80 33

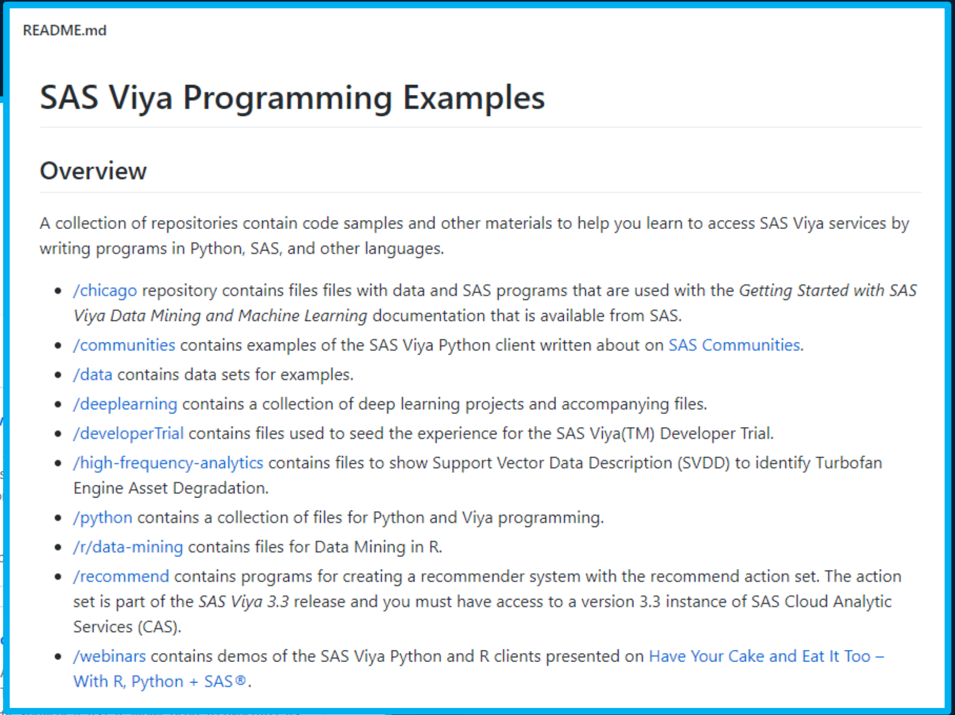
dm-flow
Library of SAS Enterprise Miner process flow diagrams to help you learn by example about specific data mining topics.
★ 32 28

sas-v
Code
access
Pytho
● Jup

sas-prog-for-r-users
Teaching and lab materials for the "SAS Programming for R Users" course, including course notes, data, and code.
● SAS ★ 31 21

saspy
A Python interface module to the SAS System. It works with Linux, Windows, and mainframe SAS. It supports the sas_kernel project (a Jupyter Notebook kernel for SAS) or can be used on its own.
● Jupyter Notebook ★ 71 39

pyth
The S
(SWAT
Analytic Services (CAS). It allows users to execute CAS actions and process the results all from Py...
● Python ★ 29 18



README.md

SAS Viya Programming Examples

Overview

A collection of repositories contain code samples and other materials to help you learn to access SAS Viya services by writing programs in Python, SAS, and other languages.

- /chicago repository contains files files with data and SAS programs that are used with the *Getting Started with SAS Viya Data Mining and Machine Learning* documentation that is available from SAS.
- /communities contains examples of the SAS Viya Python client written about on SAS Communities.
- /data contains data sets for examples.
- /deeplearning contains a collection of deep learning projects and accompanying files.
- /developerTrial contains files used to seed the experience for the SAS Viya(TM) Developer Trial.
- /high-frequency-analytics contains files to show Support Vector Data Description (SVDD) to identify Turbofan Engine Asset Degradation.
- /python contains a collection of files for Python and Viya programming.
- /r/data-mining contains files for Data Mining in R.
- /recommend contains programs for creating a recommender system with the recommend action set. The action set is part of the *SAS Viya 3.3* release and you must have access to a version 3.3 instance of SAS Cloud Analytic Services (CAS).
- /webinars contains demos of the SAS Viya Python and R clients presented on Have Your Cake and Eat It Too – With R, Python + SAS®.

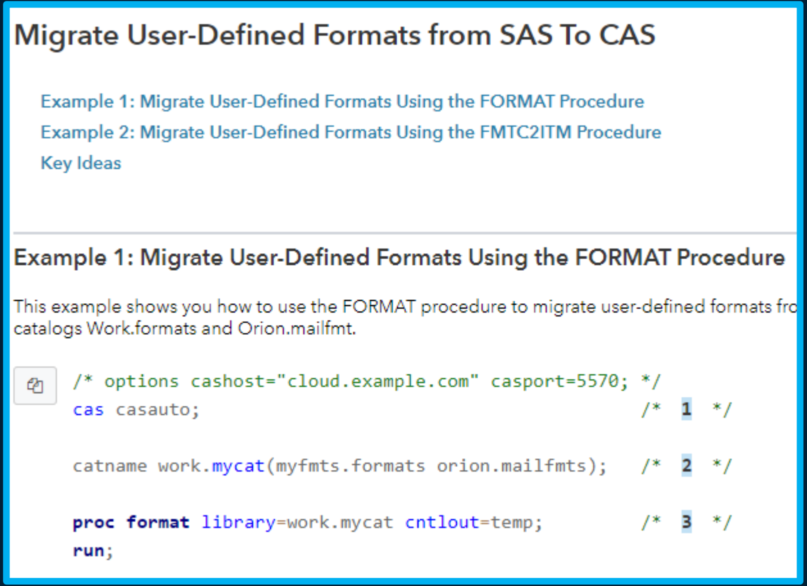
SAS Viya Data Processing

Formats

[CAS enabled formats]

[Manage your user-defined formats in CAS]

[Migrate your user-defined formats from SAS to CAS]



Migrate User-Defined Formats from SAS To CAS

Example 1: Migrate User-Defined Formats Using the FORMAT Procedure
Example 2: Migrate User-Defined Formats Using the FMTC2ITM Procedure
Key Ideas

Example 1: Migrate User-Defined Formats Using the FORMAT Procedure

This example shows you how to use the FORMAT procedure to migrate user-defined formats fro
catalogs Work.formats and Orion.mailfmt.

```
/* options cashost="cloud.example.com" casport=5570; */
cas casauto;                                          /* 1  */

catname work.mycat(myfmts.formats orion.mailfmts);    /* 2  */

proc format library=work.mycat cntlout=temp;          /* 3  */
run;
```

Viya Programming HOW

Hands On Workshop – End of Section #1