



**VINCI**  
**SAS/GRID**

# Tricks of the Trade and Open Source Processing

**Mark Ezzo**

**VINCI SAS Administrator**



**July 12, 2018**





# Poll Question

- **What is your primary function:**
- **Analytics**
- **Data Manipulation/Preparation**
- **Both**
- **Other**





# Space Considerations

- **SAS Clients have limitations of 100 GB of Project Space for Consumption in the Windows Environment.**
- **SAS Grid has ~73TBs with an expandable device.**
- **SAS Work Space per node is 2.5TB currently**





# Grid Advantages

- **Multiple Node Environment**
- **Fail-Over capability**
- **Centralized Administration (SMC)**
- **Vast storage capabilities**
- **Parallel processing for faster results**
- **Leading infrastructure for Research and Corporate Technology**
- **Base/DM SAS will not be further enhanced by SAS Institute; please begin to use the Grid for all SAS Programming.**



# SAS EG 7.1

- **Standard for coding and Grid Access**
- **Enhanced and automatic features**
- **Easy to Configure and Use**
- **Wizards for many tasks**
- **Able to develop SQL Pass-Through Queries (recommended for the DB)**
- **Automatically adapt for parallel processing**
- **Live Demonstration!**





# Parallel Process

**In EG, via the Analyze Program feature, you have the ability to adapt a program for Parallel Processing. This allows much of the program to run simultaneously in multiple jobs slots. This is much faster and more efficient than a Linear Program utilizing one job slot.**

**Can be used for ETL or Analysis with Open Source Code!**





# Parallel Process

## Considerations:

- There are costs to parallel processing so the more complex the program/data the greater the benefit.
- Small programs do not have the need
- You may have to do a few modifications, depending upon the original program (libnames, etc.)
- Can greatly reduce processing time!
- Runs in both EG and GSUB (batch)
- Example



# SAS Gsub Example

- Please use GSUB for long processes and especially overnights.
- Produces a log and lst file and is independent of the Windows Network.
- Here is the command we will be using for batch processing on the Grid:
- `sasgsub -GRIDSUBMITPGM`  
`/data/prod/admin/VHASLCEZZOM/Programs/Program.sas`
- It breaks down as follows:
  - 
  - `sasgsub` is the command
  - `-GRIDSUBMITPGM` is the option to submit a program
  - `/data/prod/admin/VHASLCEZZOM/Programs/` is the directory where the program is located
  - `Program.sas` is the program you wish to run
  - Live Demo!



# SAS Enterprise Miner

- **SAS/EM is the state of the art Data Mining and Modeling Software.**
- **Friendly User interfaces allows less programming and more analysis.**
- **Access via IE through this link:**
- **<http://vhacdwdwhsasmt1.vha.med.va.gov:7980/SASEnterpriseMinerJWS/Status> for Operations or**
- **<http://vhacdwdwhsasmt2.vha.med.va.gov:7980/SASEnterpriseMinerJWS/Status> for Research\**
- **Quick Demo**



# Open Source with SAS

## R Integration with SAS/IML:

- Many R commands give little thought to memory management, and so R can very quickly consume all available memory. This can be a restriction when doing data mining. There are various solutions, including using 64 bit operating systems that can access much more memory than 32 bit systems.
- R is also very challenged with large amounts of data
- Not an enterprise product, though it does suffice for statistical analysis
- More powerful within the SAS Grid





# Proc IML

- With Proc IML server-side integration is provided
  - Requires security considerations
  - Must set the RLANG option
  - Constrained to a single version of R and the set of R packages
  - In PROC IML, R must be installed on the computer that runs the SAS server. Everyone that connects to that server uses the same version of R and the same set of installed packages.
  - We can download any package desired
  - Can use R 3.3, we are upgrading to R 3.5



# SAS and R comparisons

	SAS SOURCE	R TARGET
<ul style="list-style-type: none"><li>METHOD or</li><li>MODULE</li></ul>		
<ul style="list-style-type: none"><li>ExportDataSetToR</li></ul>	SASdata set	R data frame
<ul style="list-style-type: none"><li>ExportMatrixToR</li></ul>	SAS/IMLMatrix	R matrix
<ul style="list-style-type: none"><li>DataObject.</li><li>ExportToR</li></ul>	DataObject	R data
<ul style="list-style-type: none"><li>DataObject.</li><li>AddVarFromR</li></ul>	R expression	DataObjectvariable
<ul style="list-style-type: none"><li>DataObject.</li><li>CreateFromR</li></ul>	R expression	DataObject
<ul style="list-style-type: none"><li>ImportDataSetFromR</li><li>ImportMatrixFromR</li></ul>	Rexpression R expression	SAS data set SAS/IML matrix

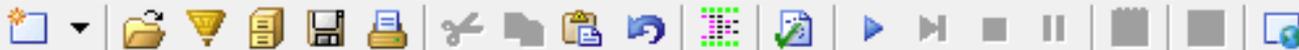




# Data Examples

SAS/IML Studio - Output1

File Edit View Program Graph Analysis Tools Window Help



SAS\_IML\_and\_R\_Integration\_Example.sx

```
/* Export data */  
run ExportDataSetToR("Sashelp.Class", "df" );  
submit / R;  
  names (df) ;  
endsubmit;
```

Output1

```
[1] "Name" "Sex" "Age" "Height" "Weight"
```





# SUBMIT R CODE from IML STUDIO

SAS\_IML\_and\_R\_Integration\_Example.sx

```
print "----- R Results -----";
submit / R;
  rx <- matrix( 1:3, nrow=1)           # vector of sequence 1,2,3
  rm <- matrix( 1:9, nrow=3, byrow=TRUE) # 3x3 matrix
  rq <- rm %*% t(rx)                  # matrix multiplication
  print(rq)
endsubmit;
```

Output2

```
----- R Results -----
      [,1]
[1,]    14
[2,]    32
[3,]    50
```



# Live Demo!





# Python and SAS

Python is rightfully showing enormous popularity in the market. It is a very powerful open-source program that appears to be supplanting R. It does have the same open-source issues such as memory, I/O, security, etc. SAS has embraced all Open-Source processing with SAS Viya. It is being installed in many companies in Private Industry and in the Public Sector and is becoming a standard. Unfortunately, we will not be getting it in the near future. Therefore, we will be exploring SASPY. You can also submit Python commands from SAS utilizing the 'X' command.



# SASPy

**This module provides Python APIs to the SAS system. You can start a SAS session and run analytics from Python through a combination of object-oriented methods and Python magics.**

**The APIs provide interfaces for the following:**

**Start a SAS session on the same host as Python or a remote host.**

**Exchange data between SAS data sets and Pandas data frames.**

**Use familiar methods such as `describe()` and `head()` to work with data.**

**Additional functionality such as machine learning, econometrics, and quality control are organized in Python classes.**



# Dependencies

- **Python3.X or higher.**
- **SAS 9.4 or higher. SAS Viya 3.1 or higher is also supported.**
- **An integrated object method (IOM) connection (one of four connection methods) requires Java on the client and four JAR files from your SAS installation.**
- **You can connect to SAS on any platform that is supported for the specified SAS releases.**
  
- **Let's view a few examples**





# Jupyter magics

- Typically, programming with this module is performed with the Python functions that are provided by the package. For example, to view the first few rows of a data set, you can use the `head()` method. However, if you are an experienced SAS programmer, there might be occasions in which you prefer to run SAS statements—such as running `PROC PRINT` to view the first few rows of a data set.
- The magics that are available with the package enable you to bypass Python and submit programming statements to your SAS session.
- The `%%SAS` magic enables you to submit the contents of a cell to your SAS session. The cell magic executes the contents of the cell and returns any results.

```
%%SAS proc print data=sashelp.class; run;  
data work.a; set sashelp.cars; run;
```



# More

If you are also invoking `saspy` methods directly in other cells within the same notebook, you may indicate that a `%%SAS` cell should share the same SAS session by passing your existing session as a parameter.

In a prior cell in the notebook:

- `import saspy my_session = saspy.SASsession() # sending a dataset to SAS as 'mydata' mydata = sas.df2sd(some_pandas_dataframe, 'mydata')`
- Then later, invoke a SAS cell and pass your existing session so that the remote data set is accessible:
- `%%SAS my_session proc print data=work.mydata; run;`





# IML

The %%IML magic enables you to submit the contents of a cell to your SAS session for processing with PROC IML. The cell magic executes the contents of the cell and returns any results. The PROC IML statement and the trailing QUIT; statement are submitted automatically.

- %%IML a = I(6); \* 6x6 identity matrix;
- b = j(5,5,0); \*5x5 matrix of 0's;
- c = j(6,1); \*6x1 column vector of 1's;
- d=diag({1 2 4}); e=diag({1 2, 3 4});



# More to Come

**We have only demonstrated the surface of what we can do with Python and SAS w/o SAS Viya. Make no mistake, SAS Viya is the far preferred method, but we can utilize this until budgets allow us to implement it.**

**This is our analytical SAS/Open Source Roadmap:**

- **Upgrade R to 3.5 on all Grid nodes**
- **Socialize the usage of R within SAS (this is occurring as we speak very successfully)**
- **Implement Python and SASPY in a test location**
- **Validate usage and best practices**
- **Pilot Project (Volunteers Please)**
- **Place it into the Production Grids**



# Question or Comments?

- **Please ask any questions at this time or you can send them to us later.**
- **We will keep you informed and involved.**





# Good Gridding!

**Thank you for attending.**

**Please contact VINCI SAS Administrators:**

**VINCI SAS Admins**

**[VINCISASAdmins@va.gov](mailto:VINCISASAdmins@va.gov)**

**with any questions or comments.**