Cyberseminar Transcript

Date: July 12, 2018

Series: VA Informatics and Computing Infrastructure

Session: VINCI SAS/GRID – Tricks of the Trade and Open-Source Inclusion

Presenter: Mark Ezzo, BS

*This is an unedited transcript of this session. As such, it may contain omissions or errors due to sound quality or misinterpretation. For clarification or verification of any points in the transcript, please refer to the audio version posted at* [http://www.hsrd.research.va.gov/cyberseminars/catalog-archive.cfm](file:///C:\Users\VHASLCMyersK\AppData\Local\Microsoft\Windows\Temporary%20Internet%20Files\Content.IE5\73RXHSNF\l)

Rob: And as we’re just about at the top of the hour, I'd like to introduce our speaker today. Mark Ezzo is a VINCI SAS administrator working for first U at Salt Lake City, Utah, for the VA OI&T Servicing and Engineering Service. Mark, can I turn things over to you?

Mark Ezzo: Yes sir. Show my screen, and please let me know if you can see the presentation.

Rob: I can.

Mark Ezzo: Great. So folks, this is me again, once again. As always, it’s an honor and a pleasure to present to you in our continuing presentations about analytics in the VINCI SAS Grid and open source. What we’re presenting today is the SAS Grid, tricks of the trade, how best to utilize it in preparing your data and your studies, and open-source processing inclusion, meaning that you’re not just tied to SAS here. You could use SAS and you could use R, and we are setting up a facility for Python going forward. So therefore, what we are trying to do in BI and analytics is include as much of the toolbox as we possibly can. I was never a believer in one size fits all. None of these companies pay my salary; therefore, we’re including everyone. And there’s a lot of popularity with R and Python at this time, and we have found that the inclusion of open source with the Grid is a very, very powerful tool. And I've been spending quite a bit of time working with people to that regard.

Here’s our first poll question. What is your primary function? Are you in analytics, do you do data manipulation/preparation for modeling or studies, do you do both, or is it other? I’ll give you a few minutes to answer.

Rob: Mark, as you were speaking, I brought the poll up, so audience members, the poll is up and available for you to go ahead and click right on your screen. Choose any of those options. And if other is the option you choose, feel free to use the questions pane to enter what other means, and I'll tell Mark what that is. We have about 75% of our audience members voted. It usually levels off right around 80, so I'm going to give people a few more moments to make their choice.

Mark Ezzo: And he used advanced analytics to come up with that 80 percentile.

Moderator: Right. And yes, it has leveled off right around 80%, so I'm going to close the poll and share the results out. And Mark, 7% chose analytics, 13% data management/preparation, 65% chose both, and 15% chose other, and I didn't get any answers as to what other means just yet. So back to your slides.

Mark Ezzo: Oh, this is surprising. I figured both would win. So very good, as anticipated. Okay, let’s proceed. Now let’s talk about the Grid and why the Grid is a preferred platform as opposed to SAS desktop. Space considerations is one of the primary functions, and SAS clients have limitations of 100 gigabytes in a project space for consumption. Many of you have seen this before. We have 73 terabytes out on the Grid, and that’s an expandable device. We also have workspace per node. If you’re in the, for example, there’s four nodes per VINCI research Grid, four nodes per operations Grid, so we have 10 terabytes of workspace for each grid as opposed to the small area you’ll get on a Windows client, okay?

And other Grid advantages, it’s a multi-node environment, meaning unlike a regular single-standing server, if we lose a compute node, the Grid marches on. We have fail-over capability. We have centralized administration where we can handle, we being the SAS Grid administrators, security, group inclusion, you name it. It’s a very powerful tool. Again, we have vast storage capabilities. We have parallel processing for faster results. I, myself, have helped many groups take processes that are taking hours or even days and knocked them down considerably, sometimes as much as 5% of previous run time, and so are leading infrastructure for research and corporate technology. And as far as SAS Grids throughout the market, they really dominate quite a bit. I would say in probably the first five years of it, they went up to, I think the number now is about 1,800 to 2,000 SAS Grids out there, and they’re increasing every day.

One thing here, which I think is very important, what many of you use, base desktop manager SAS, will not be further enhanced by SAS Institute. In fact, they stopped doing that with the release of version 9. What was that, about six years ago? So we urge you all to please begin to use the Grid for all SAS programming. It’s only to your advantage. And we offer training in the SAS Grid EG world. Tony [unintelligible 05:32] will train, Kevin Martin will train, I will train, and we have converted hundreds of users, and none in my mind have gone back to using SAS desktop. So we urge it most vehemently.

Okay, now what do we use for the SAS Grid? Now I again apologize to anyone who has seen this before, but it bears repeating, and we’re going to look at a few things also. It is the standard for coding and Grid access in SAS. There’s also SAS Studio, which we may or may not use in the future, but we can also include that option. It has enhanced and automatic features, many, many wizards. It’s very easy to configure and use. Essentially you just make a very simple profile. As I mentioned a moment ago, there are wizards for many of the tasks. We are able to develop SQL pass-through queries, recommended for the DB, and that’s another vital part of optimization. Whenever you’re pulling off of our very, very large Corporate Data Warehouse or your own research database, always, always use pass-through queries in the SQL server. What a pass-through query does, it passes all of the code, the SQL code, up to the server, allows it to run up there, you’re not at the behest of the network, and just returns the results back. If you do not do that, then it forces SAS to bring down any table, and if you’re doing several joins, that can be monstrous to workspace before it does any joining or any culling.

And one extremely useful one, which we spoke about earlier, we can automatically adapt code for parallel processing. Now what is parallel processing, or you may call it concurrent processing. Essentially it has a feature that will look at your code and break it up into designated pieces. For example, if you had one large dataset and you were running five miles or procedures against it, instead of running it in a linear fashion, and I've used this example before, and if each procedure took an hour, you have a five-hour program. This way, it takes only as long as the longest one. So you could run all five procedures in one hour.

So let’s just look at some things with Enterprise Guide very quickly, and we’ll be coming back to this for R, but I just wish to show you a few things. As far as the wizards are concerned, well, let’s look at setup for a moment. This is, besides what we do in a management consult to enable you, this is all you need to do for setup. You essentially just go in, you can create a very simple profile where we put everything in. We’re IWA right now, so all you do is click this. IWA is your PIV card. Reference the metadata module, and you’re good to go. No issues there. That’s as simple as it is to configure. You have many ways that you can look at how you want it to appear. Project views. You can have a look and choose anything you like here. It can tell you where your project recovery is. I send mine to a different area. You can control your results. Viewer can be Windows default, preferred view, Enterprise Guide, whatever. You can manage what your output looks like. You can manage your HTML, et cetera, et cetera, et cetera. You also have stored processes here. If you can create those, and we’ll talk about that in a little bit, you can tell what alpha you want there.

Now, in regards to data general, you can control Howell, and that’s not an old computer company, how you want your data to come out once you run it. For example, I like to open the data myself, automatically import data when added to a project, always prompt when closing project with temporary data references, so I do not lose anything I like. I can see what it looks like when I bring in data to the Grid, how I want it to look, and we can look at performance. For example, you can set as a 255 or the longest value within the data. There’s pros and cons of both. I limit the number of rows when importing data to look at to 50, and when I'm looking at SAS access data, which would be SQL server data, I limit that to 1,000 rather than download the whole thing. And I'll close data grids if I'm inactive for 30 seconds.

Now, in regards to other things, I don’t want to take too much time for this, have a look at this. You can use PROC SQL instead of PROC SORT in tasks, and I like to do anyway, so let’s do that. And we’ll go back to our options. You can put in custom code before query or after, SAS or query code. This is the normal output libraries that we like to do. And we have other functions, which you are more than encouraged to peruse. But the point of the matter is, that’s how we set it up. That’s how we can control it.

Now for you folks who are new to SAS or are not what we call intermediate or expert, you can create any code you like right here. This essentially is parameter driven. Just pick a task, and what it does, it creates the code for you so that you can use that later on in a SAS macro or just as a learning technique. You can import Stata or SPSS files. You can sort. You can split columns. You can random sample, rank, and that’s just data. You can look at how to describe the data, summary statistics, et cetera, et cetera. It’s very useful. When you want to do a little data visualization, you can create graphs from your data and include that in your projects. Many, many times when I'm writing the [unintelligible 11:43] code from SAS, which I'll admit can be quite clumsy, but this is much better. This will create it and you can modify it. You can do a lot of statistical work here. As we can see, linear models, mixed models, regression, and that’s high-performance linear regression, multivariate, survival analysis, and the important part is this will create the code for you. So you can take a sample, you can run this code, you keep the data, you can include it. It’s a wonderful learning facility. You can kind of look at, again, more data visualization, histograms, control charts, Pareto charts, and for people like myself who are economists, Pareto optimality, time series. This is excellent.

Times series, as you know, needs about three years of data to be relevant, and this will actually guide you in creating time series and making sure that it’s relevant and will even grade you as far as the relevance of what you’re doing. We have data mining, model scoring, rapid predictive modeler, and that is like Enterprise Miner light. And you can also look at how the data looks as far as recency, frequency, monetary analysis, et cetera. Okay? And we can also put in template.

So I encourage you folks who are new to SAS or are going to migrate, hopefully, from desktop SAS and come to EG, and you can get any code that you like here. You can manage your EG sessions up here also, import data, project properties, et cetera. The usual editing, how you want to view things, and help is very good too. Help is excellent. And of course you can always call us.

There are tools to get started. So as I said before, we are willing to train whomever, and you can also, once you get started in Enterprise Guide, I encourage you to try some tests, do some sampling, and keep the code and feel free to ask us any questions, okay? So I know many of you are using it, and I don’t want to harp on that too much, but it’s a very important thing to use. Now, we’ll come back to that later when we’re doing our open-source inclusion because we do that through Enterprise Guide in the Grid.

Now, parallel processing. We spoke about this before. In EG, via the Analyze Program feature, you can have the ability to adapt a program for parallel processing. And I will show you an example of that with the open source, but I would like to bring this up now for you. This allows a program to run simultaneously or concurrently, however you want to say it, in multiple job slots. We usually allow five, but we can increase that. This is much faster and more efficient than just piling PROC upon PROC or model upon model in one job slot. And you can use that for ETL or analysis, and you can use it with open-source code, which we will demonstrate later. And then that way, the power behind that is as follows: If you have, say, several R or, going forward when we get everything set up, Python and you want to run them concurrently, these are not stuck running one at a time. You can run them in several job slots and that way get your analysis done that much more quickly. Therefore, we’re getting a lot of ROI, return on our investment. You get a good bang for your buck from the Grid just from parallel processing alone, one of the most powerful features of any grid, whether it’s SAS or whatever grid you happen to be in.

But there are considerations. There are costs to parallel processing, so the more complex the program data, the greater the benefit. If you’re having several little things that are, like I'm going to run five or six PROC PRINT for 100 observations, I would not put that in a parallel process. If I am going to run a mixed model, a regression, an NLIN, whatever, and it’s going to go against large amounts of data, I'm definitely running that in the parallel mode. You may have to do a few modifications. I always take and put my libnames and my options within each submit group. And for people who have used SAS/CONNECT in the past, that’s all they are, R submits, remote submits to the Grid notes. And again, I cannot stress what a powerful option this is and how greatly it can reduce your processing time. And it runs in both EG and GSUB, so therefore if I am testing on EG and I have a job that runs in parallel, I can save the code and run it in our GSUB batch environment. And I will show you that in the very near future.

And this is the very near future. We recommend very, very strongly that you use GSUB for long processes and especially for anything overnight. Now why is that? Number one, you do not have to leave your computer up. You do not have to worry about network interruptions. This runs within the Grid network itself. So for example, if your computer goes down, if you lose power, if you have a thunderstorm like I may have in a moment, it’s running on the Grid. You can always come back to it, and you’ll have logs and lists, and as we show here, produce a log and LST file that's independent of the Windows network. It’s very simple to do. That’s really about all the commands you need. SAS GSUB minus grid submit program, the location of it, and again, you’re in a Linux world, so it’s case sensitive. If you say I know it’s there, why isn’t it coming up, it is 99.9 repeating, your case is off. And as I said, this is exactly how that works, and we can do a very quick live demo on that too.

Let’s go over to another node. SAS Enterprise Miner, we’ll be coming back to that. This essentially is putty. And this, as you folks all know, is, let me close this guy out. Here is what you’d be accessing, putty-CAC. So essentially I just click on that. I created that to our GSUB environment, and this is in VINCI GS2. So then I merely open it, and you would just put in your user name, hit enter, and I'm done. Essentially this is driven again by IWA, so therefore no need to put my password in. And that’s all configured for you, and we can assist in the configuration if you have issues.

Now, as we had shown before, this is the command we have, SASGSUB, GRIDSUBMITPGM, and I am just doing a very, very simple RPC, and it says this is where everything is located. So all I would do, it's going to be, is I would copy that, and we use our other tool, WinSCP, and we bring that up, and I want to go into the same node, GS2. So I say log on, and it’s IWA compatible, and there we are. That’s my program. You can bookmark just by hitting add, and here’s where we go look at our data, which is in GridShare, which we clean up periodically because it can get quite extensive. And what time is it? It is 1:20. So that is what I had run before, and it shows me a log, which I can save and I do not lose. I can keep that forever. I can export it, import it. These errors are expected. In fact, this is one of the tests you’ll see later. And it can show me a list, and this is co-generated with SAS and R, which we will demonstrate later.

But then once all that is done, as we see in our demonstration, it shows you the node it was on. It shows you the actual program. Like this is PROC IML, and this is R within it. We can run that batch. We can run that interactively. So if I have very, very large jobs, I cannot stress strongly enough that you really do want to run them in GSUB. Please do not run them in EG if they’re going to go overnight. I cannot tell you how many phone calls the SAS admins will get saying I was running something last night, now my session is not up. I don’t know what happened. I think I lost all my work, yada, yada, yada. This will prevent that. Unless all the nodes die, and that has never happened and never will, your work will be saved. You’ll have log and LST files, and you’ll have code, and you can share this and you can drag it back over to the Windows side, which is what we have here. Left is Windows, right is Linux. Okay? So do we have any questions about what we’ve seen with EG and GSUB at this point? I will assume we do not.

Rob: We don’t have any questions here, Mark.

Mark Ezzo: That’s wonderful. Thanks. Okay, so let us go forward. Now let’s talk a little bit about Enterprise Miner. Again, what we’re showing you is all the tools in the Grid toolkit, and SAS EM has been really growing out in the market by leaps and bounds. Even I get on a couple R blogs, and I have people say, I'm really surprised, all the jump and all the SAS Enterprise Miner that’s coming out there. But there’s a reason for it. It’s very easy to use. It has a very friendly user interface and allows you to actually just fill in the blanks and model that way rather than sitting there and writing code over and over and over. And we access it through links, and the MT1 is operations, and MT2 is research.

So let’s do this. Let’s go again back to that same node, which is within VINCI. And I had brought this all up. This is just a very simple set. This is a sample that I actually send to everyone. This is a very, very useful little project that SAS hands out, and I also send out a manual with it that we have in Adobe. And it explains everything, how to set everything up. It’s quite good, and it shows many of the features of it, how you can do data partitioning, you can explore multipart replacement, how you can create, import data, et cetera, how you can transform the variables. And this runs all these concurrently so I'm not waiting on something over and over. Because what we do is we come up to our options, you state preference, we come down, and we say use say use grid when available. I recommend that. You can run it on a single server, and it will also run quite fast. But I always feel it’s better, especially if you’re dealing with large amounts of data, to spread that out amongst the grid. And this is automatically adapted for parallel processing, so there’s no worries there. All you really have to do is just set it up, and SAS Enterprise Miner will take care of that for you. And also what it does is you can bring all the results back to a control point, and you can run a model comparison. And if you do here, let’s say we want to say results, and you’ll see this in the example. I always thought this was one of the best portions of it. This will tell you if you’re running several models, and as far as what your parameter of success is, this will tell you what your best model is.

So in this case I ran some interactive trees, a neural net, a tree that I essentially pruned myself, a regression, and an [unintelligible 24:58] neural net. And then saying as far as what the parameter here was, greatest profitability, the second, the interactive tree was the best one. Therefore, that’s the model you should use going forward. And as far as modeling capabilities, it’s quite large. You can build neural nets here quite easily, and you can even do it in high performance. There’s many things up here to use, and it’s a very powerful tool.

There’s a group in VINCI and the VA that use it quite a bit, but I still believe it’s somewhat underutilized. And if you folks, what we stressed earlier about taking all your data and preparing it in EG and then you can use tools like Enterprise Miner to do your analysis, that’s really the way to go because what all you really need to do is once you have it prepared, you can click here, and you can put your libnames in and any code that you want to use. And you could run that, and now every time that you would run this project or any project, because that was for something else that I was running up here, just the status [unintelligible 26:10] on inpatient data, that SAS code will be available for me, and I can just add to those libraries, I'm sorry, SAS data, and I can just add other SAS data to those libraries, add in an item and can do analysis.

You can put several diagrams under one project, which I find to be quite useful. Instead of clogging everything in one very messy diagram, you, for example, could say I have one for gender, I have one for race, I have one for whatever category I choose. I've seen people with like a dozen diagrams in there or more, and I encourage that greatly.

This is a tool I think everyone who does any statistical modeling or data modeling should really be involved with. It will save you time, and the power of it will astound you to a great degree. So I don’t want to press too much on that. I will be very glad to give any groups or individuals classes on this and assist them, and also with EG, rather than take up our time now. But the more you’ll use this, the better you will enjoy it.

So back to our presentation here. So that’s Enterprise Miner. So what have we seen so far? We’ve seen Enterprise Guide, which is our interactive where you should do the majority of your testing and programming. We’ve seen GSUB, which is our batch environment, which is where you should run all your production or let’s just say long-running code and code that you can run on a scheduled basis. And we can schedule for you. We have something called SAS CRON, which takes advantage, which you can write, and it takes advantage of the CRON scheduling within Linux. And you can set that up to run at any time. It can run once a month. It can run every day at certain times. It can run more than once a day. And we have the documentation for it, and you can get any documentation in the link I have for my e-mail. The last thing I have is a link to all of our documentation. We’ll gladly send it to you, and I can walk you through it, but that’s what the majority of people are using for scheduling. Some will schedule in EG. I kind of wish they wouldn’t if it’s going to be something late at night when backups are occurring. You’re much better off doing it in the batch environment, and we will be glad to assist you with that.

Okay, now let’s talk about something that will be much newer to most of you folks. Open source with SAS. SAS has come to the realization quite a while ago that they are not the only game in town. And they understand that a lot of people, especially in the academic world, are R users and are also a lot more, and Python is coming along very strongly. So what SAS has done, I think very wisely, is they have taken their excellent infrastructure, the SAS platform, which is really still the standard out there. If you’re in private industry or in the public sector, you’re going to notice that if they say we’re looking for analytics, they always mention SAS. And I know that several clients who use open source within a SAS Grid or within the SAS environment, because they trust SAS to handle the memory and the I/O more than open source, but they like what a lot of open source can give them. So therefore we have combined them, or SAS has combined them.

Here, let’s talk about this a little bit. Many R commands give little thought to our memory management, and so R can very quickly consume all available memory. We have found this to be true in the Grid nodes. This can be a restriction when doing data mining. There are various solutions, which we have addressed. R is also very challenged with large amounts of data. I think a little bit less so than it used to be, but you still have to be very careful how you manipulate it. I find may people will enjoy doing data manipulation in preparation in SAS and then passing that to their open-source environment, even doing some, maybe running the same analysis in SAS and in R for comparative sake. What I have noticed, where I see a lot of R usage in SAS, is in graphing. I see a lot of people, and I understand perfectly why, prefer the R graphing environment to the SAS graphing environment. A lot of it because of writing the code, but if you use the wizard, I think that will mitigate some of it, but it’s also what you know. And instead of trying to force everybody into one round hole or one square peg or whatever you want to call it, I think it’s better to be more inclusive. The more tools we have in our toolbox, the more productive we are going to be. So we say R is not really an Enterprise product, though I think there could be an argument about that. But it’s far more powerful within the SAS Grid.

Now, how do we use R within the SAS Grid? And we will get to Python in a little bit also. We use something called PROC IML, and that essentially is interactive matrix language. There are security considerations there. You want to be a little, we scan every R product that comes down to make sure that there are no bugs to it. We make no guarantees to it either. If you’re getting something from open source, we do not have responsibility for validation of the results. So you use that at your own risk. That’s why many folks will, if they want to use a procedure like that, may test it against a SAS side procedure to see how everything matches up and then can adjust accordingly. We are constrained to a single version of R, but we can upgrade that as times allows. In PROC IML, oh, and the set of R packages, we download those on an as-needed basis, but if you have a set you wish to use, merely send those to the VINCI SAS admins what packages you need, and we will have those downloaded.

In PROC IML, R must be installed on every computer that runs a SAS server. That means on all the nodes, which it is, and they all use the same version of R and all the installed packages. Anything that anyone has asked to be downloaded will be available to you. And as we say, we can download any package desired, and currently we are using R 3.3, but we are in the process of upgrading to R 3.5.

So here are some SAS and R comparisons. Method or module, and we have SAS source and the R target. We can export SAS data to an R data frame. We can export a SAS matrix to an R matrix, a data object to R data and the other way around, an R expression to a data object variable, an R expression to a data object, import dataset from R to a SAS dataset, import a matrix from R to a SAS IML matrix. So that highway can go both ways. Okay? But we can also use it to import under PROC IML since it submits R packages. If you have something you want to use like RIO [phonetic 33:52], that can work under PROC IML too.

And this is just some examples, and I put these up there for folks who just want to use this for reference. Essentially this was a very simple task where we took a SAS help dataset called Class and essentially put it into R, ExportDataSetToR, and we just called it df. And that’s just a very, very simple example, but that’s primarily how you would do most of this.

And here is a matrix example where it’s created and we have our results, and we’ll see all this later also.

And here’s our leader. Here’s our live demo. So let us come back to Enterprise Guide here, and what we did, I'll run through this very quickly. Here is what we had done before. Essentially this is just running SAS IML results and then the same way how it would look in R. A very, very simple example, which I can run very quickly, and essentially that’s what you’re going to get. It just shows you can do both in whatever methods you so choose. And because they’re transferable, if I did all my matrix in SAS and wanted to run something in R, that’s perfectly fine. Or if I have a lot of matrices in R and I want to bring them over to SAS, we showed how to do that before, and so it can go either way. So if you have a lot of R data right now that you want to do analysis in SAS like Enterprise Miner or whatever, you can bring that over, save that to a SAS library, and it’s immediately consumable by SAS products.

Here’s something else we have. If I have any problems with the environment, I wish to test it, there are tools that we can set up. I've set up many programs that essentially will show how everything is working, and I can tune the environment with it. Here’s what we spoke about earlier. I make a silly thing of a SAS dataset, and I essentially just work that out, and I just run this guy. And it shows everything coming out as SAS, but I also moved it into the R world. So that’s as easy as that can be. And this can be a saved dataset. This can be a very large dataset that you have manipulated from CDW or your research project, and this will put it into R if you choose to use R and vice versa. If you had processed mostly in R and now you wish to use SAS, you can go the other way. Okay?

And here’s how we export SAS data R. That could be any libname, any data you’ve created, and it goes to a data frame. And here’s how we have taken work from R, and we put it here into PROC MEANS. [Phone ringing.] My apology for that. So again, we’re going both ways here. Export a SAS dataset to an R data frame, import R data into SAS, and our results. [Unintelligible 37:29] need to run this again would essentially look like that.

So even though this is a very simple example, this is really the guts of how you’re going to move data back and forth between SAS and R. So this allows you to use your analytical tool of choice between R and SAS, or I think, most wisely to use both. Just use one, which either from familiarity, but learn the other and take advantage of strengths. SAS’s strength has far greater strength in data manipulation and memory, and R has a lot of really cool packages and also has things out there that SAS has not developed yet in a packaged form. I would say that R is probably a little more up to date because users are adding to it all the time, I would probably say by about five percent on SAS as far as what’s out there. But let’s be quite honest. You can code in SAS or MATLAB or R or Stata, SPSS, Python, whatever. You can really get the results you are looking for. It’s just a matter of tool preference, and here we work on combining tools.

Now let’s do a little R demo. Here are several very tiny little things I've developed under PROC IML into R, and as I said before, these would run in a linear fashion. Okay? And we can run that if we so chose, and these will go along quite nicely, but we also have the ability, this is expected. There’s a part in here that I put in for error handling. But our results will come out. Here we have SAS code, or I'm sorry, this is R results. And this we only put in so that we can show it to you. It’s not a blind alley. If you have an error in R, it’s going to tell you, and it will also in the log give you what the error is. So for example, if you have a production job that’s a hybrid of SAS and R and you run it, for example, in Gsource, this will tell you that you have an error, and this is something that I made. This is an error that I created. It can be any error you like. You could say there was an error in XYZ module. Please check this code, whatever. And also it tells me when everything is finished. Now, these are very small, so as we were discussing earlier, when you do parallel processing, bigger is better. But I decided to put this into parallel processing, and this brought this up into 10 procedures that would run concurrently. And this will give me the same results.

Now if I had enormous amount of data and I was doing quite a bit of analysis against it, and I decided I wanted to manipulate data in R or even if I kept everything in R, then I can just take it within the grid. And just by saying, going to a program, let’s say I did, and this is what I did with our demo, just by saying analyze program, analyze for grid computing for the grid, and it begins analysis. It will run and actually wrap it up into, [unintelligible 41:20] down to five because I added things before there. But we had five procedures, and now we have code that will run on each one concurrently, and this is as simple as it is. Had the project done. Okay, so even if you’re not running primarily in SAS but you want to break up your R code or any open-source code, you can do it with this method.

We had a few other things here. This was when I was helping a few folks out, and we grid enabled all that also. And now let’s look at some different examples. Here, for example, I can also put in R utility projects too. So if, for example, I say what version of R do I have, what’s available to me, I simply write the commands you’re all familiar with, R dot version, I run this guy, and it gives me all my verbiage about the environment out there, what my platform is, all of it, status, measure my version so to speak, the year it was all installed, yada, yada, yada, language R. So anytime I want to learn what I have out there, this is the way to do it.

Now here’s something that I stole from Mr. John Cashey [phonetic]. You’ll find we do not have at this time, if I run this, okay, you’ll see that this does two things. And we don't have X, we did not put [unintelligible 43:08], so how do we mitigate that if I want to grab? Essentially we just put it out to a plot. And I believe I have open, so I essentially took that, ran it out, and I put it out to a JPEG, and there it is, [unintelligible 43:26]. So I can handle any viewing or any R output in that way, and as I demonstrated earlier with WinSCP, I can move it from the Linux environment back over, or I can also write it out directly to the Windows environment in any form I so choose. So that’s entirely at your behest. So we can do plots there.

One thing we did, here’s query building. In this case I took, I don’t want to do that one. I essentially took ethnicity, ran a quick query, and then took what I had from there. Here I created SAS library, SAS data, and just ran an R. Just ran it all in R to get output again in JPEG. So set it up as we do in SAS, [unintelligible 44:30], the name. This is our preferred way. You can do it either way. You could actually bring the data up and use the query builder, which will give you the options, or when you become more familiar, you can just use something that Kevin Martin developed and maintains called SQL optimal, which is essentially a saved macro variable with optimization attached to it, obviously. So we can run all this. And what did it do for us? What it did was take data out of CDW and essentially PROC into SQL pass-through, so it all ran up there and just returned our results. Then we imported the data, or exported rather, from SAS into class. And what we do then is plot it, and we have that plot up also I believe, and there it is.

So I can use either way. I could have done that quite easily, just as easily in SAS, but we’re just showing the capability of how we can use R also. So let’s look at, here we go. Let’s look at this. Pretty much the same thing. I was just playing around a little bit. And on this one, this was a different way to import data as you can do this also. You can just use other procedures within R if that is your preference and set up your data to R. All right? But if you wanted to do it here and then move it into SAS. We will expand that a little bit. I essentially used R to take R data and create a SAS dataset. So you have two ways to go. You can do this either in SAS. You can do it within R. It’s really what you’re most comfortable with, okay? So that is R. Do we have any questions about R at this point or open source in that regard? Let’s go back to our\_

Rob: Mark, we do have one question regarding R.

Mark Ezzo: Sure.

Rob: Can you use packages that require other software? For example, to use RStan, which is a package in R, for Bayesian models requires Stan software. Is this possible?

Mark Ezzo: If R can call it, all right, then you can do that within SAS. That’s really your only limitation. The only limitation we have in that regard is a lot of the R packages. Are they downloaded down there? If you run it and you don’t have it, then merely send it out to us, and we will download it. Unfortunately, though, it’s not as simple as you doing it yourself because of VA security alone. So if there’s any issue with the package, just let us know, and we can mitigate that for you. So your only limitation really is the limitation in the R language itself.

Now if that’s it, let’s talk about Python and SAS. Python is catching on very strongly, and SAS has really melded with it beautifully. Python is rightfully showing enormous popularity in the market. It is a very powerful open-source program, and honestly appears to be supplanting R in many regards. It does with many sites that I'm aware of. It does have the same open-source issues such as memory, I/O, security, et cetera. 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, due to other important VA ventures like electronic health record management, we don’t have the budget for it. I just mention that in hopes that we get it in the near future. Therefore, we are going to be using something else called SASPy. And you can also submit Python commands from SAS using X command. What is the X command? X command is a system command call, so let’s say I use SAS to set up my data, I save it in a library, and then I can actually do X and call in a Python module to run that and output it and then just pick SAS up again.

Let’s go down. This one, as it says, SASPy. This module provides Python API 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. That is not something I made up. That’s an actual term. The APIs provide interface for the following: Start a SAS session on the same host as Python or a remote host, exchange data between SAS datasets and Panda data frames, use familiar methods such as describe and head to work with data. Additional functionality is machine learning, econometrics, and quality controls are organized in Python classes. And again, as I mentioned earlier, this is really only just accommodating a tool of choice. I think the one advantage that open source has is you’ll see a lot more packages come out a lot more quickly. Now again, are they validated? There have been many cases where people have released packages that weren’t really up to snuff, that they didn't validate or had bugs in it, and I know of some sites and clients that suffered from that because there were small errors in it they did not realize. So many times people will use open source, validate with SAS or whatever method to ensure it. But I do not say that to discourage use of open source, just to be cautious about it. I actually encourage the use of the marriage of SAS and open source.

And what are dependencies? We have to have Python 3X or higher, SAS 9.4, which we have, or SASA Viya 3.1, which we do not have. Hopefully we can get that in the future in a year or so. I see that out there quite a bit. We have to include, that’s really not germane to you folks, we include a Java environment also so SAS can connect it. And as it says at the end, you can connect to SAS on any platform that is supported for the specified SAS releases.

And here is just 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 dataset, you can use the head method. However, if you are an experienced SAS programmer, there might be an occasion in which you prefer to run SAS statements such as PROC PRINT to view the first few rows of a dataset. What that is saying, essentially, is you can vacillate between what you want to use. If there is something that you want to use in Python, just like we have with R, you can use it in Python or you can just switch over and use SAS, or you can combine them, which I think is actually the best method. The magics that are available with the package enable you to bypass Python and submit programming statements to your SAS session. And there’s a meta-macro SAS magic, enables you to submit the contents of a cell to your SAS session. The cell magic executes the content of the cell and returns any result. And that’s just a very small example, PROC PRINT, yada, yada, yada. That’s within Python itself.

If you’re also invoking SASPy methods directly in other cells within the same notebook, you may indicate that expression there, SAS cell, should share the same SAS session by passing your existing session as a parameter, excuse me. A prior cell in the notebook, we have that, importing SAS, sending dataset to SAS as mydata, mydata. So I have done work in Python, and I now have all the data in Python. Now I can put it back in SAS. I can do that for graphing or publishing or whatever. And then you can pass it back and forth to your heart’s content. This way is taking it from Python into SAS, which we can go the other way too. I could spend hours talking about R inclusion, R Python inclusion. Because we’ll still be installing this in our test environment, I’m not actually going to get as deeply into this. I’m more seeing what your interest is of using it. And if it’s anything like R, I'm sure it will be extensive.

And we can also do this directly in IML. The IML magic enables you to submit the content of a cell to your SAS session, yada, yada, yada. Cell magic executes the content of the cell and returns any result. The PROC IML statements and the trailing QUIT statements are submitted automatically. So it takes care of the wrapper for you. Okay? So that’s working with matrices within Python Anaconda, submitting back and forth between SAS.

And there is much, much more to come. I know we’re getting up near the top of the hour. I can take a few moments for questions if we have them. And we have only demonstrated the surface of what we can do with Python and SAS without SAS Viya. Make no mistake, SAS Viya is the far preferred method, but we can utilize this until budget allows us to implement it, if ever. I won't kid you.

And this is our analytical SAS/open-source roadmap. We’re going to upgrade R to 3.5 on all Grid nodes. We’re going to socialize the usage of R within SAS, as I do a few times every day. And then we’re going to implement Python and SASPy in a test location, test it out, validate the usage and best practices. And we’re probably going to ask for some pilot project volunteers, and I would welcome anyone to [unintelligible 54:44]. And then when all is happy, we’re going to put it out on our production grids, both in research and operations. Okay?

So, questions or comments at this point? Or you can send us [unintelligible 54:58].

Rob: Mark we have\_

Mark Ezzo: Yes, sir?

Rob: Okay. We have one question queued up, and it’s a long one, so bear with me, okay?

Mark Ezzo: Fire away.

Rob: There’s a bit of a long explanation after the question. What APIs for VA data to grab non-personally identifiable in aggregate are available to those outside VA for Veteran health research? This is asked because some pilot clinics in commercial healthcare are seeking population health data at county levels much like data shown in the GDX report formats.

Mark Ezzo: Mm-hmm.

Rob: It would be nice to have a GDX format that includes IDC-10 or DRG codes of interest. I'm happy to submit via e-mail regarding a sample of the kind of data as needed.

Mark Ezzo: I think that’s excellent, and that is really something that would be, in my opinion, be extremely useful. Unfortunately, that’s also a matter of policy. So if you would be kind, sir or madam, to submit that e-mail, I will discuss it with the powers that be, and we may do a little tandem test project to validate this and ensure that it is safe and can be used because there is justifiable paranoia about sharing personalized data, believe me. I don’t think we can be too safe in that regard. So why don’t you submit the proposal? Send it to VINCI SAS administrators. We’ll take a look at it, we’ll work out some of the results, and that could be something that I feel would be very valuable to the VA as a whole. Any other\_

Rob: This person is asking for your e-mail, and I think it’s on your next slide.

Mark Ezzo: There it is.

Rob: At this time, that was only question we had pending, but audience members, if you have questions, please feel free to use the questions pane in your GoToWebinar dashboard on the right-hand side of your screen. As I said, Mark, there’s nothing pending right now, so if you have sort of closing comments that you want to make, I bet people will jump in in the meantime.

Mark Ezzo: Sure. Well, I didn't expect many questions, really, because I really, this, I broached a lot of knowledge in this one hour. I mean I could have devoted an hour, hours to any one of these subjects. I just wanted to scratch the surface and show which way we’re going. And in closing comment, I don’t look at a SAS Grid merely as a SAS platform; I look at it as an analytical platform. And we are always looking for ways to expand its usage including other tools, other techniques, and some things like this very excellent proposal where people can, and I encourage this also, send proposals to us what you would like to do, and we will aid you in getting it done, if possible. So we’re here to assist you in all your analytical endeavors and just to make your life easier because this is all very important work that just helps our America’s heroes. So thank you once again, and if there are no questions now, feel free to use the e-mail to send that to us or call me directly, and I would be glad to speak with you folks. You have yourselves\_

Rob: We had one come in.

Mark Ezzo: Sure. Fire away.

Rob: This person asks would you mind sharing/making the code samples available?

Mark Ezzo: Not a bit. I’ll tell you, we’ll put it up in our SharePoint site, and just e-mail me, and I’ll give you the link to the SharePoint site. And I encourage you to use that because Kevin Martin, who I think documents extremely well, has taken quite a bit of time and effort to put forth some excellent documentation that will assist you in most anything that you need out on the SAS Grid. And we enhance that at all times. So yes, I’ll put that on our SharePoint site, and I encourage you all to consume it.

Rob: The person you were speaking with before with the proposal is asking for your phone number.

Mark Ezzo: Okay. That will be, why don’t you send that to e-mail, and I’ll send it back to you, and it will have all my contact information.

Rob: Perfect. Thank you, Mark, for your presentation today. It’s extremely important. And audience members, if you would please stick around for a few moments when the survey comes up after I close the session, we would appreciate your answers. We really do count on them to continue to bring you high-quality Cyberseminars. So thanks again, Mark, and everybody have a good day.

Mark Ezzo: You too. Bye-bye.

[ END OF AUDIO ]