Mark Ezzo: Hello, everyone. Today we're going to be doing an HSR&D Cyberseminar. And the topic is Basic Research and Preparation, and a bit more, some tricks of the trade with the SAS Grid. I'll be presenting it, Mark Ezzo, data scientist, VINCI SAS BI administrator. And, let us commence.

Now, here is the focus of the seminar. We're gonna, we're going to look at how to best approach a project via the SAS Grid, usage of desktop SAS, or more specifically, hopefully, non-usage of desktop SAS, limitations of not using grid resources, increased utilization of SAS Enterprise Guide EG, up here. Available code-generating wizards, and which is, are also a great learning tool. ETL, basic data preparation, and and a little advanced data preparation, also, basic analysis, and some advanced procedures.

We're going to show you running R under SAS, which has become extremely popular. And, many, many folks have converted from RStudio and native R to using the SAS Grid, and R under the SAS Grid. And, to show you why; and, of course, any questions, or comments. And, well, you can do those dynamically, if you like.

Okay. Basic assumptions about having a research project, we assume all paperwork is completed, and approved. And, if that does not happen, then there will be no entries to Active Directory, and we cannot proceed from there. Even if I gave you permissions in the SAS Grid, you could not get to the project data, or the project folders unless you have completed it, and you're in Active Directory.

So to that, Active Directory groups are created and populated with the correct individuals. Usually, it takes, really, provision about a day for everything to permeate down. There is, are, or, what about 50-plus VISNs in the VA? That's, in the world of technology, that is quite a large number. We assume also, data managers have created the appropriate – the appropriated or the appropriate tables of views, your source, and default schemas in SQL Server, and have provisioned you with access to them.

We also have to assume, and we will not get into this after, the SAS Grid, if we do not do this. The SAS administrators have provisioned a study in both metadata and Grid storage. And, hopefully, and this helps quite a bit in the research world, you have developed your hypothesis for acceptance, rejection, even if it's only implicitly.

Essentially, for all of you non-statisticians, you must – it's an objective. You are trying to prove A, that race and gender matter with this, with a, a certain bit of data or, or a protocol, or whatever. So, you have – you have – that's just a, it's just a crude example. So, you have to have something in mind to do. And essentially, it's developed in this physical world as a hypotheses.

Okay, now, this is an idea of a provision study. Once your study is provisioned, both in Active Directory and into the SAS Grid; for example, whenever you'd come up, and you'd bring – you get into the SAS Grid EG, and you go into SAS app, or whichever application server you want to go into, you're going to see things like this. I use, this is Tim Trautman's bogus project here, and I just use an example: org Precipio \_\_\_\_\_ [00:03:43] given source schema.

Now, these will come up. And you will see in the metadata, the SAS metadata world, these things that you're only provisioned for. Now, this also can be anything in the CDW. For example, you may have CDW sstaff [PH] or CDW Pool. You'll see all of that come in also, so have no fear there. But that, this must be done before we, we can provision this for you. And what are the advantages of using the SAS Grid?

We have a, an extremely robust environment, three compute nodes per Grid, operations, and research. The usage, our research is about two to one to operations, which is to be expected. And good news is that we have provisioned, and have purchased, and we have to install. We're going to install new blades. And they are going to have twice the power of the current ones. We currently have a lot of storage in the, in the, in the Windows world for a project.

You have, I believe only 300 gigabytes on the project voltage of p, or p2 drive. Now, on the SAS Grid in total, we have 704 terabytes of storage. And obviously, you you can, you can have at it there to your heart's content. However, we do police it. If we notice that the individuals are, are consuming 20, or 30, 40 terabytes in a project, we will, we will question you as to that fact. And either compress the data for you, archive the data, or ask you to delete it if it's no longer in use.

Another great asset of the SAS Grid is greater memory allocation per job. We can, we have several application servers, which I will show you, and with various amounts of memory. For example, we have an R app, and a SAS EM for Enterprise Miner app. And they have essentially four times the memory of SAS app. And the reason for that is because we use open source in there.

And as you know, some of the limitations of open source is that they're completely memory intensive. They don't have the capacity to to use a hybrid model of memory and I/O that SAS can do. Maybe at some time it it will improve with that respect. We are in an enclosed Linux network, infinitely less interruptions allowing completion of jobs reliably. That means that we have less to do within the Grid itself with the Windows network handshake that most people need.

Now, we have that with Enterprise Guide. And if you leave your applications up too long, or overnight, there is a chance that you might get interrupted Enterprise Guide, which is why we encourage using SAS batch, and SASGSUB for any overnight jobs, or very long jobs. We have, right now, and that this may, or may not increase, 3.7 terabytes of workspace per compute note. That, that is what I alluded to previously.

We, when SAS feels that you're running out of memory, or that your your procedure that you're doing is using too much memory, it will break it all off. For example, if you're doing a SELECT DISTINCT, it may put that in the I/O world rather than the memory world. What is the advantage of that? The advantage is that you're not over taxing memory, and you're not slowing down jobs, or even a halting jobs of you, or other individuals.

There is additional security. Not only do we have the Active Directory security, but we provisioned people in SAS metadata. And we can, we can control what you can see or not see within the Grid. And we can make groups and add appropriate users per the designation of the, of the PI. And that has been very effective.

It's very secure model. It adds an additional level of security. And of course, we're using the SAS's award winning UIs, user interfaces, both the SAS Enterprise Guide, and SAS Enterprise Miner.

I'm probably not going to show you too much of Enterprise Miner today, probably not a bit at all. I, I've given seminars with that, and it takes quite a while to, to really go through what's necessary for that. But you'd, let's just say that it is something that is icon driven. Essentially, you pass, you select icons, put them into your, into your display, and fill up the parameters. And off you go. I'd gladly do an Enterprise Miner seminar at some other time.

And we also have, which is another huge advantage, dedicated SAS analytics support staff. The advantages of SAS Grid versus what is essentially since SAS 9 point whatever came into being, it is obsolete, and it should be deprecated. SAS, the desktop SAS is obsolete. It has not been updated since the SAS 9.x, essentially. It will never have another update. And every time we get Windows updates, which happen frequently, then it becomes even more deprecated, meaning that you're going to run into more and more issues.

It's not nearly as robust. It does not nearly have the, the memory that we have on the Grid. It's just, it's just the wrong avenue to be in. SAS, I, honestly, I wish they would allow it. But SAS allows desktops, SAS as a courtesy. And I think eventually we are going to deprecate it within the VA

It has very, very few programming aids as opposed to SAS Enterprise Guide. So you're essentially, just mostly programming – is very, it has very few wizards, though there are a few. And for budgetary purposes on desktop SAS, there are SAS stat procedures available. Meaning all you can do are are very basic things, maybe a little bit of data preparation. You cannot do any regressions. You cannot do any survival tests or analyses, et cetera. You're essentially have Proc MEANS, PROC FREQ, PROC UNIVARIATE. And that's about as elegant as you're going to get.

So in a nutshell, what we're, are coming to as a conclusion; don't use it, if you can avoid it. And if you feel you're compelled to use it, please contact the VINCI SAS administrators, and we will show you how easy it is to migrate to the Grid world. Advantages of SAS Grid versus open source, now when I say this I am not deprecating open source in any, way, shape or form. I believe in open source I think SAS, and R, and Python, and Julia, and Java are all tools you should be throwing into your toolbox.

It's, I've heard people about which one is better. And to me, I I think that's really an argument without any merit because you, we have the ability to use them all. Use what you're most comfortable with and what fits the, the the mission of what you're trying to do.

However, having said that, the SAS have the advantage of the SAS Grid versus a RCBO [PH] or or native R, or Jupyter Notebooks. It's the far more stable infrastructure and environment, and it has many more bells, and whistles, and will allow you to do much more analysis, and data preparation than either R or Python at this point.

It's an enterprise product. It can handle enormous amounts of data more easily. Much of what I see being done in the SAS Grid world these days is a lot of data preparation, and analysis in the SAS. And sometimes people will do analysis in R. And I see a preference for using R graphics, but I think that's because most people don't really understand the SAS graph wizard.

And having said that, there is nothing wrong with R graphics, they're very good, and they're not that difficult to use. SAS, as we've discussed previously, is not memory reliant. Of course, you need memory but if you run out of memory in a procedure, then it goes into the I/O world. And to really slow down the SAS Grid, you really have to to put your mind to it.

I'm not saying it has never happened. But when we've had some users on, maybe about 20 users who are running some aberrant R programs, or trying to do some very sophisticated PHREG analysis, we can slow down. But the the advantage of that is, if node, one of the nodes is slow, then the SAS middleware, SAS will switch you to another compute node.

And unlike Windows products, which are linear, if we lose one of our nodes, then the Grid, we do not lose the Grid. We essentially will reduce to two nodes, but we're still running. It will be a little bit slower, but it, it really takes a little – it almost takes unplugging the environment to stop the SAS Grid. We have proprietary technical assistance, meaning we can call, and you can call SAS technical issues. We'll supply you with a number, and their people will reply to you within 24 hours, or earlier.

And that's very, very useful. I I would say that as far as the industry, and as far, and it, as the analytical industry, it's probably the best help, their technical assistance, we can get. It's very easy to use because we use SAS EG. And there are wizards, of course, in the open source world, I I believe SAS EG with their wizards.

And the the essential UI, which we will see, is is much easier. We have the implicit wizards, which allow you to, without knowing SAS, just filling in parameters of what you want to do. And one of the more beautiful things of it is it can create code.. And you can save that code, and either A, learn how to use, and code natively with SAS by viewing the code.

Or B, once you have a successful task completed, you can use that in a macro, and just incorporate it into anything else that you're doing, into your whole process. And we'll see that as we go along. We can combine SAS with open source successfully. That's been very popular in in the VINCI world. And it's increasingly so, and that's one of the reasons why we have doubled our resources.

And hopefully, and I'll allude to this later, we could get, if we get our SAS Viya product in very soon, which will allow the the hardcore data scientists to be able to combine within one venue, SAS, and R, and Python, and whatever else you wish to use. There are probably more coding examples available because SAS has been around since the '70s.

And I do not know of anything that you are trying to do; if you go out, and Google, and get into a SAS blog, or or a SAS help area, or technical support. I find it almost beyond the realm of possibility not to find a solution very quickly.

A massive, it has a massive user, a user community. Though this is also true of most open source. So everyone's out there helping each other. You can get your SAS expertise, or your Python expertise just going out into the Google world. And I say, et cetera, because we shall display it.

Now, let's discuss ETL your data preparation. Whenever you're dealing with CDW, always, and I repeat, always use PROC SQL pass-through when accessing SAS SQL Server tables unless they are very small. If you, say, get into the SAS app, and expand your libraries, and get it into your project, you may – for example, inpatient data, or or chem lab data, it can be very voluminous.

And if you double click on that, and you have not set your parameters correctly in the SAS EG, you can actually wait quite a while for that to properly adjudicate. And it, it will consume a lot of resources. So if you're doing a DATA step with, combined with, like, set, and your LIBNAME points directly to a table SQL Server, the detriment to that is before any of the calling happens, any of your program statements, or utilized, it must bring the entire table or tables down into the SAS work area. As opposed to pass-through, which allows you to send your programs, your SQL programs up to the, to SQL Server itself.

And it processes up there, and only returns what you have programmed, and requested back to either your permanent SAS storage, or your work SAS storage. Now, this can cut off hours of anything that you are trying to do. And it's just more efficient all the way around. It returns only what you specify. It does not stress the system. And when the system gets stressed, it stresses other folks, also.

There are wizards that can create templates and teach you for ETL. There are many, many data wizards. You can do joining, merging, setting, calling whatever. And we'll look at that. And customization can be added to any generated code. Essentially, once you have the code, and you run it, you own it, and you can now start to copy portions of it, change portions, change the sources in it like your LIBNAMEs, et cetera.

It's essentially at that point, you own it, and you can just put it into the program area, and do with it as you will. Now, having said all this, please consider space considerations, and only say what is necessary. And I understand that there will be times where you were going to want to save 1, 2, 3, 4 datasets as you're working out the final data set for analysis. And that's fine. That's that's actually just an iterative part of research itself.

But then when these are completed, what you want to do is delete those, and just save the final dataset. Now, when you're doing a large ETL process, and even within your program, you are having huge work datasets. Once they are completed within the program itself, you can run something, it's called PROC DELETE. And that will delete that unnecessary work data set, and free up the resources for you to consume further on.

So if you're running something, and and it, by the end of the program, it would be a couple of terabytes of data in workspace, but you really only need 50 gigabytes. It's very, very prudent, and will help you if you just put PROC DELETE in, in the interim. And have the ability to delete that data, your code will run faster, and that system will be less stressed.

We can archive for you. If you say I, I have this data, I'm not going to be using it. I may not be using it for quite a while, but I want to save it, why, it's under peer review. Or we may revisit this process within a year or two. We have the ability to archive it, and back it up, and retain it for you.

And now, let us do a quick demonstration. And I have to get out of this mode, and go up to here. And this essentially is – we're in the operations grid right now. Like, let me show you what that will look like. This is what your, your profile should appear \_\_\_\_\_ [00:19:32]. I put operations Grid up there. And you just say it's remote, and you'd put in a description, if you like. But it's obvious.

Remote, and this is what you have to connect to, that's the metadata server. So in operations, it's VHA, CDW, DWH SAS MD1 [PH]. For research, it's VHA, CDW, DWH SAS MD2 [PH]. But we're in the opera missions world, so we'll just do operations for now. And all you need to do is click, "Use integrated window authentication." In VA speak, that is your PIB card. And it's essentially, it's a one-stop shop.

And it, you can, you will get into either VINCI, or or the operations client servers using PIB. And this just extends all the way down. And as we know, most people don't use the password. They don't even know their passwords at this point. So that's your profile. And that's how it should appear before you're able to use the SAS Grid.

Okay. And, of course, you can set up many, many profiles. You could even set up a profile locally. And as you can see, you can run things locally. And, if you, if what you have is completely on your Windows drive, you could just use the local application server, and run there.

Now, as I said earlier, we have several application servers, and the main ones are, as we see here, local for the local client. R app, this, what that is, that is – all of these application servers are virtual, they are, I create these internally, and set them up, and set up the security form, and the configuration.

Now, R app is R compliant. And if you're running R scripts on here, you need to use either R app, or SAS EM, Enterprise Miner. And these are both memory enhanced because, of course, as as you all know, open source is far more memory intensive. And the main one that everyone has access to is going to be SAS app. That has less, that has less memory, but it's, it's still quite robust, have no fear.

And as you can see, when you get in these, you'll see the green checkmark. And files and libraries will appear. Now if, for example, I click under R app libraries, it's going to show me all of the libraries that are available to me. And as I said to you previously, we have libraries even in CDW, not just project libraries. Now, how do you, how do you understand the properties of a library?

All you merely need to do is right click, go to properties, and it will show you all of the connections information. You'll see an explanation here, general dimension tables for CDW. And it shows you it's on A01, the server name, R app. Of course, we're running SQL Server. And it shows you the options like the schema here is Dim. So when you, if you're going to do Proc SQL pass-through, you would want to say, and you want to use a table, for example, you're going to have to say Dim dot.

And it's, also, shows you that we have a very large read and insert buffers. We defer the connection until we need it. The connection is global, and a few other things. And then if you want to know what the name of the library is in SAS, you just come down a little bit. And it says library of CDW underscore Dim [PH].

Now there are, for example, smaller tables in the CDW and the dim world. And you could use your LIBNAMEs there. But in most projects except for, like cohort, and very, very small tables you may create in your default area, that most of the source data is quite large. So I don't, I do not encourage using a DATA step and LIBNAME in that regard.

So that's how you can get some information on the tables. You see that we have several libraries here. And in this case, we show certain things like the SAS library, SAS help, and SAS user. The WORKLIB, so if you create anything, it will, it will be in the WORKLIB. In SAS user, you'll, you'll see nothing unless you create it there. And we, we only see what's available.

Now, the other bit under it is files. So if we expand files, it shows us essentially the main areas, which are admin; Grid share, which you will have access to for your your login only. And that's where we run a lot of the SAS batch.

And so we have Grid share, and if I open that up, you'll see that there are a lot of people who have used it, including myself. And that's, that is the batch world. Now we can also see that we have ops. And you know that if you have an ops project, you just go to wherever it is for, et cetera. Now, that's a bit different in the research world. And we, we can have a view of that a little bit later.

But there, they are all the ops projects under there. But regardless, if you want to see your files, regardless of the venue, you can go, and look at the files. Now, having said that, you may also wish to use WinSCP, which is an excellent product. And it allows you to – essentially, the left side is Windows, and the right side is, is Linux.

So by, for example, logon here, and I just put in – in my case I'm on my admin account. But all you merely have to do is type this in, OITSHCEZOM0 [PH]. Hit okay, and in this case, it's asking for a password. I take that back. I am not on my – this, this shows a good lesson, I am not in my domain world. I am actually in – I'm sorry, in my admin world, I am in my domain account.

So all you need to do is put this in it, hit okay. And it comes up. And this is the area where I make a lot of the Grid reports. But the the cool thing about it is, on the left side, if I wanted to, for example, move a a PDF file over, I just merely need to do this. If I wanted to move something back, just slide the other way.

So it's a point and click environment and very, very useful. And you can, you can open things up. And you can, it's, it's, I could, I could spend probably an hour just showing you all of the things we can do with that. Most of you are familiar. And if you have problems configuring in, the SAS administrators will gladly give you a hand in that regard. So let's take that out.

Now we – this is, this essentially is an Enterprise Guide project. And what it has, it has process flow, and several programs that we have created under the process flow. And as you see, it shows icons. And I can actually start somewhere, and just say run from here, and and it, it would. You can do it a few ways. That, for example, some of these are just merely programs. And I'll show you, show you these as we go.

You can also set your options up here. And this is very powerful, your startup options. You check for updates in your configuration. General, project, and process flow, you control this. If you have an autoexec process, you could just click, automatically run that when I start. And an autoexec process, maybe something is where you configure your environment, and you set up your LIBNAMEs. You set up macros, whatever.

Auto recovery, because we do a lot of recovery here in case something happens. We enable auto recovery. So if if the server dies, or you incorrectly come out, or there are network interruption, it's going to save it to an area. And you can see where that area is here. Now, results, this is very important. You can set up your results, which happen automatically. I choose to always come out with HTML, SAS's report.

For now, I can also come out with a PDF, so I had to, I do nothing as far as setting these up. I can also do rtf, which is essentially Word for now. I could do Excel, I could do a listing, I could even do PowerPoint. And I could just set them here. Or I can use the SAS ODS editor, and that stands for Output Delivery System, and just put code in, in that, in that regard, and I would be good to go. Now, display results, display tab2 result. And you'll, we'll see that later.

That's, essentially you have a tab of code, log, and results, and graphics, et cetera. And I can say how I want, how I want it to go. I can move focus to tab, it is displayed. It displays SAS log when errors occur, absolutely. Change SAS icon when warnings occur, show generated wrap, code wrapper, et cetera. I'll give you time to have a look at that.

Now, I can also set and control, for example, we're using HTML, HTML5, which is a new HTML. Sometimes depending, you may have a little problem with HTML5, so you can switch that back to just HTML, and you can control the style of how it comes out. I just use HTML below. You can also configure your SAS report, and the same thing.

You can figure any of those we saw under there. You can configure your graphs, default, best fit for SAS, that's passed aside, et cetera. And you can make a stored process, if you like. It can output results with stored process. Now what, what is, what is the stored process? It is essentially code that you create that others can consume.

You can essentially publish an icon in your project area. And say, let's say you're doing a model, and you have all of the ETL in there. And you've had all of the configuration like the LIBNAMEs, et cetera. You save it as a stored process, and others within your group, or whoever you decide to share it with are able to access that stored process in the SAS system.

And we also, let's look at data. This is dated, and how we set up data, and how you can control it with default columns, yada-yada-yada. I will take a moment to review that. And very important is performance. If I have, and this, this is format data locally applicable. I always set mine the largest value found within the data.

Now this is incorrect, always set to maximum numbers of rows, and set to determine data. Actually, that's in importing data. I always like to say testing. Maximum number of rows in a data Grid with this SAS access, I set that to 10,000. That's actually a little bit high. But 10,000 is very manageable, you can set it to 1,000, 5,000, whatever. And you can unlock open data, enter an activity, and that's in minutes.

I'm going to call open data before running code, or automatically refresh open data after running code. All of these are very powerful and very necessary. And then we have query, you can control your queries here. Attempt to join tables in query, the value of query whenever changes are made.

Now, this is very useful, have a look at this. And one thing I'll say about queries, you have two types of queries. They're, they're all SAS queries, but you have queries and in SAS data. And say, you can set up libraries to Excel or or any Windows product.

And then you have queries over to SQL Server, too. So be careful when you, when you run your queries. But this will, this will control how your queries look, and what your queries are doing. All updated, we really don't use much of that.

Here are tasks. And it just shows you. I like to say use Proc SQL instead of Proc SORT for sorting in task, Proc SQL is a little bit more efficient in my mind. And include SAS procedure titles and results, so essentially, you know what you're doing, what, where it came from. And then you're allowed, you can have SAS programs, which means you can insert custom code before or after. Initialize Grid, if available, we do that automatically. So you really don't have to worry about that.

There's merging tools, version control, you get integration, for example. But you can do an operation if not too much of research. We don't allow that, the firewall is not allowed. We have security and we mask our credentials. We can do a little bit of LIB administration. The transfer mode is binary as opposed to tech. Outgoing, an e-mail SMTP, with that we configure automatically.

As you know, you cannot e-mail from VINCI, but you can e-mail from operations. Application logging, if you still want to set that up. So that's your, that are your options within Tools, and I very strongly suggest you take a look at those, and configure them as you like. Now, I always set my end viewing – I always set my program, for example, just to… If I do this, and I say view; I always say standard, which means I I deal with one, one for code, and one for lock.

Now, it comes in, and I wish it wouldn't. It actually comes into vertical split. I personally don't like that because I don't like to have the code for the log cut off. So I always set it to standard. But having said that, feel free to use anything you'd like. And that essentially is how we go through, and that's Enterprise Guide. That's a pretty good primer for it.

Now, let's go back to our presentation, and let's put this back in. And, let us continue down. Now, one of the powerful things in SAS, and and for SQL Server is the use of temporary tables. What does a temporary table do? The temporary table allows you to take SAS data, and move it up, and it adds a temporary table.

And and the key word there is temporary. When you end your session, when you end you’re at SAS session, the temp tablespace up on SQL Server disappears, it's not saved. If you want to save it, you'll have to write, you'll have to save it to a writable schema. And in in the research world, that would be your default schema. You can save many things up there. And therefore, you can use those as the tables.

But if you just want to use something on the momentary fly, like, I, in cd work as we have here, this is something I actually did for a user. Or we did simultaneously, pardon me. She created a program called Hip Fractures [PH] with parameters, and then we, we set it up, a LIBNAME. And I, I'll show you all this in actual code.

This is the code we had brought up earlier. And if you look at this, this is a very easy way. And this method works in any remote database. So we just set up a, a a LIBNAME, SQL Server. We have our DATASOURCE, and DATASOURCE is in the odbc ini file. And it essentially tells SAS, and Linux world where the data is.

Let's start with what server resides on, what the database name is, and a few other, a few other items. But without this, you cannot connect to SQL Server. And in this case, we just put in a simple rebuff of 5,000, sirP [PH] up of 5,000. But the really important aspect of this, all you need to say is DBMS temp equals yes. Now that gives me the ability to create a temp table in the area I define. In this case it's CDW work on CDW, VHA CDWA 01 [PH]. And then I just say data, AAAup1 [PH], which is my LIBNAME.

And I'd call it ICD10km. And I set, and I set it as the hip fracture thing I created here. So having now have a lead table, I have a temporary table up there. And I can run Proc SQL against it. And I create a a temporary work table called ICD10 underscore hip, where I'm selecting a, just the ICD10 code from here; my SAS data, which I've made a temp table.

And I interjoin it in the CDW, CW dim ICD10 on ICD10 code from my temporary table I created connecting to ICD10 code up in CDW. So I essentially have a lead table where I can limit what I want to see, and connect it to production data, I can, the CDW data. And here I am, I now have a work data, hip10, and I can also do this with something else.

Create a table, \_\_\_\_\_ [00:38:12] fractures with all of the information I'm looking for. Again, from this one, delete table interjoin on V diagnoses, and that anyone who has ever worked with that knows the size of it. So any lead table or any way you can call the data before you're downloading is extremely useful, and effective. And so we run this, and we output our data here. Okay?

That's version one. Now, let's get – there' is another version, another way to do this. We'll page down, and we create the temporary table. And this is a little bit, it's essentially the same concept but a bit different. I am saying when I create this, Aup [PH] , I am creating – this should, actually, this is an error. It should be AAAup, and I creating something called MySites. And all it is, I just keeping from my code, from SAS code, station 3A.

So if we come back down, and we go back to our process flow, and I look at temporary table 2, you can see I'm creating. And let's change that now. This, I stole the code. I just, probably because I just forgot to change it. So if I say AAA up1, I'm creating a temporary table called MySites, which allows me to do as we did before.

And in this case, I'm doing this within pass-through. Then I create a pass-through tunnel which I could just call simply tunnel, to CDW work. It's, pardon me, DATASOURCE CDWA 01, which is the same as up here. And as, folks, most of you know, Kevin Martin developed something called, which is very useful, SQL optimal, which is a, a global macro parameter.

And if you include that in your program, it sets it up so you, it gives you the most efficient environment to do SQL pass-throughs. And in this case we're creating a table worked out test. And you can do a lot within, in these tables. We can create, and I can put where clauses. Where patient, in this case SID, not equal negative one, and patient SID not equal to zero.

I could have put just, just greater than zero. And patient SID is not missing. And it would be as select, now we go to our tunnel. And let me reiterate that anything that you run in SQL Server SSMS, all you really need to do is put this wrapper on it, drop the code in, and it will run for you. And we select our variables. And to run against other tables, that's essentially, is regarding, and The CDW work area, and then chem lab panel, SC, we do a a join outpatient visit, SC, and a join Dim, location, and then stop code.

And of the final is, I only want to see it for certain stations, so we've connect to MySites. So all this is done, and it's just fill this in for MySites. And I put my normal work codes in, and I'm done. Now, that lead table, I'm doing it with pass-through, will, will remove, in this case, it, it would probably removed 20-some hours of work when it, when it was done just with data steps, and, and set statements.

So extremely powerful, there's two ways to do it. And if you ever have any questions, you can contact me, the, at the – pardon me – via VINCI SAS administrators, and we'll be glad to to chat with you, and help you out. Okay, so let us proceed. And, we come, yeah, and that's what I just showed you within the code.

And now we can schedule an EG. Now, you can schedule entire projects to run, and it's really just as simple as – I'm gonna take this out of the presentation because we're always jumping back and forth. If I have something here, and I just want to say through the, the process flow, and merely, if that opens. And I'm going to this start page, and save, and \_\_\_\_\_ [00:42:53], yada-yada-yada.

Or let's see, with this, we'll count up. Now, I have to save this as a project. And let me try something else. No and I don't know if I want to do that with time at the moment. Let's just say it's very simple that, in EG, you need merely – and in this case, this is not the project that I was looking for.

You merely right click on the project and then, and just it'd say schedule, and a wizard will come up. And if I have time, I'll show you that very quickly. And speaking of wizards, I come down, we have many wizards are available. You can customize them to create a template for macros, et cetera.

And let's do a quick demo. Now, for wizards when you have everything here. There's ways to get it. You can either say view, task, which our other way, you saw we removed this. So if you don't see them there, just say view, hit tasks, and you see your wizards here. And this can do many things.

We have ANOVA capability, control charts, data, data mining, and describe graph, multivariate, PARETO free for the economists, PARETO optimality, regression, survival, time series, tools. So let me move over to the other world, just let me find my other server.

And in this case \_\_\_\_\_ [00:44:29] and then we can go down here. And I'm going to sign in as the, an administrator account because I have things open already. You folks are probably more accustomed to remote desktop. This is essentially my remote desktop.

It will take a moment for the, for everything to come up, meaning all of the, all your PIB information. And then while we're waiting for that, here are some of the options. You can see examples for data. That allows you to append tables, compared data, copy files, create formats, create format out of the data set, data set attributes, delete data sets, and formats, downloading data files to PC, filter, and sort.

Anything you can imagine you can do with data, you can do here. You can even do, build a query, you can do random sampling of your data. Essentially, it's a PROC SURVEYSELECT. Rank your data, source your data, split the columns up, it's extremely powerful. Upload the cache, [PH] which we don't have yet when, but we will when we connect to to Azure.

And while we're connecting here, let me do this. And let me wait for, and login attempt. I'll tell you it's \_\_\_\_\_ [00:45:46]. So while we're waiting for that, let's look a bit more. You can do data mining. You can do model scoring. Rapid predictive modeler, essentially, that is very rapid. It's, it's like an ran application from predictive modeling. And I could spend an hour on that alone – recency, frequency, and et cetera, et cetera, et cetera.

Now, you can also do very simple descriptive statistics which you're all aware of. And but, and there's a, and you can do all of the statistical. Now on the graphing, I know a lot of people, and one of their largest complaints is about SAS is the graphics. And if you're going to code it, I heartfeltly agree with you. But because we have wizards here, and we can show all that, the wizards will produce everything for you.

All of these here, for example, I go to my program. And I just say – let's say I run this code, and I'm making a very simple work DATA step of just where the stations are less than. I forget what I had, 458, 463. Okay. Then I come back to my process. And it comes out to agency, and then I do some statistics against them.

So if I open this, this is what it would run. And all that was created via the wizard. And it shows me my code that I created just by going to summary stats and filling in the parameters. And in this case, just that, that was in the script of summary statistics wizard here. So I come into this. It will come up, and I just say I want to choose agency as my data.

We opened, and I see it's confirmed. I'm gonna be doing this on SAS app. It's work data. It's called agency. We go to the next. I can just drag, and drop things over, and run. I'm not going to, in a matter of time, I only 15 minutes. I I won't go through the whole thing. But this is as easy, that it slides you. It will prompt you. It will, it'll, it'll actually assist you.

And if you're trying to do something that it is not feasible, it will stop you. So that's summary statistics. And this is the data that it creates. And that this is a results. So you, if you have any questions about how to do something, then all you need to do is go into the wizard. Now that was – let's come back here, summary stats.

And from the same area, I created a simple bar chart. It's an ugly bar chart, but it's a simple one. And again, it's done with a wizard, and it creates code for me. So if I'm not sure how to do it, I just go into the wizard, and I run it. And I have my code and I can learn from it or I can incorporate that.

Now, I can modify the task. I just come here, and go next. I have everything in in here, and I I can just change accordingly. I can say agency. I could, I can just say, like, station I could change that to. And I could stack it by agency, and change that, and the I go to next. And it just showed me a little bit of legend stuff, finish, bar chart. And off we go.

That's another ugly one, but I'm just, I just wanted to show you how easy it is to change things with the wizards. So any questions you have of anything within SAS, you'll find within the tasks here. And of course, you're more than welcome, they, to give us a call. Now, coming back to the process flow again, you can also do line plots. And we'll just look at this very quickly.

And there it is, you can run a simple line plot. Again, it shows you the code. And if I wanted, I wanted to, I can just come back, and modify the tasks. It's all that, it's really immaterial what you're doing. It's all really the same structure of how to proceed. So what have, we've shown you how you can do against data.

We ran a program to get data, simple descriptive statistics, a simple bar chart, a simple plot. And how it's easy to, to just go in by clicking on it, and hitting modify, and outputting reports, which I could then download for publication if I so chose. And I can control the output quite easily within the code itself.

So I very, very strongly urge you to use wizards when you have any questions. They're extremely powerful. And I, I'll give you a good example. When we had a lot of people go from, from certain products; like, SPSS is a, is a good example. When we are deprecating it because of the cost, and they converted over to SAS Grid, a lot of the SPSS folks were concerned about programming.

And we showed them how to use the wizards, et cetera. And that, believe me, that alleviated quite a bit of stress, and they became very functional programmers very quickly. So let's go down, and graph wizards, which we have showed you, also available. And generating results from publication of code for use in programs, that's graphs via wizards.

It's simplicity itself, as I've shown you, and can be accomplished completely without coding. No need to learn a graphics language, and believe me, you don't want to. And that we've already had the demo. And let's go, and let's just talk about our current capability right now. We have centralized shared workload management and why it matters. Effectively manage jobs and users prioritizations, compliance, and auditing.

Saying when you submit a job to the Grid, the Grid middleware will select the, the the most viable compute node for what you're trying to do and usually, that means the one that is least stressed at the moment. But that also will change, that, it's a little more elaborate than that, a little more elegant than that.

We have high availability and high availability in the tech world means that if one node fails, it, your work will fall over to another node. Or the other nodes are available, et cetera, and so, well, you're not losing capacity. We don't lose, we lose some capacity, but we don't lose functionality. It's just distributed processing, meaning we have job slots which we can use on – you can run parts of your job on all three nodes. And that's improved performance, meet changing demands.

We can grow incrementally meaning we, all we need to do is add new nodes. And we leverage commodity hardware, which reduce costs, which is near, and dear to everyone's heart. And let's, let's look at R for a little bit. Now, in this one, and we go back to our process flow. And we come up; let's go up here. And it's we're going to \_\_\_\_\_ [00:53:35]. Come on, up you go. Here we are.

This is a, an example with with R, a very simple example. In fact, I will even show you more. Let's bring up some, a new version SAS Enterprise Guide. Again, we'll hook up to operations. Well, then I click on that, everything just for demo purpose. But while we're waiting for all of that to come in, you'll see, this is very simple. And the reason all of this is.

For example, the first one shows Proc IML. And we're just setting up a, a comparison of matrix. So this is a matrix under Proc IML. And this is submitting our code and showing the same matrix. Now, you can only run R under Proc IML. But all you really need to have is this submit block. We're saying submit R. And you can only do R on R app and SAS EM.

So if I'm running this one on SAS EM, that's completely compliant. All I need to do is hit run. And here I am. Now, just this, maybe a little bit more elaborate, so when this adjudicates itself it shows that we're busy. So this is how easy it is to to run R code. And honestly, I only have eight minutes, so I may have to curtail that demo a little bit.

But let's proceed, and let's talk a little bit about the future. The future is what we're working with now. We have, we have a POC going. It's, that's gone very well as far as we've had users involved, what they can use. And then we we have a few data scientists, and they're, they're very happy to see that you can use SAS via, and incorporate SAS data.

Because one is very fluent in SAS, and very fluent also in open source. One is very fluent in open source and lightly fluent in SAS. And and they're both able to use it, and it enhances both abilities. So SAS Viya architecture is enabled by a wide range of capabilities. And it's an extension of the success of the SAS Grid. In the cloud world, you pay for everything.

If you have stored data up there, and, and, you know, the data we deal here, and you're talking about terabytes, you're paying for that every second you store it. So the, R best model is a hybrid model where we do a lot of the ETL, and a lot of the work down in this, in the SAS Grid world. And then we can, we can export our results of the data like they to use with SAS Viya. Or you can use it with Power BI or whatever other tool is up there.

So what, what I was talking about was how using 9.4 and SAS Viya is, and it's, it enhances. It enhances from the current strategy. Each is designed to solve different use cases; they coexist very well. Data, models, and codes can be accessed or accessed directly. In SAS 9.4m7, it's just another data engine. SAS is an answer. And just, you use Python, Spark, RS, Git, whatever, and SAS together within one, one area.

SAS can augment open source, you can increase your productivity, leverage your assets, people, and platform. You bring power SAS open source, you create deployable analytics, and goals to embrace, and extend. So it's not either, or, or mutual exclusivity. It is essentially total inclusivity of open source and SAS. You have the power of the SAS infrastructure, which in my mind is superior to Python or or anything out there in open source.

And then you have the ability to have – SAS being closed sourced, and testing, they don't get some of the newer things out to market as quickly as open source does. So we have the best of both worlds here. And for more information, extending R, if you download these, there are a lot of things, videos, YouTube videos. We have blogs, articles, open source integration, node installation, cheat sheet, and Enterprise Miner.

And there is information on GitHub, which if, and I don't see how they can avoid it, if we're gonna go to the Azure world, and research. You can do this now in operations, get to GitHub. How to use GitHub, or SAS \_\_\_\_\_ [00:58:09], and GitHub, integration with Jupyter Notebooks for all you people who are familiar with Python.

Sample codes of SAS machine learning and the best machine learning you can do in a SAS world, and and arguably, even in the, in the open source world in in Enterprise Miner. But that would be a seminar for another day. And I'd be, I have given seminars on machine learning. I'd be glad to do so, and show you open source combined with machine learning. I'm hoping the next one I can do in the Viya world.

And so you see machine learning, and Enterprise Miner, flow diagrams. And resources offered, this VINCI SAS admins offer online training for you, or your group, and you just contact us this way. And for specific questions, or for example, if you cannot get into VINCI to use SAS, many of those come to us, and sometimes it's as simple as us just logging you out of a disconnected session.

But if you really can't get in, we're not the people to talk to. You want to talk to VINCI, put in a ticket to VINCI at VA dot gov. And in my signature and I've included here, I have a link to our our SharePoint site that has all the SAS documentation you would ever want to see, both internally, the vast majority was written by Kevin Martin.

And he, he did an excellent job. I've included some things up there. And we include some things from SAS into, SAS Institute itself. And just click on the link in my signature, and it will take you to all the available documentation. So a good griding everyone, thank you for attending. And just contact the VINCI SAS administrators. Now I can entertain any questions or comments.

Rob: Thanks, Mark. We do have a few questions, if you have a couple of minutes extra, we could go over. That's okay with you. Right?

Mark Ezzo: Please do.

Rob: Okay so attendees when I close the webinar, please, if you have to leave early, please take a few minutes to answer the questions that pop up. But not waste any more time. Can we use SAS Grid for sample size estimation?

Mark Ezzo: Yes, as you see you can. There's something for Proc, called PROC SURVEYSELECT. But as I showed you earlier if you just go into data here, and I go to my wizards; and you'll see that it has sampling. Where was sampling at? Which, which wizard am I in?

Yeah, if I just go into describe, you'll see that we have – not perhaps – summary details, summary, summary \_\_\_\_\_ [01:00:59]. Yes. It it does get, offer sampling. I think that's up in data, but please ask your next question while I find it.

Rob: The next question, this person asks, how do you release slash delete temp SQL tables if created by SAS? Does SAS clean up automatically at end of run?

Mark Ezzo: Yes once you end your session, the temp tables will disappear. It's, it, it's the same as your work data. If I end my my SAS session, all my work data in SAS disappears; and conversely, all the data up in SQL Server, all of your temps tables disappear also. Here's the random sampler, right there. It's under data, just click random sample, and the wizard comes up.

It just allows you to, before you select your data, and just follow the wizards all the way through. And it shows you the kind of methods, simple, you can just narrow it down. Unrestricted duplicates allowed, which is unusual but it happens. And you can just go through and set your random sample up that way.

Rob: Thank you.

Mark Ezzo: And, it's actually SAS native code. It's PROC SURVEYSELECT.

Rob: The next question, can we use our own local instantiation of SAS Enterprise Guide to access SAS Grid resources to which we've been granted access? And they continue, i.e., create a connection profile to the Grid resource without going onto VINCI?

Mark Ezzo: That depends, if you are operations the answer, it is probably yes. If you are research, the answer will be no because the reason VINCI was created was because we wanted to secure data. Because of all the personal data that is held in the VA, and some past data breaches, VINCI was created to reside behind firewalls that you cannot export data out, or import data, and except through prescribed resources.

So if I am on my own desktop, I cannot get into VHA CDW, DWA [PH] SAS MD2 because that is behind the VINCI firewall. In fact, I can't even get to GitHub or the Internet behind the VINCI firewall. So that short answer is no.

“: Great, thank you. This next one asks when is SAS Viya available, or Veeya”?

Mark Ezzo: “Vigh yah”, Viya is the correct pronunciation. That, or like I said, we are, we went through a POC. We actually just got the depo, we're going to install it, and put some project folks up there. And hopefully, it's going to come available in the next fiscal year. And as you know, fiscal years in the VA are September to September. So hopefully, we can get it up there in the next fiscal year.

We'll start off with a small footprint, probably allow 25 users, 25 to 55 users on a, at a time. But you can do everything you need to do, and in mostly on SAS Grid. This is for the really hardcore data scientists, the one who use Python and R. And you don't even have to use SAS \_\_\_\_\_ [01:04:20]. You can use pure open source. The the advantage of it is, is the enterprise structure. So we're looking at next fiscal year.

Rob: Thank you. Do you expect a monstrous impact with the impending EHR transition?

Mark Ezzo: I wouldn't say monstrous, I would say, yes, impact. But it's going to be manageable. We already have it out there under CDW work 3, is is the millennial, or, not millennial, the EHR, the EHR data. So that's, that's all out there now.

People are currently using it. And as more and more comes on board, we will, we get it through CDW. And that's how the first, the majority of people who are going to see it, it will be through CDW.

Rob: Okay, thank you. This one came in close to the end, probably about ten minutes ago, asking can you show the last script code again? I'm not sure if I can give you any more detail.

Mark Ezzo: The script of what? That's, that throws a wide, a wide loop saying script.

Rob: Okay, well, why don't we just move on to the last question. Can you talk about how to find and use the centrally defined SAS macros and functions?

Mark Ezzo: Yes. I, what you really want to do, we actually have documentation to it. If you look down here, and let's back up a little bit. If you come down here, the central site, there are a couple of paperworks down there about what's, what's available, and how to use it. Now, it's just as simple to use that as we, as we've shown here before when we were doing temp tables.

And as you can see, this is an example. Just say & SQL optimal, and that will bring everything up. If I, if I run that, it shows all of the parameters. Now, if you come into the, the site, the documentation site you'll you'll see everything, and anything. There is a huge paper on on how to use the the stored macros.

Rob: Those were the last question that we have. Attendees, before I ask Mark to make final comments, I will ask you to, when I close the webinar momentarily, please stick around for a moment, and answer the short number of questions that comes up when I do so. So Mark, any closing comments?

Mark Ezzo: I would say that you're looking at what I think really is the best analytical platform on the VA. And we're only going to improve it with SAS Viya. We're probably going to extend this out into the 22nd century. And one of the things that, that I've heard complaints about is all of the silos. We have a silo of R, a very small silo Python, which is difficult to use in VINCI.

A large silo of SAS, and why we can’t have them talk to each other under the Grid, under Proc IML, I think that where you're going to see a far improved environment, or a truly modern environment with SAS, the SAS Grid, and combined with SAS Viya. And I think you're going to see people who have never used SAS really consume it mightily. And we would become accustomed to it, and enjoy it.

And conversely, people who haven't used open source or who are very light in open source, start to use that even more. And I think you're really going to get the best of all worlds with a combination of SAS Grid and SAS Viya. And having looked at all of the other – and I am not deprecating any of the other ones, but I I think as far as pure research, and data science is concerned, you're looking at the crown jewel.

And we'll all be around to help service you and help you out in that regard. We're, our our SAS administrators group is very skilled in training, and understanding the data out there, and statistical procedures, and modeling. So we look – we are, anxiously anticipate this, and helping you folks moving forward, and in your very important, and necessary research. And that, let me close. That's my closing comment.

Rob: Thank you, sir. Have a good day, everybody.

Mark Ezzo: Take care, bye-bye.

[END OF TAPE]