JEREMY KEPNER: All right. We're back. So I think we just went over a sort of a tour of some of the more complicated analytics that people can do. And now, we're going to show some examples of not those specific analytics, but using the Reuters data set that we had before some of the analytics that we can do here.

So again, this is in the D4M API. We go into the Examples directory. It's Apps. And now we're going to deal with tracks. And this is a data set here.

The data actually, this Entity.mat was actually created in the previous. And this entity analysis actually creates the Entity.mat. It's the same file. So we just have over here.

And so let's get started. And just to remind people, we do load entity.mat. You see we have this E here. That's the data. It represents, essentially, almost 10,000 documents and 3,600 entities in that. If we do spy E. transpose.

AUDIENCE: Can you move the bottom of your screen up? Because we're seeing the top half of your thing.

JEREMY KEPNER: That better?

AUDIENCE: Much better.

JEREMY KEPNER: OK. And so there you go. That is the entire data set. Again, spy plot, very useful tool for doing that. So you can see we have locations and people and times in this data set.

We also have organizations in here. You can zoom in on it if you want to. You can see, the are the locations, and then the organizations, and the people.

Zoom in a little bit more, you can actually look at the actual values here. And you see there's this common popular location here. What is that? Ah, it's the United States. Yes, the United States does appear a lot in Reuters documents as one would expect.

What do we think this one is here? Any guesses?

AUDIENCE: New York.

JEREMY KEPNER: What?

AUDIENCE: Is that New York?

JEREMY KEPNER: Yeah. Oh, New York. See, it's already there. New York. Yes, we can read. All right.

Organizations, anything really popular here? Maybe this guy, is he popular? International Red Cross, right? And people here, we don't have any really popular people in this list. Ahmad Shah Massoud, I have no idea who that person was back in 1996.

Anyway, so that's the data. And you see, by the way, when every single time you click. Because when we make those spy plots, I have to do some compression on the strings just to make it work. But we actually always print out onto the screen the full exact string.

So if I want that string, you can just then copy it if you want to say copy and paste it or something like that. Or the full person or something like that, you can then copy that and paste that. You can do something like that like, E, this guy, yeah.

And there, it's showed. That's all the documents that contain this person's name. And then this shows the character position, I believe, that they appeared in the document. So again, you have all kinds of fun with that.

You can do row, that. That gets us all those rows. And then we can pass those rows back in. And let's say we want to do starts with, let's see here, how about organization?

Everyone correct my spelling here. I'm going to type this wrong, organization slash. All right, there we go.

AUDIENCE: Starts with.

JEREMY KEPNER: Yeah. I always get that wrong. Stars with-- starts with. See? There you go.

All right. There you go. So this shows you all the organizations that are cited in the documents that contained this person's name. You know?

This is kind of the spirit of it, right? I mean, it's like just, oh, I want that? I can get that. Oh, I could then say, oh, all right, well get me-- you know? And you could just keep going and going and going.

If I did r c s of that, right, it would never return those as triple. So r, those are all the documents. C, those are all the columns. V, there we go. You get the idea.

So again, very, very powerful type of syntax there. All right. So I'll just give you a sense of the data set that we're looking at. So let's look at the first example here. So we're going to do track, analytic, build, test.

So we're going to build some tracks out of this data. So I have these documents. And they have locations in them, and people, and times. So I could say, hey, if there's a person and a location and a time in a document, maybe I could call that's sort of a track.

So let's do that. So what do we do here? So the first thing we do is we loaded the data. The string values are those character positions. We don't really care about those. So we're going to just get rid of them and just convert it to a numeric like we've always been doing here.

And then I'm going to say my thing that I want to track is going to be anything starting with person. So I set that. And my time thing is going to be anything starting my time. And my location thing is going to be anything here starting with location.

So I've done the starts with to get these ranges. And now, the first thing I'm going to do is I want to limit my data to only rows that have at least one of all three of those. So I'm not dealing with I have a person and a location and no time or whatever. So I'm just going to clean that up.

So basically, I get all the people. And I sum that. I basically sum across the columns. So I basically compress the columns. And then I sum the rows. All right.

There we go, sum those. I get, then, all three. And then I filter them back out. And that just reduces this to the ones that contain just the ones that have these.

Let's see here. So now, I want to collapse these. I want to create, essentially, just edges and times. So I can do that with the call to type syntax and the val to call syntax. And going and bopping back and forth between that, I get a set of edges, the edge list, which is essentially the document and the time.

And now, I'm going to combine these back together into a new associative array, which essentially still has the same text label, which is essentially the document. But it has columns

of time. And the value is space.

All right. And then I'm going to do another one, which is, again, has a row, which is the document or the edge. And it has a column of space and a value of time. And now, I can construct a track from this through this wonderful sparse matrix multiply.

So essentially, I transpose Etx. And then I'm going to just get the people and convert those numeric values. And then we do this cat value mul, which will actually convert that.

These are time tracks. And these are space tracks. And again, it's a little difficult to explain to you exactly why this works, why these matrix multiplies give us the answer that we're going for. Because we're going to have to sit and think about the actual matrices.

This is a great example to go do, and then explore these various associative arrays to actually see why these matrix multiplies actually give us the answer that we want. So in fact, we can take a look at those. I just want to look at Figure 1 here.

So this shows us, basically, and I plotted the transpose of this, the people on the right. And then these are times. And so, basically, for each row here, I have a listing of times. And if I click on one of them, it will give me a location. OK.

In fact, I think that number's the number of times that appears. So basically, we have here the person, Daniel Smith was in a document with a time stamp of 1996, November 12. Oh, that's almost-- yesterday.

And then the location New York appeared once in that. Here's another one. And so this is a track of sorts. We basically have a person and a set of times and a set of tracks. That's one kind of track.

Another kind of track here is now we have person and locations. So that was the other matrix multiply that we did. And so now, we have person Carole King, location Buffalo, and on this time. So those are two different ways of representing the tracks.

Obviously, these are triples. But then you can use either of these matrices to do additional queries and other types of things. All right. So that just shows you how using matrix multiplies and other types of things you can construct more sophisticated graphs or data structures, in this case tracks, which is a very interesting type of thing.

Let's move on to the next one. That's going to be TA2. So this is a slightly more sophisticated tract builder. Again, so when I read the data in, i create my three sort of categories here, the object, and the time, and the location, or the coordinate.

And then I have a function here called find tracks, which actually just goes and creates those tracks that I essentially did in the last section. To be honest with you, the reason I did that is because some of those matrix multiplies used to be really, really, really slow. And so I did a sort of special function that took advantage of certain properties of the data to make it find these tracks much faster.

Eventually, I broke down and just optimized the matrix multiply. In the past when I ran that last query, before it would have taken like a minute to actually run the analytic, which got annoying. So I optimized it.

But we still have this code. This code shows all kinds of little tricks and techniques for doing things that are slightly better and using triples instead of associate arrays if you want to do optimization. So we leave it here. But the matrix multiply performance is now pretty good that these tricks are less necessary.

So what we want to do here, we have this track now. And I want to do a track query. So I have a person here, Michael Chang, another person Javier Sanchez. Now, Michael Chang was a tennis player at this time. Was Javier Sanchez also a tennis player at this time?

I don't know. I think there was a Javier Sanchez that was a tennis player at that time. So we just want to look. We're going to just do, essentially, here A and just say give me of this track. And say give me the listings for these two people, P1, P2.

And then we use our Display Full command to sort of make them in a nice neat tabular format. And you see here, basically, here is Javier Sanchez' listing. OK. And here is Michael Chang's. And you see there's no overlap here. We don't ever have them in the same time or same place.

We can also do things like track windows. So we can say I want to set a time range here and a location, Australia. So if we have our track thing here and I say, all right, give me the time range, T, and then equal to all locations in Australia, this shows me all tracks that essentially went through this location in this time window.

And these are the different folks that they list, Sanchez, Melissa Russo, whoever that was,

Michael Chang, and Michelle Martin. So those are just an example of a more sophisticated analytic. And here, we're using the fact that for our associative arrays, we actually have defined equals equals.

So this only works, though, if x is a constant. So it will check the value to see if that value, if it's a string, if it's equal to that string, or if it's a numeric value, if it's equal to that numeric value. But it only works with a constant.

One could argue that maybe I should make this work for a list of strings. But then the MATLAB syntax doesn't really work there either. If I have a matrix equals equal to a list of-- I don't know. I don't know if that really works.

So we try and preserve the MATLAB syntax where we can. And again, then we're just getting the columns. Again, this thing returns an associative array. This equal equals returns the associative array of all things that are equal to that. And then we can look at the columns.

All right. Moving on here, next one's TA3. So those are fairly simple track builders. Let's begin to do something that's, I think, kind of-- doing that track analytic, one could imagine doing that with existing techniques that are out there, existing tools and stuff like that.

It would be long. You would write a lot of code to do it. But you could do it. Now, let's do some things where you go, like, wow, this is really something that would just be prohibitively complicated to do it using other techniques.

So once again, we load our data. We convert it to numeric. We get our object and our time and our space keys. We find our tracks. And then we've built something called FindTrackGraph.

All right. And this is actually not that complicated. But it is more than, like, one or two lines. But what it does, it says, OK, I have this track. This track is a sequence of locations in a particular time order.

Well, now, I want to build a graph that's location by location. So if a track started in one location, and then its next destination was another, that will, of course, create a new graph. OK. So I now have a new graph, which is essentially 220 by 220 locations.

And we can actually take a look at that. And that's this graph here. So this basically says, you know, there was a track that started in Belgium, and then its next stop was Albania.

Or here's another one. It started in Australia and ended in Colombia. And obviously, we have a dense diagonal here, because by definition-- well, actually a lot of times, that's just the way it works. And so again, here's Damascus, Florida, all this type of thing.

So now, we've created a new graph of these tracks. Now, we can do something like a track pattern. So let's say I just want to look at the tracks associated with people associated with the organization International Monetary Fund.

So I'm going have starts with person. And I'm going to limit my data. So I'm going to basically limit it to data that begins with the organization. So now, I'm building a new graph, G0. OK.

FindTrackGraph of just that data set. So I've basically taken my A, which is this graph, and I said, oh, just go back and find me the people associated with the International Monetary Fund. And now, I can do things like, all right-- because this track graph the value is the number of times that occurred.

So for instance, if I say show me now all edges that occurred more than twice and where are the tracks that were due to people associate the International Monetary Fund were greater than 20% of all the tracks that occurred. So basically, I'm looking for something that happened more than twice, in that IMF folks did more than twice. And of all the data, the IMF people did it a lot of the time. So we're in a lot of people. It wasn't just a really, really popular track.

And so we see here, now we get Karachi and Afghanistan. So we had, essentially, at least two of those. And all of them were people associated with the IMF. Afghanistan and Britain. Here's Britain and England.

Well, obviously that's a little bit-- Britain and England are the same thing, right? Here's Islamabad, Islamabad, Moscow, Moscow. So here just shows you the kinds of things that you can do with typing in again.

This is a very sophisticated type of analytic. If you were to try and to do those things using existing types of things-- if you knew this was the analytic to do and someone handed this to you and said, go implement it using another technique, you definitely could do it. But discovering the analytic using existing techniques would be very, very time consuming.

And this kind of tool is very, very easy for you to explore and get the analytic just right. And I would say that in a certain sense what D4M is doing here is doing the same rule that we've

always used MATLAB for in signal processing. People use it to play with their algorithms, to figure out their algorithms, to get their algorithms right. They know this will give us the right answer.

And then when they deploy it and actually make it part of a real system, sometimes they'll just take the MATLAB code and make it a part of the real system. But more often than not, the target system or the target application will require you to port it to some other language, maybe C++, maybe Java, for deployment reasons. We still see that happening today that, you know, algorithm development is one thing, deployment is another thing.

Even if people use the same language for doing algorithm and deployment, usually deployment people end up having to completely rewrite what the algorithm analysts wrote anyway. Because the algorithm analysts had certain things they were concerned about. And the deployment person will have completely different issues that they have to worry about. But I think D4M allows you to still do that same kind of model on these new types of data in a very useful and productive way to get the productivity that we want out of that.

All right. Let's see here. So finally, one last thing, a more complicated analytic, which I call sort of a multiple hypothesis tracker. Essentially, what we're doing here is we're loading the data. We're going to just focus on one person here.

And then the locations are specified by time. I mean, the time is specified by the time column. And location is specified by location. And then I'm going to have this function called find multiply hypothesis trackers for Michael Chang with respect to the data set E.

And what this does is this says, all right, in the previous thing, I was basically just making one pick. If I had a document and Michael Chang was in it and there was multiple locations and times, I would just sort of pick one of those locations and times. Here, I'm now going to show you, for Michael Chang, for each document, all the locations and times.

So here's, basically, Africa and time. And let's see here. Maybe we can make this a little smaller. That would probably help a bit, a little smaller.

There we go. You can barely see that. But you see here, this basically shows all the times, all the locations. And this shows you, essentially, in theory the true track could be going through any one of these. There isn't a single track really for Michael Chang. There's multiple potential tracks.

And then this complex value I happened to store here just because I'm using complex values, the first one is the character distance. The location is-- so Michael Chang appears in a particular word position in the document. And it tells me that Austria appears 278 characters before him and that the time stamp appears 11 characters before his name.

And so over here, you see the time stamp appears-- this is a different document, but with the same time-- in different locations. So you can then use this data to actually go back and say, you know what, I only to pick the words that are closest to Michael Chang. And I want that to actually be the real track for Michael Chang.

So that just shows the more complicated things that we can do with that. So with that, that leads us to the end of the examples. And if there's any questions, I'll be happy to take them now if there's any. All right, good. I'm just showing you some of the kinds of things that people can do. And so there we go.