MINE CETINKAYA-RUNDEL: Hi, my name is Mine Cetinkaya-Rundel
and I am a Developer Educator at Posit,
as well as a Professor of the Practice at Duke University.
And in this video, I'd like to introduce you to Quarto.
So if you've been hearing a lot about Quarto
and wanting to dive in and have been waiting
to see when is the right time for me,
or if you've played with it a little bit
and want to get a sense of what more I can do
and what more you can do with it,
hopefully, this will be a good introduction for you.
So we are going to start things pretty simply. Authoring a document with Quarto
We're going to start with a single Quarto document.
I am going to open that up in RStudio.
So let's go ahead and open that up.
My Quarto document is called index.qmd.
For those of you who are familiar with R Markdown,
you're probably familiar with file suffixes called rmd.
And Quarto documents are called qmd.
At the top of this document, things
should look pretty familiar.
We have a YAML with some metadata about our document--
a title, the format that we want to output to.
So we're going to make an HTML document first.
And I'm going to use the RStudio visual editor. Using the RStudio visual editor
So again, if you are familiar with R Markdown
and haven't played with RStudio visual editor lately,
you can switch over to the source editor,
and things should look pretty familiar to you.
But let's go back to the visual editor where you get a document
that feels a bit more like Google Docs or writing
in an app-like notion where things are a little bit more
what-you-see-is-what-you-get.
For example, I have some information about the data set
I'm going to use here-- the Penguins data set from
the palmerpenguins package--
and I might want to bold the name of the package.
And I can do that by using the toolbar here
or I can use keyboard shortcuts to do that as well.
The visual editor has other handy tools on the toolbar.
For example, I can directly add a link to this.
So let's go ahead and add a link to the palmerpenguins website.
And hit OK.
So that now is a live link.
And if we were to switch over back to the source editor,
if you are familiar with writing things in Markdown,
you can see that under the hood, it is
writing the Markdown for you.
It's just that you are not the one

having to write the Markdown.
So let's go ahead and render this document
and let's see what it looks like.
A setting that I like to use on my RStudio IDE
is I generally like to preview my documents in the viewer pane
so that I can see my source code on one side and my render
document on the other side.
So we should be able to see our render document here
in the viewer pane on the right-hand side of our RStudio
IDE.
Things will look pretty similar between the source document
and the render document because our source document is also
using this visual format.
But the difference is that on the source side,
we only have the code, and on the output side,
we're seeing the code and the output to go along with it.
So let's introduce you to a few of Quarto's features now.
We can see that we are taking a look at the palmerpenguins data
set, and we have our visualizations here for it.
But we might want to take a look at the actual data itself.
So I am going to go ahead and insert a code chunk here,
which I can do either with the Insert, Code Chunk menu here
or using my keyboard shortcut with the forward slash
and bringing up the Insert Anything toolbar.
And I'll say I want an R Code Chunk.
And let's say that I want to glimpse at the penguins data
set.
And let's go ahead and Render that document
and we'll be able to see the input and the output
on the viewer side. Code chunks and chunk options
Scrolling down on this document, I
have a code chunk that generates a plot.
So one thing that I might want to do
is add an alternative text to this figure, for example.
And I can do that by using the fig-alt chunk option.
The visual editor nicely does YAML completion for me
so I can choose from here which chunk option I want to use.
I'm going to go with fig-alt.
And I have a multi-line chunk option
so I'm going to use this vertical bar here.
I've already written the alt text
so I will just copy and paste that here so you don't
have to watch me type this out.
But basically, we're saying that this
is a scatterplot of penguins' bill depth and length colored
by species.
And we say something about the trend we see.
It's relatively strong, and a linear association.
Let's go ahead and Render this document.

And when we Render the document, the alternative text
won't show on the output, because again, it's
not part of the output necessarily
but the underlying source code.
But if I pop this out to a browser window
and scroll down to my figure and right-click on it
to inspect it
I should be able to see that the alt text is actually
available for me under the hood.
If I'm writing a report, I might want to not necessarily show
all of this code on the output.
So one thing I can do is I can hide the code one
chunk at a time with another chunk option, echo: false,
and that will only hide that particular chunk
option, that particular code chunk's code.
But if I want to do that at the document level,
I have a few options as well.
I can actually go to the YAML on top of my document.
And under execute, I can say I want to do echo: false.
And let's go ahead and Render this full document.
And we will see that all of the code is now hidden
and we only have the output, which
might be preferable for you depending on your audience,
whether you want them to see the code or not. Inserting cross references to figures and tables
And here we go.
Here is our output without any of the code.
Over here, we have a figure that's
being generated with code.
And in the text, I have said the figure below is a bar--
is actually a scatterplot.
Let's go ahead and fix that
That scatterplot of species of penguins.
We can add cross-references to this.
So, in order to add a cross-reference,
I need to do two things.
First, I need to change my chunk label
to include the word fig at the beginning as a suffix.
So that's fig dash.
And second, I need to add a caption.
So to add a caption, I will use the fig-cap.
And maybe let's go ahead and re-use this information
from here.
All right, so now I have a figure that has--
a label that starts with fig and a caption.
All right, so let's highlight this,
and let's say we want to Insert a Cross Reference.
The figure is available for me to choose here.
And if I Render the document one more time,
it will create a link to the figure itself.

So now I don't need to worry about when
I render my document, if it's below or above, particularly
if I'm outputting to something like PDF.
I can do the same thing with tables as well.
The only difference is that I would
need to add the tbl for table suffix.
And similarly, I need to add a caption for that too.
So this will be, let's say, the first 10 penguins.
And let's go ahead and do this without using the Insert menu.
I can simply grab the label and I
can say-- instead of the table below,
at and the name of my table.
And I can Render my document.
And it will do the counting and the labeling of my figures
and tables for me.
So I have a figure 1 here and then
a table that's going to be cross-referenced underneath it
as well.
And voila, here we go.
We also have a Table 1.
The last thing I'd like to show in the visual editor. Adding a citation from a DOI
is adding citations, which can be a huge hassle sometimes.
So here we are going to cite where these data come from.
So in order to do that, I'm going
to say I want to Insert a Citation.
And you can see that I have some options to choose from in terms
of where my citations can come, but I
am going to go ahead and grab the DOI of the paper
where the palmerpenguins data were originally published.
And I can say that I want to Insert this.
And when I do so, two things happen.
Reference gets added here.
And you can see that my YAML already also gained
a new line called bibliography.
And if I go to my files, I have actually
gained that file called references.bib.
And in this file is the bib reference for this paper.
So if we go ahead and Render this document,
we will see that the reference has been added.
If I hover over it, I can see it.
Or if I click on it, it will take me
to the end of my document where I have a section called
References inserted for me.
So we've talked about creating a single document with Quarto. Seamlessly switching between output formats
And this has been a HTML document so far.
If I want to change the output to something like a PDF,
all I need to do is indicate that on my YAML.
So let's go ahead and give that a Render too.

I don't need to change anything in terms
of the syntax in my document.
Right.
Now we have a PDF document with the same content,
also hiding the code.
So respecting the YAML options that we had indicated earlier.
And the nice thing about this PDF document
is not only was I able to obtain it so easily,
but it also shows up in my Viewer tab as opposed
to a pop-up window.
Now let's go ahead and change things up even more. Creating Quarto presentations
We've gone from a single HTML document
to a single PDF document.
What about making slides?
So I'm going to make HTML-based slides using
our revealjs as my output.
And once again, I am not going to touch anything
in my document.
The only thing RStudio asks me to do
is to reload the visual editor when
I change the output format.
And let's go ahead and Render that as well.
And here we go.
Our document title is now a title slide.
I can actually scroll through the different slides
that I have.
Each of them are determined by these headings.
So let's go ahead and actually turn this
into a slightly more reasonable slide deck now.
Things worked but my text is a little bit squished.
I am going to take a look at the Outline view
here, where I can see the different sections
in my document.
And I'm going to make these second-level headings.
So these particular Data, Species, and Penguins
are going to be second-level headings.
And let's go ahead and insert some first-level headings
as well using the Insert Anything tool.
We'll say that I want a first-level heading.
Maybe first, I want something like Introduction.
And then at the end, I might want something like Analysis.
And maybe a slide here, something
like Modeling results.
And then I might finally want another slide
at the end called References.
Remember those references that I had,
I might want to move them to the very end.
So I've created this hierarchical structure
for my document that I can nicely

view here in the outline.
And if I Render my document, I should
be able to see that that same outline is apparent in my slide
deck as well.
My first-level headings are going
to turn into section dividers in my slides.
And then my second-level headings
are going to turn into indicators for a new slide.
So we can see that same outline here.
And I can go ahead and scroll through my slides.
Some things that I'm seeing here, for this data slide,
things look a little too big.
Maybe things are not filling in.
So using the visual editor, I can start tweaking things.
I can write full-on CSS for my slides,
but I can also use some of the predefined classes.
So one of my favorite is the smaller class, for example.
So let's go ahead and do that and Render that.
And that should squish my text, make it a little bit smaller,
allowing me to fit a bit more text into my slides.
Never a great idea to put a lot of text on your
slides, but if you're in a pinch and you just
want things to be a little bit smaller so everything fits,
this is a really nice fix.
Let's also go ahead and get rid of this option.
Or actually, let's go ahead and say that I
do want the code to be shown.
So I'm going to say echo: true.
So let's say I'm teaching, and I actually
want to show the code as well.
We'll see that our slides get busy
again because we have both the code and the output
on one slide.
This one looks pretty fine, but over here, we're. Customizing the output location of code in presentations
seeing that the figure is squished in order to make room
for the code.
Well, we have a remedy for that as well.
What I'm going to say is for this particular slide
and this code chunk, I want the output location
to be, let's say, column-fragment
and see what that gets us.
So what this is going to do is put my code
in one column of my slides and my output in the other column
of my slides.
But it will actually add some sort of animation to my slides
and delay the output.
So I might get something like this
where when I move to the next slide, I can see the code.

This still looks pretty squished so
if I don't want that, I can choose a different option once
again using the YAML completion with my tab.
And I might say, OK, just move it to the next slide for me.
So the plot moves to the next slide.
So we've made a single document so far.
And we have made also a slide deck.
Now let's imagine a situation where
you have multiple documents, a project,
and you want to put them all together in something
like a website.
So what we're going to do is--
let's go back and change the format to HTML again.
Reload our visual editor and Render this document. Creating a website from scratch
as a single HTML document.
And let's go ahead and add one more document,
maybe something that says "about,"
something like an About page.
For our web page, I'm going to create a new document called
About, probably has some boilerplate, our language here.
Maybe let's go ahead and select all of that
and delete it and say, this is a demo website.
And I'm going to save this file as about.qmd.
And now I have two documents, two qmd documents
in my project.
An about.qmd and an index.qmd.
And I have their outputs as well.
Let's go ahead and clear those out,
those outputs so that I have just my source files here.
To bring these two together in a website format,
I just need one file, plain text file,
specifically called underscore quarto dot YAML, dot Y-M-L.
And in here, I need to actually tell Quarto
how to bring things together.
So I want a Quarto project.
I want the type of this project to be website.
Note that I'm not even typing these.
I'm using the YAML completion.
And in this website, I want the title
of my website to be perhaps, welcome to Quarto.
And I want a navbar.
Let's say that on the left, I would
like to link to my index.qmd file and my about.qmd file.
And let's go ahead and save this.
And I am going to just relaunch this project from the project
navigator in RStudio.
And once I relaunch this project,
we can see that I gained a new tab called Build.
And let's go ahead in this new Build

tab, click on Render Website.
And voila, I have a website.
Let's pop this out to a browser window
so we can see things a little bit more nicely.
I can see that on top, I have a navigation bar with my two
pages, my Hello, Penguins!
page with my analysis and also my About page.
And they're automatically linked to that.
And basically, clicking on the name of the website
takes me back to my home page. Creating multi-format documents
Let me show you one other feature that's
actually a newer feature of Quarto, starting with 1.3.
Let's go ahead, and for this index page, add another format.
So I am going to say that I do want an HTML with default
options, but I also want a PDF of the same document as well.
And let's go ahead and render the website one more time.
And if I take a look at my output now,
we can see that not only do I have the HTML
file on my website, but I have an ability to actually download
a PDF version of my document.
So if you are creating websites but you have artifacts in there
that you would like people to be able to download as PDFs,
I think this is a really handy tool for doing so.
So we've gone from a single document to a slide deck,
to a website, but all of this is living on my computer.
To wrap things up, let's talk about how. Publishing the website to QuartoPub
do I get this and make it available to others.
Obviously, I could use version control to get things to GitHub
and publish from there, but another option
is to use Quarto Pub.
So I am going to now go to my Terminal.
And in my terminal, I am going to type publish or quarto
publish.
And when I hit quarto publish, it
gives me some options for where I might want to publish.
I'll say that I want to publish to Quarto Pub.
I have previously linked my account here.
So if this is your first time doing it,
it's going to ask you to actually create and link
an account.
But yes, I do want to.
And I do want to authorize this.
The site name comes from the name
that I gave it in my Quarto file,
so that was welcome to Quarto, and has simply created
a version of it without spaces.
And I'm going to hit Enter.
And we can see that it's rendering
my page one more time.

And it is now shooting this--
once it's rendered, publishing it on Quarto Pub.
All right, and here we go.
We can actually see that my page has
been published on Quarto Pub.
This is the view for me to manage the site.
I can delete the site, for example,
or change its Slug if I want.
But I can also--
I'm also provided a direct link to it
that I can share with others.
So I have the simple website that I
have created with the two pages now published on the web
and ready to share with others.
I hope you found this introduction useful.
And what I will do at the end of this
is commit and push all of my changes to a GitHub repository
where you can simply pick things up from where I left off
and continue your exploration of Quarto.
[MUSIC PLAYING]
Hi.
I'm Mine Cetinkaya-Rundel.
I'm a professor at Duke University and a developer
educator at Posit.
And in this video I'd like to talk
to you a little bit about how I use Quarto as an academic.
We're going to focus on two parts
of my life as an academic-- how I
use Quarto as part of my teaching,
and how I use Quarto as part of my research.
So let's start with the teaching.
We're going to turn to RStudio and take
a look at this document, a simple HTML
document with some code chunks where we create
a figure and then a table.
So let's go ahead and render this document.
To take a quick look at the output,
we're doing a very simple data analysis
with the lovely palmerpenguins package here.
And we'll be able to see that there are already
some things built into the document
I'm using: some hyperlinks and whatnot in my document.
I have some cross references in my document as well.
And what we're going to do next is enhance
this document for teaching use. Linking to documentation from code with code-link
Now, let's suppose that you are teaching
coding to your students.
One of the things that's often helpful to do
is to encourage them to read the documentation for functions

whenever possible.
And generally we tell our students,
why don't you Google the function?
I think we can do one better.
So one of the nice functionalities of Quarto
is that we can actually add automatic hyperlinks
to each of our functions.
So what I'm doing is in the YAML I
am adding a new field called "code link," and let's go ahead
and render that.
And here we go.
We can see that each of our R functions are now hyperlinks.
So if I was to click on a function like library,
I get directed to the HTML documentation
for that function.
And let's go ahead and click on something
that is from the tidyverse.
For example, what does geom_point do?
So for one of these packages that has a pkgdown site,
the code link takes me directly to the documentation
for that function.
And particularly for ggplot functions
these are extremely helpful, because then your students
can scroll down to the examples and not
only see the examples as part of the documentation,
but the rendered versions of those plots as well.
Next let's go and do something else. Informative YAML errors and YAML completion
We are going to add a code chunk option.
And I am going to say I want to hide this code chunk,
and I want to hide any of its warnings and messages as well.
So in order to do that, I will say, include false,
and let's go ahead and render this.
We can see that it doesn't render because I
have made a syntax error.
But the nice thing in the context of teaching,
particularly your student's learning Quarto,
and particularly if they're coming at it from having
learned R Markdown previously, is
that when you make a syntax error like this,
the message that you get from Quarto is really useful,
and it tells you exactly where the issue is.
And so let me go ahead and fix that.
I should be using a semicolon instead.
And once I do that, if I click on the tab,
I can actually use YAML completion to make sure
that I am typing "false" in exactly the way
Quarto expects me to.
And we can see here now that that code
chunk where we're loading packages is not in our output

anymore.
So as a professor, one of the things I do regularly. Creating Quarto slides with revealjs
is go from writing course notes or a book
at times to writing slides for in-class teaching.
And it's really useful when I can use the same source
document as a starting point for each of these different venues.
And with Quarto, it's really neat that you can
without having to change any of the text in your document,
simply by changing your format.
So in this case, we will make this reveal.js
to create HTML-based slides.
It's going to ask us to reload our visual editor.
And let's go ahead and render and see what that looks like.
All right. PDF export of HTML slides
And now we have a slide deck version
of our document with a handy sidebar tool.
And one of the things that I really enjoy here
is that I can not only make my slides full screen
or take a look at a speaker view,
but a question I get asked very often from my students
is, can we get a PDF version of the slides
so that they can annotate on it, say, on their tablets
or something.
And there is a neat PDF export mode that they can click on
and they can immediately get the PDF export
should they want to print it or save it as a PDF file.
Next, let's take a look at how might I
use Quarto in the context of teaching in the classroom.
Oftentimes I am teaching in a classroom. Annotating slides with chalkboard
where there's a projector screen behind me that happens to cover
the whiteboard, for example.
Or a student asks a question and what I really want to do
is draw on my slides.
Maybe something like, if I scroll to the figure here,
maybe something where I want to annotate my slides
and show the three clusters of penguins.
I can do this by adding fields to my YAML again.
So under reveal.js let's do "chalkboard true"
and render this again.
And you can see that in addition to the little hamburger
menu for the tools on the sidebar,
I have gained two new tools here.
One of them is an annotation tool.
I'm going to go ahead and click on that,
click a color that I might want to use,
and I can then use that to annotate my slides.
And the neat thing is you can actually
save an annotated version of your slides as well.
So if you tend to annotate on your slides a lot,

maybe do some derivations as you go through the class
and then would like to post an annotated version
of your slides, you could save those.
I'm going to go ahead and remove these annotations
by clicking on the Delete button on my keyboard.
And now let's get to that point where I actually
want to write something on the board,
maybe not just annotate my slides,
but there is no board to be seen in the classroom.
So I can click on the Chalkboard option
and basically start typing, start writing and annotating
on my chalkboard as well.
And I'm going to go ahead and get rid of that too now. Advancing slides for your audience with multiplex
Another handy tool is one where as you advance your slides,
the posted version of your slides advance
along with you as well.
This can be particularly helpful for students maybe
who may have lost where you were at in that lecture,
so that as you advance through your slides,
if they have it open on their computers,
they can follow along.
In order to do that, I'm going to add the multiplex option
to my slides.
And when we render this, we can see
that we have now gained two outputs to our slide deck.
One of them is just penguins.html,
which is what we had before as well,
and the other one is penguins-speaker.html.
So what you want to do is you want to be posting the .HTML
version on, say, your course website,
but you want to be presenting from your "-speaker" version,
and that way, as you advance your slides,
they will advance for your audience as well.
By default, when we look at our HTML slides, Revealing code in slides with echo
you'll see that the code is not visible here.
The reveal.js mode by default hides your code,
which is not helpful if you're actually
trying to teach the code.
So let's go ahead and turn all of that code
on using the execute option in our document YAML.
You can see that I can control here
a bunch of things about how the code is executed,
so I'm going to go ahead and set that to true,
and let's go ahead and render the document one more time.
I can see my code, and underneath it
I can see the table output.
If we go back to the previous slide. Highlighting code with code-line-numbers
where I have some lengthy ggplot code, and suppose that

I want to give some insight into this code.
Suppose that I want to talk a little bit about what
the scale_color_colorblind function does and then maybe
a little bit about what these theme functions do.
So I might want to highlight my code, so let's go ahead
and see how we can do that.
In the relevant code chunk I'm going
to add a code chunk option that says, "code line numbers,"
and in here I am going to say that a vertical bar to indicate
a pause.
I want to first talk about line number nine.
That's the scale_color_colorblind.
And then I want to talk about lines 10 to 11.
And let's render this again.
If I now advance my slides, I'll see
that first, line number nine is highlighted,
so I can talk a little bit about that.
And then I advance again, and I can
see that now lines 10 and 11 are highlighted,
and I can render them again.
The neat thing about this is that we
are able to specify this as one of our chunk options
as opposed to having to edit our code
and add syntax to the code chunk itself.
Now, the problem here, though, is. Customizing output location with output-location
that my plot is really squished because reveal.js
is giving the space for the code chunk
itself and then using whatever space is available
that's left over in this slide for that plot, which is not
very helpful.
One approach to solving this would
be to create another code chunk that references this one using
its label, but a neat Quarto solution to this
is to use the code chunk option Output Location.
And I can choose to--
the default means that the output
is going to show up underneath your code chunk.
Fragment means it's going to appear when you advance
the slides, and the Slide option says
to put the output on the next slide.
So let's select that so that we can get a nice big plot
on the next slide.
All right.
So I have just my code on this slide.
I can advance through and see the highlighted versions
of portions of that code.
And now my plot itself has moved to the next slide. Showing code chunk fences with echo: fenced
Now suppose that you are actually
using Quarto to teach Quarto.

You actually want to show your students
how you can add all of these code chunk options.
In the rendered document they are
impossible to see since they're hidden from the output.
But we can easily reveal those by adding a new code chunk
option called "echo fenced."
And let's go ahead and render that too.
And now our code chunk actually prints not just the code itself
but all of the code chunk options as well.
So I didn't have to create a new sort of indented code chunk
or something to be able to reveal these.
And you can see that the only code chunk option that's not
visible here is the echo fence.
So it's smart enough to hide that one
and show the rest of the code chunk options
so you can easily talk about how to write Quarto documents
with your learners as well.
Now we've added a lot of options to these slides that
are very much slide specific.
Let's go back and turn things back to an HTML document.
So I'm going to change my format to HTML,
which asks me to reload the visual editor one more time.
And let's go ahead and render that again.
All right.
And let's talk about one last code chunk option. Code annotation
that can be really helpful in a teaching context,
particularly if you are writing long-form documents, course
notes, or a book where you might want
to annotate your code further.
I'm going to scroll down to the very last code chunk
here where we're creating a table,
and I'm going to add some annotation to it.
So I might say this is my line number one.
So the hash sign and then one.
And we're going to basically annotate
each one of these line numbers.
And let's go ahead and say this is four.
And right underneath my code chunk,
I'm going to start writing up my annotation.
So something like, "start with the penguins data frame,
then slice the top 10 rows, select some variables,
and make a table with gt.
This code annotation feature that's
available with Quarto 1.3 onwards
allows you to annotate lines, each line of code,
and basically add some text to it so that you can--
if you were teaching in person, these are probably
the words you would say about the code,
but now it's really easy to capture them

in a written document as well.
And it doesn't have to be one line per code.
So I could do something like give the same annotation
number to that last line of code and say
that I want to capture both of those comments
in a single item in the list underneath it.
And here we go.
That item four has gone away, and now I
have an item three that basically covers everything
that's underneath it.
So we've showcased a bunch of Quarto features. Authoring manuscripts with Quarto templates
that are helpful for teaching, but what
about the research or the publication
side of an academic slide?
So let's talk about that a little bit next.
The neat thing about writing papers with Quarto
is that it comes with some article templates.
So I'm on the Quarto documentation page here,
and I am going to scroll down here a little bit
and go to the GitHub repository, where
we can find a bunch of journal templates here.
I tend to write papers that use the American Statistical
Association journals template, so we're
going to take a look at that here.
If the template that you tend to use for your papers
is already listed here, that's fantastic.
You can get started right away.
And if not, I encourage you to open
an issue in the repository.
Or if you're interested, contribute the template
to the repository so that others can use as well.
So for the JASA template, it says to install it.
We're going to use quarto use template, and then
quarto journals jasa.
So I'm going to copy that line of code,
and let's go to that page here as well, go to my RStudio
and open up my terminal.
And let's go ahead and paste that there.
It asked me if I trust the authors of this template.
Yes.
The directory name can be My Awesome Paper,
and let's go ahead and see what we got.
In my files pane here I can see a new directory called
My Awesome Paper, and there is a Quarto file in there
with a very extensive YAML.
And if you have written papers before that, whether either
with R Markdown, with the articles package,
or just using LaTeX, know that the preamble
tends to be pretty long.

So I am going to give it a new title, "My Awesome Paper
About Data Science," and maybe I can just
add my name quickly here.
And let's go ahead and render the document.
And here we go.
The output looks like something I'm
actually pretty used to seeing when I write papers
using other tools as well.
And the neat thing with this template
is that it comes with some pre filled-in text that tells me
exactly how I might include figures or tables in my output
and how I can even include some citations as well.
So let's switch over to the visual editor, Inserting citations from Zotero or from a DOI with the RStudio Visual Editor
and let's talk a little bit about citation.
So I would like to add a citation to my paper.
Let's say that I want to add something right
at the end of my introduction.
And I can do this in one of two ways.
I can either use the Insert menu and say
that I want to add a citation, or I can use my Insert Anything
tool and say citation.
And you can see that I have even linked up my Zotero library
here.
And so I can click on that.
I happen to have one of my papers saved in my Zotero
library.
So if you use a tool like Zotero or Crossref
or something like that, you can actually link these
up to your RStudio, and as long as you
have them running on your computer, on the same computer,
they automatically link up anyway,
and I can go ahead and insert that.
So let's go ahead and save that and render this document
one more time.
Right.
So at the end of my introduction,
the new citation is there.
I can click on it, and I can see that it has been
added to my list of references.
And what RStudio also did for me at the same time
is it added that entry to my BIB file as well.
So I can see that has already been inserted into my BIB file.
What if you don't use a tool like Zotero?
Or maybe you are looking for a paper on Google Scholar.
So how about we look for the Welcome to the Tidyverse paper.
And I'm going to go ahead and open that up,
and we're going to grab the DOI of this paper.
I have copied that from here.

Let's go back to RStudio, and let's say
that we want to add that citation,
suppose here, using the Insert Anything tool again
and citation.
I'm going to say this time that I want
to find a paper by its DOI.
So I paste the DOI here, and it goes out and searches for that,
and it allows me to insert that citation either
as an in-text citation or not.
Let's go ahead and hit Insert.
And once I have done that, that has also
been added to my BIB file.
We can see that that's been added to my BIB file,
and a reference has been created for it as well.
And if I render my document one last time,
I'll be able to see that that citation has been inserted
to the appropriate location as well,
and I can find the paper in the list of references
at the end of my document.
So we've talked a bit about using Quarto for teaching
and a little bit about using Quarto
as part of your research.
If you've been intrigued or if you've
already been a Quarto user and have enjoyed these features
that I highlighted here, there is an abundant number
of features available in Quarto that we haven't gotten a chance
to touch on in this video.
So if you would like to learn more,
I encourage you to go to Quarto.org.
If you are just getting started with Quarto, Wrap up and learning resources
the Get Started menu will be helpful,
where you can learn about how to use
Quarto using your editor of choice, whichever
one is your favorite.
And if you have already dived into Quarto
and know the basics, I suggest going to the guide to dive
further into any of the different outputs
that you might be interested in creating,
like a website or a book or adding interactivity
to your documents.
And to stay up to date with what's
happening in Quarto land, the blog is a great place to go.
Particularly when there are new features,
they are generally documented here.
Thank you very much.
[MUSIC PLAYIN
notice it's a little delayed hey everybody Welcome to today's R
Studio Enterprise Community Meetup I'm joined here by Tom and we're just
chatting for a second to see if the LinkedIn stream is working as well

how are you doing today Tom I'm doing great happy Tuesday it's always a good day of the week
awesome let me go over to LinkedIn and see if we cool and see
ourselves I'll just mention that for today's slides the link will be at the bottom of
every slide so you can always copy it from there if you come in later um as well as we're going to be screen
sharing and using Multiplex for the first time today which will allow us to like actually change the slide on other
people's computers which is kind of neat so we'll try that out awesome well look at that it is working
on LinkedIn so hi friends thank you so much for joining today welcome to the
rstudio Enterprise Community Meetup I'm Rachel and I've met so many of you
already so thank you for joining back today I'm calling in from Boston today and we are streaming out to LinkedIn and
YouTube right now so if you've just joined now feel free to say hi through the chat maybe where you're calling in
from so excited to have you here for our third quarto Meetup in the past few
months if you missed the past two and welcome to quarto workshop and building a Blog
with Cordo I'll be sure I'll be sure to share those in the chat as well
but today we are joined by my friend here at rstudio Tom mock who is going to
teach us about building beautiful reports and presentations with Cordo
if this is your first time joining one of these sessions this is a friendly and open Meetup environment for teams to
share use cases teach lessons learned and just meet each other and ask questions
together we're all dedicated to making this an inclusive and open environment for everybody no matter your experience
industry or background so I will also share a link in the chat where you can find out about other
upcoming events too as we have these meetups every Tuesday at 12 eastern time
we also have a data science hangout every Thursday at 12 eastern time and
would love to see you at those too I just want to say during the event today you are able to ask questions
on the platform that you're watching from and it will pull together here for us or also anonymously
through slido and we'll share that link in just a second here and pull it up on
the screen but thank you again so much for joining us here today I'm so excited to have you
with us here too Tom and I will turn it over to you
awesome thanks so much for hosting Rachel and for all your help in getting this stuff off the ground
um we're gonna be talking about beautiful reports and presentations with Porto uh it feels a little bit meta in
the sense that like you're presenting about quarto with quarto on Cordo kind of like all the things wrapping up
together and something we're trying out today and I think one of the benefits of using a format like Cordo or reveal.js through
Cordo is that uh like for now Rachel's actually sharing her screen so I'm able
to actually advance and modify slides on her computer just by changing them on
mine so you can imagine a situation kind of like today where maybe my computer's acting a little funky and I'm actually
allowing my colleague to screen share or to work you know remotely on their own
computer or like for y'all if you access the slides from the URL at the bottom of the screen
I can actually advance and kind of display the slides as you're progressing through them whether in a teaching
setting or in kind of this webinar style setting so we're going to get kicked off and I

will run with it so first off I just want to mention that you know we talk about this pretty
frequently but uh Cordo and our studio have very deep integration together so
you have What's called the visual editor so this is just a screenshot of a few different things from the visual editor
mode in our studio you can use this to do word processing and kind of modification of quarto
documents including presentations and there's lots of helpers there for inserting divs and spans some things
we'll be talking about as well as things like tables Graphics links code chunks
lots of help in terms of making you more productive one of the benefits that Cordo provides
is that the visual editor mode is available for really all the different formats so again whether you're writing
a presentation or a report or something else those can all be edited in in visual editor mode
the first thing we're going to talk about for today and we've got a lot of kind of content to cover so I'm trying to provide a lot of overview and kind of
exciting broad overview of things that are available in quarto so first off we have static documents or essentially
reports that you're writing I'd like to think of these as like your your daily driver you know you can use it for basic
kind of exploratory work but it's powerful enough to also present uh and
kind of display results to your colleagues uh even publish to a journal or as the final output for your team to
share with uh leadership or other stakeholders I think Cordo and static documents are
most powerful when they're combining this idea of like written thoughts and observations like the human and the loop
in terms of I'm adding context and adding pros and text about what I'm working on but I'm also including the
source code there so I can be reproducible in creating my journal article or creating my documentation or
creating my report I'm putting together so in that Spirit of kind of combining source code with uh with the report
they're actually generating we're going to kick off by talking about parameters and parameterized quarter reporting
you may have used these in our markdown or maybe you never use parameters in our markdown before but parameters work
inside Cordo as a way to create different variations of the same document by changing out small
components most commonly you see examples of this like I want to create a report for this
specific region so like for today's tidy Tuesday data it's you know 50 different states and all the different things that
are available within those and you can imagine I want to generate a report for Texas as well as
Massachusetts or New York or California like all these different states that I want to create separate reports for
but it could also be covering a specific time period so if you're doing like a Time series analysis maybe you want to
report over the you know the past year but maybe someone asks you say okay well I just need the data for the past month
rather than restarting that report from scratch you could actually use parameters to regenerate that with just
changing out the date or kind of the last example just re-running a single analysis multiple

times and changing components so maybe you're doing machine learning with Heidi models and you're switching out the
different engines that are available uh for that model so you can still do like a random Forest but change it out from
Ranger to the random forest model to a spark model for example and we'll walk through a few of these different examples there's two ways of actually defining
parameters uh in Python if you're using that with quarto we use what's called paper mill style which is a way of again
providing parameters and generating many reports from one set of source code
in this situation uh the tags are actually generated um within the code chunk so you use the
Hash Pipe syntax then you define tags and then you have like a vector of parameters that you're creating
so in this case we have Alpha and ratio that are being defined as both being 0.1
these can be accessed later on in the document by name so they're not um they're really kind of like first
class objects in the situation of like you can't name something else alpha or ratio those are going to be the actual
um in kind of memory object that you're using in Python in r or using the Knitter engine uh
we'll Define them in the yaml so you'll use what are called the params key and then you can define specific parameters
there again like Alpha and ratio and giving specific values whether that's numeric or text or other strings
these are accessed in source code using the params dollar sign syntax so you'll
actually extract something like Alpha from the params environment in this situation you could actually have Alpha being in the um in memory as an object and then params dollar sign
Alpha that you can access separately but generally you don't really want to name them exactly the same to avoid kind of
namespace collisions the last option here is you can actually
Define and modify parameters from your command line interface to Cordo or via
the terminal so if you call quarter render on a specific document you can modify the parameters with the dash p
and then Define the new parameter you want to change so if I wanted to change Alpha to 0.2 and ratio to 0.3 I could
have multiple calls with uh Dash p and then Alpha equals 0.2 and Dash p ratio
equals 0.3 and again so regardless of kind of how you're building out your document what
language you're using there's there's these ways to approach it and modify the document so that when you're generating
it you can get multiple reports from the same set of source code
now that we've kind of talked a little bit about what parameters kind of how they're defined let's actually show a
little bit of how those could be used so I have this example of a pretty simple report in this case we're using
the Palmer Penguins data set and we're generating out a report that looks something like this on the right
just says you know when it was published and it's got data about this set of penguins and and which ones are
classified as this specific species at the top left this slide on line five
of the yaml it says params equals species equals to adeli so that's the

default in terms of if we render this document as is it's actually going to always generate a report for how to
light Penguins you know it's not changing itself unless we say we're going to change it but we can access
that parameter with params dollar sign species in a couple different locations so we can use it in our filter step on
line eight of the first code chunk and we can also use it in inline code with
the backtick r param dollar sign species and again report on it both in inline
code as well as in our overall code chunk but the big takeaway being we could
actually change out that parameter to a different species and actually get a whole new report from the same uh same
quarto document so if we were to do this via the CLI we
might do something like quarter render that report or quarter render that report as say like an IPython notebook
if we're doing it from Jupiter uh in that situation and then we could define specific parameters with the dash p and
change species to Gen 2 and get this Gen 2 dollars report out as opposed to a
different name so we can actually say the output is going to be a new report name because we don't want to overwrite
the existing adelai report that we've generated so this is fine in terms of like I can
render it with the CLI I can modify that parameter on the Fly um but maybe I don't want to use the
command line interface to do this rendering I could also do it with something like pure R code so it might have a render
function I write in this case I'm using the r package Cordo and quarto underscore render this allows me to take
in an existing report penguinreport.qmd modify the parameters
According to which species we're applying and then name the report according to again which which species
we're using so here we can find all the unique species across this whole data set walk
across them or regenerate the report with the per our package and render that
on each of the three penguin species this will create something like the Adelaide report the chin strap report
and the Gen 2 report all of which have slightly different parameters that have been modified and now we have three
different versions of the report by species as opposed to having to rewrite the entire report from scratch or copy
paste code across so using parameterized reports like this even if you're doing it for you know
really sophisticated uh nice looking reports that you're building is one of the powers and kind of superpowers of
using code and using things like quarto to generate out those documents
another example that might be more applicable to folks coming from like the machine learning aspect is maybe you
have a report and uh you know we'll just say that we're adapting it from some tidy model stocks in this case we're
using our parameter of a specific engine so the default is using the ranger engine and we're using that for a random
forest model but we could also change out that engine to any other engines that um the Tidy
models ecosystem can accept so we can take that same tidy models example report use the random Forest

engine or the ranger engine or Spark engine and again all you know reusing
most of the source code but just changing out that engine and running it in a slightly different context
if we do this we can actually generate out two sets of kind of predictions or
two sets of metrics from this report so you know these really don't look any
different but just that we have run the report twice and if you look at some of the very small value changes we can
actually see that the values did change so like on line 9 on both sides we have
a prediction of 5.25 versus 5.32 and the accuracy or the rmse and some other
metrics are actually slightly better when we use the ranger engine for this
so regardless in terms of like whether you're trying to regenerate a new report that you're actually using to tell a
story or if you're doing exploratory analysis and doing like different models or different engines that you're trying
to run you can use parameters to modify and decrease the amount of code that you
have to write for doing those steps
so while I think parameter is really exciting and I wanted to kick off with that as like there's a benefit to using
Code one of the other benefits that Porto provides is you know regardless of if you're using lots of code or changing
it out or reusing it you might actually want to generate multiple different types of outputs so maybe you're
creating a PDF document a Word document and an HTML report you know those are
three different ecosystems that all take quite different um kind of code behind the scenes
Cordo as much as possible tries to make that code easy to use in terms of using
the same syntax across all of them and unifying the work that you're doing across different output formats
so let's say we have this report I have a dog named Howard who's a Boston Terrier so I took an example from
Wikipedia there's this article about Boston Interiors I copied it over and converted it to a a Cordo document
so with this unified syntax I could do something like quarter render that document to HTML or to PDF
now I haven't optimized this document only for HTML meaning I haven't written
a bunch of CSS and I haven't optimized this document only for PDF meaning I haven't written a lot of latex or really
any latex I've only used quarto style Syntax for this or kind of unified
syntax that works across formats when I generate this report out I get
something that looks like this where I have a nice looking article I've got an image in the gutter I've got hoverable
citations I've got a floating table of contents you get all the power of what HTML can provide for a nice you know
report or document you're creating but without changing that document and literally just saying render to PDF I
also get a really nice looking PDF document that's pagenated and has you know whatever the limitations are PDF
like doing that in terms of its still reports on the history it still has citations and
images in the sidebar or the margin but I haven't necessarily had to optimize
only for HTML or only for PDF I was actually able to use cordo's unified syntax across both very different uh

kind of outputs that I'm creating so we're going to dive in a little bit to what the syntax is how you can use it
to kind of be safe as to the different type of outputs you're creating and walk across some of that
and what I mean by this unified syntax is that Cordo makes heavy use of what are called fence divs and bracketed
spans so here on this page we have fence divs we have three colons that start the
fence and three colons that end the fence so you can imagine that when you're writing these you always need to
have kind of a front and a start or the enclosing of some content within those
once we've used three colons to indicate the start and the end we can then apply
a specific class or specific name of the div that we're applying so it might be
something like dot class or dot panel tab set or or whatever it is just some
name within those curly brackets that are after the fence div and then anything within the start and
end of that fence div will be modified so everything inside that will be affected in some way or you can alter it
in a useful way moving forward this structure again starts and ends
with an equal number of colons there's always a minimum of three but they can be however many colons you want as long
as it's three or more and you match the start to the end you're going to add those curly brackets after uh the three
colons and this indicates the start and the end as well as the class so again like dot class or sometimes it doesn't
require that dot and it might be something like word one dash word two or
sometimes dot dash class overall just that when you see this syntax inside maybe quarter
documentation or in other examples uh making you aware of what those mean
one example of how to do this is we can actually do figure layout of different images and include like alt text with
them and other things so we'll have three colons curly bracket layout number
of columns equals to two and this will create a two column layout and as long as you have two images it'll
put one on the left one on the right and then it'll start over and if you added more images it would stack them always
adding up to two columns you can modify this and have number of rows or number of columns be larger but
overall just showing that this type of syntax again can apply to word or to PDF
or to HTML it's basically allowing you to be safe across different types of output formats
another example of how to use this is something like dot panel Dash tab set
now this one is very specific to HTML but I just want to show off kind of some
of the things you can do with this some of it can be used only for protecting yourself against different output
formats and other times it can just be useful in terms of providing cool things
that HTML can do so a DOT panel Dash tab set allows you
to create little tabs that you can step through and kind of organize your code in that way
after you create the fence div again with three colons to start three colons to end and then your class of dot panel
Dash tab set you'll define specific tabs by level two headers so you know two
pound signs element one that'll be your first tab two pound signs element two that'll be your second Tab and then you
can have content within each of them what this looks like in practice is you

might actually have code and output so if I were to click on code first it'll
just show off the code that is unevaluated but if Rachel were to click on output it would actually show the
output there on her screen so now she can actually click through and see you know have some type of
interactivity or have some type of interaction with the document in addition to me actually having to step
through all of it again this is limited to HTML output because there's that true interactivity
but just showing that um these these fence divs are not just
for kind of doing some of the basics but allow you to do more across all the different HTML formats as well
if I wanted to create or show how that was actually uh shown on the page
um that dot panel tab set and then I have a level two header code I have some
you know R code that I'm putting in there I have another level two header called output and that one is actually
going to eval equals true basically show the output but Echo equals false and not
show um the source code itself this is just a basic example but you can
imagine that you could have again these multiple tab sets that you're using with fence divs allowing the user to explore
your document and interact more with it as they're reading through walking through the document
the other item that allows you to again either protect yourself against different formats you're creating for or
just do cool things are what are called bracketed spans we have a sentence on this page that is
missing some words this is a magic text basically text that shows up after the
fact when I click next on my side of the screen we can do this again with what is called
a bracketed span so we do square brackets around specific text in this case magic text for the previous example
we're showing and then just like with our fence divs we apply a specific class
to that text the main difference between these two is that offensive will apply to like an
entire paragraph or collection of text and a bracketed span could be applied to
a single word or single words within an overall sentence so you can imagine you
know applying to the whole paragraph versus small components of that again to you know help tell your story or
indicate something on the page here I'm using a DOT fragment class
which is specific to presentations but this lets you actually delay the presentation and delay the specific word
to show up later as you're moving through that slide deck so overall you know bracketed spans can
do a lot but just showing one example here quickly inside the presentation format
the structure would be something like this you're going to surround specific texts that you want to affect or the
specific object you want to affect with square brackets around it and again you have a left and a right square bracket
indicating a start and an end of what's being modified once that is written out you're going to
add curly brackets to indicate the start and end of the specific class so
something like Curly bracket dot class or dot fragment and then closing it off
with another curly bracket so that it can be fully interpreted and again this just allows you to modify
specific components of the text or style your document um in in specific ways
one example it's a slight modification of this and one that we use in that report about Boston Terriers is we
actually create inline footnotes so something like some inline text with a

kind of tilde and then the square bracket footnote or this uh up up Arrow
kind of symbol this allows us to add a footnote or kind of this like endnote to a specific word
so in the document it'll show this as like a hyperlink back to the footnote or
the endnote but also allow you to inject a lot of text about that overall so it could be kind of like a citation or just
an aside or a comment that you're making that you don't want to display all the time
in HTML these can be hoverable in terms of in an HTML document I can actually
hover over those footnotes and see them displayed kind of as an overlay in the document in a PDF they don't necessarily
have the ability to render that additional JavaScript or css so they're just linked to the footnote or the endnote at the end of the page or at the end of the document and what this looks like in practice is
I might have a sentence like for example did you know that Howard is a dog and then adding a specific footnote
specifically he is a Boston Terrier although sometimes he acts like a cat so by providing that footnote you know
that specifically all the way through cat that whole section is hidden on the page but added as this footnote
providing an additional context about the sentence as we're walking through it so if you're using this for kind of
scientific documentation or just kind of a sides you're making a useful thing to add to your documents
the unified syntax can also do more complex things in terms of we move back to fence divs we can overall structure
our document so here from that actual article I showed I have a mixing of
images aligning those with text and paragraphs and putting it together something like this
uh so for here I've applied a layout class and I have the image as 30 of the
width of the page and then layout equals to 70 for paragraph one and then below
that uh a hundred indicating that paragraph two will take up the remaining uh width of the page below the other two
and again regardless of the output format PDF or word or HTML I can structure and kind of typeset my document using this type of syntax I could also move one of those images
into the margin so you know I had the small boss interior that was on the right side of the screen I have a
traditional kind of hyperlink to an image on this page and then
um I have a specific class on that called a DOT column Dash margin and that
actually takes the image and throws it into the margin of the overall document while the rest of the body continues on
and then as we talked about just previously add things like footnotes so information about where did I get this
detail the average life of a Boston Terrier is around 11 to 13 years this was a retrieved from the Boston Terrier
Club of America and providing additional context about that citation or around that detail that we're adding
so this unified syntax again while I think that writing for HTML is some of the most powerful way you can actually
do this in certain situations you have to write to PDF or you want to write a
PDF or to word docs so you can collaborate on them in a different way you can use this unified syntax across
different output formats with a quarter report you're putting together

again while I think that um Cordo reports you know in terms of making them look beautiful and and typesetting them
is very useful ultimately a different value that quarter provides is that it can also add code in both the display of
that code The Hiding of that code or you know just the output of that code itself as a graphic as a table or as raw text
so in your document you might have something like this where you have an HTML document you're creating and you
want to have you know the code not be shown maybe you're generating a report that's going up to leadership they don't
necessarily want to read our code or python code they just want to see the final report they don't necessarily care
how you got there maybe they just want to see the details so in this case you can still keep the
source code so you can keep your dot qmd that has all the source code the links out to the data basically how you got
there but when you generate out your report it doesn't show any of the code it just shows the output and the
commentary that you've added to it you can still turn on code individually
so let's say you turned on Echo equals false at the top and the yaml at the code chunk level you can actually add
and say like for this one code chunk I want it to be Echo equals true so regardless of what the yaml says turn
Echo and actually show the code for this specific document you can also rather than hide all the
code and make it where no one can see it is you can do what's called code folding and in this case you'll have code
folding that you can hide temporarily in terms of it'll fold all the code behind a little menu bar that you can click on
and that little menu bar will be called hidden code so you can click on each individual code chunk and display it and
then collapse it as you move forward and again this could be turned on at the
specific code chunk level where I could have code folding turned on for one code chunk so even if I was hiding all the
rest of it for one that might be really important I can still say hey this one allows you to toggle the code showing on
or off again foreign
going a bit further you know maybe we want to show the code maybe you don't want to show the code we can also use
what are called code dash tools so in your format you might have format equals HTML or PDF code tools equals true gives
us this little overlay where the user can make all of the decisions so at the very top of the document they can have
this little drop down and they can you know hide all the code or show all the code or in some situations even view the
source document so you can embed all of the source code into the document and when you click on it that allows you to
show all the source code before the results were generated or link out to an
external Repository again this just allows you to make those decisions as you're using it for
reporting but you can choose how much of the code or how much of the contacts you want to display for your users

if we wanted to show The Source in terms of like literally embed the source code into the document you could use code
tools equals Source equals true toggle equals false and no caption and this
will allow you to see the source code but there all the code is also displayed in the document and you're not trying to
hide it you're just trying to also embed all that source code so that users can see it or you as you're automating that
report can also see how the the document was generated and if anything has changed over time
overall when you display or look at the source code it might look something like this where you have an overlay that pops
up displaying what was all the source code that you used literally the markdown as well as the code chunks and
all the options of it so again very useful for kind of checking that your source code is actually valid for what
your expectations are for other situations where you don't
necessarily want to embed all the source code into the document or in some situation where you really don't want to
show the code itself you want to have it in like a private GitHub repository or maybe it's just really long and you
don't want to embed all of that because the user experience is not ideal you could actually use code Tools Plus
source and indicate an external repository so here we're linking out to
quarto dash Dev and linking out to a specific markdown file that displays the
source code for that specific page so they can actually go and look at it on a first class you know version control
system as opposed to embedding it into the document itself but again you get to make that decision
in terms of displaying the code hiding the code embedding the source code all this different things you can do for
your users experience we can then move on to things like
making the overall document look a bit better so maybe you do want to show the code but you have like a dark theme or
you have a specific style that you want to apply to the code so by default your code might look
something like this where it's highlighting things like Library D plier in your group by and other function
calls are highlighted in blue your arguments are in green and it's doing
some highlighting of specific components but it's really white on white with this
background there's a lot of highlighting options that I've included in a link at the full
highlighting options but you can modify this throughout your document
so for our different themes let's say we're using an HTML report we can stick with highlight style equals Arrow which
will look something like this where you have blue for function calls you can have logical statements that are a
little bit lighter blue numerics are in red and it allows you to kind of read the source code a bit easier as opposed
to them all being one color all being black or gray or something
but again what if you made this into like a dark mode so here we've modified the yaml header
to have the same highlight style of Arrow which is the one we're trying to use for accessibility reasons for um
contrast and other things but then we have the ability to toggle on and off a

light mode in a dark mode so in Cordo if you use a theme equals light and theme equals dark and change
those out you can actually have a toggleable option to make a switch from light mode into dark mode
now this again will modify how the code itself looks on the page because your background is black
so you can use a highlight style like arrow that works for both light and dark
and it will actually modify the background color of the source code to also match the light or the Dark theme
so again by changing out some of the styling you can make your overall document look the same or have the same
kind of theming or consistency As you move across it
so while you know changing code itself is kind of cool overall you're probably a bit more interested in what you know
what else you can do with that code so in R there's a package called downlit
and this not only provides like syntax highlighting but also links out to
specific source code so if you're using download and actually evaluating code this will actually allow
you to click on the functions in the chunk uh chunk output and it'll actually
link out to the documentation for that so you imagine someone's like oh I saw this cool package called per and it's
got this function called walk that Tom was talking about in the presentation you can actually click on my source code
and click on per walk and it will take you to the documentation for per walk so
you can read a bit more about that function so download again will provide this linking of the r code to external
documentation so that when people find something they've never used before they can actually move forward into the
documentation and have a lot of power there to learn on the Fly
so while code is cool ultimately I think that you want to do something like the overall Aesthetics you have Howard here
and his you know finest Garb he wants to look good for the presentation you also want your report or your presentation to
look nice for your users or for your uh your kind of stakeholders
so for HTML documents out of the box Cordo is styled with something called bootstrap 5. this is a CSS framework for
websites and version five is one of the latest versions its overall applies a lot of different theming to the overall
document so Porto comes pre-installed with 25 different themes allowing you to set the
Aesthetics or the appearance of the overall document and you can use them in one or two lines of code
so in your yaml header you might Define a format equals HTML and the theme as
something like litera that is a nice minimal theme that really makes the
information pop as opposed to trying to make a really fancy background or something
but there's lots of different themes that you can customize and go further maybe you want to start from the litera
theme but further customize it with a DOT scss file that's modifying the
structure or the appearance of the document beyond the defaults that are applied with the litera theme so you're
not stuck with it you can continue to customize it with either CSS files or scss files in this case if you're using
the theme output boots watch themes are you know there's

many of them again I mentioned 25 I just have a small selection of six here on the page so again these are going to
change all the different elements typefaces menu bars nav bars the overall
kind of weight of different fonts and then colors of both the background the foreground and buttons so when you're
creating these HTML documents this again allows you to create nice looking documents and not having to change all
the specific components yourself but using these pre-built themes that look really nice out of the box
foreign when I gave some initial overview talk about
um kind of reporting or in our first Workshop we did there's a lot of people asking about PDFs you know obviously I
think HTML is one of the best formats and you can do the most with it but in certain situations uh like a journal
article they want to get a latex file or they want to get a PDF or in some organizations just they want PDFs
because they're emailing them around or doing things like that so quarto can accept what are called
templates that allow you to style or kind of take it as a default structure with DOT Tech files so if you have those
laying around or modify them you can further customize and optimize for PDF
if that's the route you want to go there are there's also support for kind
of first class Journal articles from Cordo so actually taking a document not
having to go and do write all this latex code that you're customizing it with but just using one of the quarto journal
formats and Publishing it out to specific kind of
specific publication that you want to go to like the ACM or the Journal of statistical science or something like
that and you can end up with a report that looks really really nice where this journal of statistical software uh where
again you're not having to write all of the custom latex to make the PDF look like this but rather making use of
Porto's built in kind of chordo journals formats that are available and just
writing things like what is the abstract what is the title what are the authors and their affiliations and it will take
that and build it into the document to actually provide a useful output or what
the journal statistical software is expecting when you actually deliver the article to them
again quarto wants to support this idea of reproducible science you know whether that's science in an organization and
you're creating kind of static PDFs or HTML documents or actually publishing in
Academia to to journals or two specific journals in some cases
I want to say there's like 10 or 12 journals that are supported as of today
but we're also taking kind of feedback from the community as to what other journals are needed in working with
specific groups to to get those built out
I also want to call out a nice project from the nfms open science team so they
put together a quarto extension called quarto title Pages the tech templates
were written by Eli Holmes in the Lua filter they kind of makes it available to Porto was written by Mikhail
and they have reports that look something like this so yes this is not necessarily like a journal article this

is more of like a professional PDF report that you might use in a business setting allow you to put your company
logo and stylize the report in a specific way while also adding subtitles and authors and other things to the
document something you could deliver in like a Consulting setting or in a business setting there are additional ones but mostly
this is focused on improving the title page which is a big deal in some of the PDFs and making those look really really
nice as the default so just wanted to call out some great work that they've already done and
linking out to their repository that talks more about that
so we've talked a lot and kind of gone through a lot of different things with reports in terms of HTML reports theming
them including code PDF reports and articles and how you can modify those I
want to spend the last kind of section of today talking about presentations again I'm using quarto to write the
presentation today about Porto so I really believe in the tool and its ability to help you tell stories
there's a lot of supported output types and I'll just call out three specific ones reveal JS which is an HTML format
that's what I'm using today it has the most power and you can use all of kind of HTML and CSS and even some JavaScript
to do a lot with your presentation but there's also support for things like
PowerPoint if you need to work in an office setting or Microsoft Office kind of Suite or if you want really prefer
working with like latex or PDF you can also use the Beamer format to work with that structure
so regardless of kind of which format you're going to do I'm going to focus on initially things that are cross format
and then talk about some things that are specific to reveal.js that are really powerful and exciting in my mind
I also want to call out again two other community resources that I'm linked to here on the slides so Emil wrote this
kind of slidecraft 101 and we'll talk about that at the end of the presentation about colors and fonts and
using them in quarto and then Megan put together this nice overview of customizing quarto slides and adding
little section headings that say like you know rather than just being 37 of 58 slides through which section are we on
and what's going on in that page foreign
for the things that are shared across the different formats so again latex or HTML or PowerPoint there's many
different things that are shared across that so you can use what are called incremental reveal or incremental lists
that allow you to display specific list elements one by one you can control the column layout the title The subtitle the
author the you know the kind of title page of your document you can control the headings as well as you're using
Code so you're able to display that code hide that code and actually generate
things like tables graphics and text as the output of your code that you're generating from your quarto presentation
the first thing to remember is you know for creating slides it could be as simple as this uh in the yaml header

Define a specific format so I'm using reveal.js but that could be PowerPoint X
or um or Beamer and then each of your additional slides will be indicated with
a level one or a level two header and that's what indicates a new slide so slide one is a level two header slide
one uh two more pound signs slide two and that'll be slide two
and this allows you to structure your document in a way that you can read the source code but the actual quarter
representation knows how to turn it from that source code into the final output that you're creating
so overall you know very short slide but just one that shows you how quick you
can get started with creating uh slide decks with just a few lines of markdown code
there's also this idea of incremental lists and an incremental list means that it increments across it so you might
have like three or four bullet points and rather than displaying them all at once you can actually have them be
incremental where it'll show bullet point one then you click an arrow bullet point two click the arrow bullet point
three and so on so that you can actually walk through your slides like I'm doing today as opposed to them all showing up
at one time so you can do this from the yaml header with incremental equals true
or you can turn it on and off via fence divs so might have something like this
where again I use three uh colons to start three colons to end and then I
have a DOT incremental class and this will work across any of the presentation formats and it'll say first Howard
sleeps and then the next one Howard Sprints he's always sleeping or running around in terms of he's he's an active
dog or he's a very sleepy dog but you can walk through the different parts of your presentation and control
the overall structure in this way you can also turn off incremental so
let's say that you had incremental equals true and you want to say that well you know I want to say that Howard
eats and then he sleeps like that's kind of one thought that I'm putting together even though it's it's two different
bullet points so if he eats he's immediately gonna go take a nap he's not gonna go run around after that so you
can turn on or turn off these incremental lists from the yaml or from pandok uh fence divs with the dot incremental or non-incremental classes you can also control the overall
structure of your presentation in terms of creating left and right columns or how those appear
this one's a bit more complex in terms you have what's called a DOT columns class first and you might notice I have
four colons here at the start and four colons at the bottom this allows me to Nest additional uh
fence divs within that kind of parent div so overall I have a DOT columns class
and then I'm creating two or more columns that I want to create within it So within the dot columns class I create
another uh three colons with a DOT column and I can control the width so
you know make it 50 50 or 40 60 or whatever kind of percentage I want from left and right
and then I'm also using reveal.js's dot fragment class that allows me to display
the content on the left first and then display the content on the right after that's done
so you can kind of build off of these uh classes and brackets and fence divs
together and overall structure your document across different formats

foreign so while those type of things are useful and that gets you a lot of the way in terms of if I stopped right
there you could actually get started making presentations like you could actually make a nice presentation from that I do want to call out some specific
reveal.js options that are very powerful and allow you to do a lot of different things
so I'm gonna have Rachel use one of our secret tips so on this page we can use alt plus click to zoom into a section
and if I pop up this tiny picture of Howard on the screen she can all click again to back out of that and she can
actually zoom in specifically on Howard so you can imagine that you might have source code or you know something else
on the page that looks small but you can zoom in on it to highlight it
Rachel can also do command plus on her screen that will actually increase the
size of things like the URL or the logo at the bottom so she can zoom in a little bit on the page and that will
actually make that a bit bigger and zoom out a few times and make the URL a bit smaller and overall just make that part
a little bit easier to work with that's part of the benefit of HTML is you can modify things temporarily on the
Fly you can also use things called absolute positioning so like that last image I
put it very specifically in the bottom left so that we had to go find it and zoom in on it I use some code like this another fence
div and I use the dot absolute class and then I can say what is the geometry so
from the bottom 25 pixels up and from the left 100 pixels over and then
display the image as normally with a fragment class which shows up later but you can place it anywhere on the screen
in that way and really tightly control placement of images or logos or even text that you're putting on the page
that allows you to better tell stories as you're moving through it
another one that I really like for reveal.js is using what's called output Dash location this is actually used
within code chunks so you might have something like this where you're loading the ggplot package doing a quick plot of
empty cars and then if I Advance the slide it throws that image directly next
to it in the second column and it's a fragment so it shows up later and this
again allows you to step through showing the code and showing the output as you're moving through the document
there are many other output Dash location classes but just this is the one that I find the most useful in terms
of delaying showing it and putting it in the column next to the source code so I can have long source code and a nice
image next to it just like with HTML or using PDF with
different themes or kind of partials you're bringing in quarter presentations and specifically
reveal.js also come with built-in themes so there's 10 themes listed here on the page and you can apply those with the
theme argument in the yaml header and again these will change all the different elements of the presentation
uh including kind of the tables the headers the background colors the
foreground colors lots of different elements you can change on the Fly very very quickly

if for some reason you didn't think the theme was good enough or there was something that you didn't agree with you
can always customize it further and apply your own custom.scss file to modify specific
components of the presentation so like for today I wanted to use a very
specific font you know I like the fera mono class of fonts for code and at our
studio we use open Sans as kind of our default font family so I can import those fonts from the
Google fonts API and then apply some different colors make the text the bottom orange make the level two headers
this nice dark blue kind of the star studio blue and also use open Sans and ferramano as my default font and my mono
space font so again I you know not a lot of code but just modifying a little bit to make
it look just like I wanted to so using a theme and making it go farther with some of these different customizations
I can also add a footer or specific logo so I could use a local image so maybe I
have the rstudio logo locally at the bottom right of my slide or I have some footer that says like footer text or a
URL or another hyperlink to the source code for my presentations I like to again use
the URL to the hosted our studio logo so I don't have to keep bringing it around I can just link out to the URL of that
one and then for the footer I can also use the uh you know a specific hyperlink
so with markdown style code I can say reports and presentations recordo and then link out to the slides that are
hosted on Porto Pub but for this page you might notice the footer is not that in terms of you can
actually change the footer as a one-off and if you look at the very bottom it's kind of small but did you see did you
notice the footer and using a fence div you can apply a footer class and modify
the footer Slide by Slide by slide so for one footer you might link out to this URL and for the next one it could
be an entirely different footer you can override the default that's defined in the yaml
but again that will only apply for that uh specific slide so if I Advance the next one it'll go back to my overall URL
at this point you're probably seeing dozens of pictures of Howard and his nice gentleman outfit I've used what's
called a slide background to Overlay many different images across that not the most effective use of it but
just showing you a couple different possibilities of how you can use images or other things as your slide background
to tell a story I did that with um you know a level two header slide
background and then used these curly brackets who actually define specific things about it so made the color equals
black the background image was one of Howard gentlemen and I made them 100 pixels wide and repeated if they were
not big enough to take up the whole screen overall there's a lot of different things you can do with this in addition
to the background color you can use a local or a URL to an image you can make
the image cover the background or not cover make it a specific size change the
opacity or the transparency you can do a lot of customization to really make it
easy to inject background images to your slides but you're not just limited to images
you can also use it with gifs if you're using those to tell part of your story or tell a joke

and you can even use it to just change the color so I might do something like slide background background color equals
black and this will make the overall uh body black but make the text white it
will try and always do a high contrast color for changing the font color itself so you don't have to do that manually
you can just change the background color and the foreground color will change along with it
and just like with images or gifs you can also do background videos these are rather than background image it would be
background Dash video and you can control the looping or does it play audio or is it muted and again things
like the size or the opacity so you could actually embed a full size video taking up the entire screen if that was
what you were trying to do in your slides or in your presentation
there are some other helpers so things like R Dash stack and combining that with fragments I can take three images
and display them on top of each other very efficiently um one by one by one
again you might have like several different images in unison that you're displaying so something like here's a
Boston Terrier in a field here's a juvenile boss interior laying down on the ground and here's an adult Boston
Terrier wearing fancy clothing and you can all kind of stack them on top of each other and apply them one by one by
combining the dot R stack and the fragment classes to display them overall
as we close out we've got a couple more slides to go I just want to again call out Emil's article and kind of how we
approached using color and custom styles to apply and change his documents
I hyperlinked out to his article so that orange article will actually take you to his overall blog post
but in his scss styles that he added he included a DOT blue and a DOT yellow
class and these just inject a little bit of color to help tell his story a little bit differently in his presentations
so in action he might write some text that looks like this he uses bracketed spans to say Cordo in blue enables you
to weave together content which will show up in yellow and executable code in yellow into a finished presentation
to learn more about Cordo makes it blue again see this specific link
so by doing that he's using the existing kind of context of bracketed spans but adding in his own custom classes to
style the document a little bit differently and he gets a nice looking page like this where you can really highlight that
quarto content executable code and finish presentation is the story here so
while he might read off the slide or I might read off this slide you can really see that we're focused on Porto content
executable code and that's a nice summary of what quarto provides content text and executable code all together
and you can further extend this with what you need to do or what you want to do with writing your own classes
again he used the bracketed span syntax so you know square brackets around some
text and then applying his specific class to that but you can also use our Studio's visual editor and he actually

talks about this in his article and you can use you know insert a specific span or specific div and then it'll pop up
saying what are the type of classes you want to apply to that so this is really useful for maybe you can't remember all
the different syntax you can actually insert divs and spans with the visual editor inside our studio
foreign that's it for today we we got it all done in an hour somehow we covered a lot
of different things with reports and presentations theming them including HTML PDF doing reports and presentations
together and separately parameterized our markdown and parameterized quarto and how those are different and the same
and I wanted to give you a few links out to some nice articles and content
so if you just want to read about quarto you can go to quarto.org docslash guide
that's the overall guide for Cordo but there's also sub links to documents or
articles and reports or Journal articles if you're trying to publish in an academic setting I also wanted to call out again a few of
the open source efforts by other people outside of kind of the core quarto team so quarto title Pages by the group from
nmfs or Emil and Megan's articles on slidecraft 101 and quarto slides and
what those look like so a lot of nice content here and you can access my slides at Thomas
mock.cordo.pub slash reports Dash presentations or uh Rachel was kind enough to give me
a short link which is much better rstd.io slash quarto Dash reports that
has all the content for today I thought that might be a little easier
there thank you so much Tom really really appreciate it this is awesome I'm
just getting started with Cordo too and this is really helpful thinking about some of the formatting I was trying to
figure out uh there are so many great questions
um how long do we have before I I'm here for the the full Journey so I
blocked off the remaining part of this before lunch so I'm in for the Long Haul awesome great so thank you all for for
all the great questions I did want to remind everybody that you can ask questions in LinkedIn if you're watching
there on YouTube live or you can ask anonymously um and I will show that slido here as
well I was also playing around with a few polls as well so just asking if people have used Cordo and what other
journals you would like to see so Tom if there's other questions you want us to ask everybody else feel free to put them
in the private chat um but one of the let's go we'll go back
and forth between uh YouTube and Linkedin here a question from Jimmy was any plans to
make quarto template functionality uh similar to what our markdown draft
does if I'm being honest I'm not very familiar with our markdown draft so give
me uh 10 seconds and I'll look up that function but just very honestly I'm not
very familiar with the r markdown draft option um so let me make sure I understand the
question and we'll go from there so let's see
ah okay so kind of like using it as a template is my understanding is our
markdown draft uh creates and obviously edits a draft of an R markdown based on a template
so there's a couple ways of doing that in terms of uh the rstudio dev team is
working on some more integration of like there are our markdown templates that exist in our studio and that's a kind of
a first class idea of you can like new file new or markdown use the specific

template a dark Studio Dev team is still working on that idea for Cordo and then for
um quarter documents themself you can use either the extensions kind of workflow or kind of other ways of
bringing in kind of a overall document importing code or specific chunks and then further modifying it that's a good
option and I'll go back to the dev team and see if we have some additional thoughts there but um yeah cool cool to
learn about the draft function and kind of learn something new on these presentations
thank you a question from slido was can you apply the same code highlighting to
different outputs so for example HTML or PDF without the call having to be nested
under format yeah so I would say the if you're doing
something like styling um you're typically you're you're taking that step forward saying the defaults
aren't good enough and I have very strong opinions um so I would say the safest option is
doing them separately because like PDF has certain things it can do and HTML is
certain other things that it can only do um so while there are times where you
can reuse the components like even when I was using [Music] um The Blood Theme which is kind of like a
Dracula style theme for presentations I rendered it to PowerPoint and some of
the theming came across basically what changes it could make but overall it wasn't one to one per se so if you
really want to go further and customize it like there's first class ways of doing that in all the different formats
so PowerPoint you can import a a template from PowerPoint PDF you can do
additional theming or Tech partials and for HTML you can do CSS or scss to
customize them so while yes you know for code highlighting alone there are some
options where you can try and reuse it there might be a time where you get frustrated and it's better to kind of
hop out to if you're really trying to do like PDF and make it look a certain way further customizing it in that way
awesome thank you um I was trying to go over to
let's see trying to go over to LinkedIn for one of the answers to this question
so I will ask a different one first oh wait here
it is Daniel asked any websites that in your opinion have the best uh shiny
Cordo output examples that you could share with others at your company
yeah I would say you know today what quarto provides with integrating with shiny is traditional websites in terms
of there's not a dashboard style chordo yet that the quarto team is actively
working on dashboard style documents or dashboard layouts with Cordo that that
will come in the future um but you know if you're really trying to invest in like shiny and trying to
make it look really beautiful with shiny alone uh there's things like shiny dashboard or fresh or BS lib specific to
shiny that allow you to structure and and kind of theme those overall documents in the future we'll have kind of a
better answer for you which is like if you're trying to create a dashboard with shiny and Cordo together
um there will be a first class dashboard format however if you're creating it today it's
it's basically going to look like a website that has shiny code built into it so I would you know go to the
cordo.org there's a gallery and that has a lot of the content today and will be updated over time
um and I'll drop that chat link to Rachel so she has the gallery link

um but you know today it talks about like interactive docs with shiny or with observable or Jupiter widgets
um but you know they're really going to look like websites as opposed to dashboards right now
awesome thank you yeah I put that link into the chat I also see someone had shared it on LinkedIn a bit earlier too
so thank you for sharing that um someone else asked
um are there built-in themes for PowerPoint yeah so you know again some of the theme
components work um across format so like I was using like the The Blood Theme that's built in
for reveal.js and some of that came across in the PowerPoint um there is an article on quarter.org
for um PowerPoint presentations specifically and again once you get into the level of
theming that basically says like not only do I care about outputting to PowerPoint but someone has strong
opinions about that theme um so if you use what are called PowerPoint templates that's a first
class idea in the PowerPoint World both in and outside of Cordo that is how you
can use like a company theme that modifies the title page and the layout and the colors and the fonts for
PowerPoint specifically and because that's a proprietary format there are some limitations around how much we can
do that in Cordo as opposed to how much of that is like this solved answer is use a template that that PowerPoint has
I really am I'm really impressed by Tom's ability to talk and find The Links at the same exact time and send them to
me so I just put that in the chat as well thanks Rachel another one was how do you keep long
quarto documents organized for yourself as you're working on a presentation
yeah there's a couple ways if I'm being honest about like writing a presentation I'm usually have many different things
open in terms of like I have some experimentation files that I'm like testing out specific ideas and then I'm
putting those into the individual slides but I will say that in terms of like when I write quarto documents in our
studio the level two headers the code chunks all of those show up in the document outline
um so if if you use that you can navigate pretty quickly even through like a 800 or 900 line presentation so
like today's one because it's almost 60 slides I want to say it was around like 700 lines of text overall
um and using the outline view in our studio allows me to kind of move between
slides really quickly especially if I'm like changing something at the title at the very first slide and then jumping to
slide 50 like using those allows me to navigate the document a bit better
the other thing I will just mention is that Cordo um regardless of the format does allow
you to use what are called includes so you can imagine that the final document
I'm putting together for leadership might actually be from four or five different people and they might actually
write a specific section and then you could just import or what is called include them in the parent document
so if you look for includes on quarto.org it will actually allow you to
access existing content and bring it into the parent document and actually
just inject like all of that qmb or that markdown file in that location
so especially if you have like a bunch of different people working on one thing yes you could collaborate across Source
control but they could also write their own kind of thing and you just bring it into the parent

great thank you um okay this is a little bit of a longer question
um so essentially like what are the benefits of converting from our markdown
to Cordo especially for someone who maybe has hundreds of of our markdown
reports yeah I would say overall the the beauty is that we support both in terms of if I
say we as in our studio like the core team that works on our markdown is the same team that works on Cordo so you
don't have to change everything which is a great thing like if someone told you tomorrow hey stop doing what you're
doing and move to this that would be very disruptive and that's not the goal of anything with Cordo there are
differences and there are trade-offs and benefits for using Cordo or using our markdown for certain situations but
especially if you've got like a thousand line our markdown report or you've got a workflow that's working great like keep
using it like we're continuing to support our markdown we're not like abandoning it we are still doing bug
fixes and kind of improvements over time just if you think about it with like new features there were limitations in terms
of how we extended our markdown the way to extend our markdown was to write another r package that extended it and
with quarto a lot of it is baked into the one in install so rather than having
to manage multiple packages or have it be only with r we're trying to not only
improve things for our our users but also bring essentially like all of our markdown to the Julia language into
Python and to teams that don't even use code like just bring a visual editor and
markdown style scientific communication without having to manage data science
languages so it's a bit hard of a question to answer in a short format but
I would say just like if you have existing our markdown you can keep using them for basic our markdown you can use
quarto to actually render or convert those but you definitely don't have to convert over
and would for somebody hoping to convert some of their docs is that welcome to Cordo Workshop a good
resource or what would you yes yeah that first uh overview as well as
some of the content on quarto.org it talks about uh taking an existing arm markdown and a basic one at that where
the conversion cost is essentially zero because Cordo and our markdown are
essentially backwards compatible with each other the difference being that like our markdown has specific CSS and PDF and
latex that it applies and Cordo does things a little bit different visually like the the code will still work
between them for the most part but the visual representation is different between the two
okay yeah the two-hour webinar had some some additional content that would go over that
awesome um I see a question from Rob on LinkedIn
that was would things like check boxes or radio buttons be available or would
one need shiny for that level of complexity there's a couple ways to approach it
um so yes if you want check boxes and radio buttons that execute our code then
you would need shiny right because you have to have a server that takes that checkbox you're doing and executes it
along with some R code however um both an r in Python and in JavaScript
within Cordo there are things for check boxes or cross language talking so in R

there's things called HTML widgets and crosstalk that provides basic filters
and select with check boxes and little buttons in Python there's something called
IPython Widgets or IPI widgets that again give you both a JavaScript kind of
framework and then some some toggles you can do with it or native to kind of the JavaScript land
but also applies to other languages observable.javascript is included with
Cordo so you can build out custom kind of interfaces and check boxes
um you know just through Court of alone that interact with both R and python code so I'll drop a link to Rachel for
observable JavaScript that's going to look very similar to shiny in terms of like the the buttons you're expecting
um and it can be embedded with r and with shiny and with python but it's actually JavaScript that you're writing
just in a very friendly way like it's a great library and I'm glad it was included with uh the integration
recorder great thank you heading back over to
slido some of the questions over there any plans to add more categories to the
author argument and the yaml because it's well known you can add
authors and affiliations but it would be cool to also add role yeah absolutely so the the default
things that are available from Cordo are kind of limited to what's available for like the categories in the author
especially once you get into specific Journal formats the authors section and
affiliations really expand because certain articles and certain journals want new things added on like there are
requirements so um yes some of the journal articles that already exist those Journal article
extensions allow for this and if you're interested you can also add your own
format that adds whatever you want like you could add a favorite pet as an
author argument to your yaml and just like write an extension that provides that
um if you wanted to learn more about that I would look at the kind of article templates
um that you can modify on quarto.org and those are kind of like dot Tech formats
or css that you're bringing in that allow you to further extend what's available in the ammo header and that's
really what those extensions are for is adding on custom things that you want to see that maybe aren't as generally used
awesome thank you and I see you had shared with me Tom a question for everybody and it was what kind of chordo
content would you like to see in the future and so I just put that as a poll over on the slido as well if you want to
share there awesome okay I see someone just asked a
question over on LinkedIn and it is mayank asked can we include custom HTML
CSS JavaScript code combination for parts of a quarter document as we did
with our markdown yeah absolutely so there's option when I
talked about includes from before that include could be used for including
child documents or other qmds but I also just dropped a link for Rachel called includes you know before body after body
in header resources metadata files those different options allow you to include
like literal JavaScript code or css or like Google Google analytics dot HTML
files that are injected into the document and those are able to be included in a couple of the most common
locations so before the body after the body in the header like there's specific
locations where HTML needs to inject that type of metadata for it to be used

by the website so Cordo has a first class support for doing that with the include Dash something else options
great thank you um Arun asked on YouTube does Cordo work
with a Sharingan yeah so um it there's a long answer and a short
answer and I'll try and do both quickly so the uh the long answer is that
Sharingan uses a library called remark.js which is not pandok compatible
markdown so while it was useful with our markdown it's funny like Yi Wei was
actually talking about it he never intended for it to be popular it was for his use
um and it required what he called like Ninja level knowledge to to do certain things with it
um reveal.js is pandok compatible markdown so that's the one that that
Cordo settled on as the HTML format is that because you could use it with the visual editor and it was compatible with
all the rest of pandok um that was the one that they they settled on if you take an existing Sharingan presentation uh and try and render with quarto it will work in terms of on
putting that in scare quotes um it will render it into some type of presentation but Sharingan used a lot of
custom themes a lot of custom CSS custom classes and different ways of writing
that is just not fully compatible with quarto that all being said and continuing the
long answer I was a Sharingan absolute power user loved using it and I've fully
converted over to doing reveal.js that I really really enjoy it I if you know if
I wanted to I could still write Sharingan but reveal.js adds in a lot of things that clicked with me and again
the unified syntax across all of HTML is nice where I don't have to remember how
do I do it in this one specific package but this is the way to do it across all HTML style outputs in quarto was nice
option thank you quick reminder if anybody's
looking for the content and slides from The Talk today I'll just share the link up here on the screen again
um I see a few slider questions that I have a lot of upvotes right now and one
is any plans of creating tutorials um for DOT I don't think I know what
this is but dot TX templates for PDF output for Cordo would be amazing
yeah so there's some ongoing work for uh writing things like Lua filters
um and other things like that um let me see if I can find the Lua development in creating formats
so this article I'm sharing with Rachel has some of the information about like when you're writing a custom format what
all do you need to include um ultimately in terms of like while we
might end up writing like dot Tech uh template examples or theming components
Tech is its own massive ecosystem and while you might be able to get introduced to it from
quarto.org for example it's such a broad topic that I'd be afraid that lots of
things would get left out so multiply it's like if your team you know what cordo's trying to provide is a lot of very common and Powerful templates so things like Journal articles where I could publish to this journal without knowing any Tech right like that that is my mindset I don't know any latex and I can still publish
in the format that this journal wants so jss wants this you know latex file that they're getting I can do that with Cordo
if I want to customize PDF to go a certain way I really need to learn a
little bit of tech or Tech partials uh to to further customize that and that's a skill that I would need to develop to
be successful in that role um so yes and no in terms of there's some documentation around creating

formats and how that can be used for specific things and distributing them developing filters and things with Lua
but there aren't major plans for for Tech templates today again I do want to call out that like uh
the open source Community has already started building out some of these so I'll call it again this group and
they're in the gallery on quarter.org they actually put together
um source code for for kind of how they've approached building out those so
grabbing that content in FMS Cordo pandok I'm actually looking for
this one so give me a second to find those I could go back to my presentation
and just do it that way there we go all right so I'm going to drop this into the chat for Rachel so
that one actually talks about how to modify their templates and how to modify
and use and DIY some of it that resource is amazing like it's a great amount of
work they did to both do it but then also write about how they did it so
while that won't teach you everything about dot Tech and partials and everything else it at least gives you
like a nice structured way of learning and again because it's an open source project over time additional people
including us and others will build out educational content or kind of extension
content around that so specifically the how to use DIY and
how to modify on that link that Rachel just shared and if you wanted to actually screen share that page
um that I can walk you through some of that on the side it's a it's a great site and kudos to that team
awesome let me get that up I was just trying to uh copy over to LinkedIn too
and just let everybody know if you're looking for these real-time links and you're watching on LinkedIn I wasn't
able to do both at the same time so if you want to head over to YouTube but uh here let me put this
up yeah so again this page has like seven or eight templates that they provide that are great like they just
work out of the box but off of the left side bar if you see the how to use DIY or how to modify those actually if you
click on like how to modify um it talks about you know the first
thing modifying pandok template sounds like fun like yes it's kind of uh tongue-in-cheek but you know you can
modify it how it works what is going on and it talks a little bit about like what the things you do for quarto as
well as like some of the things you use for their specific format but again because there's so much customization
you can do in the dot Tech file that's completely separate from Cordo um you do at some point have to learn
that if you really want to customize all of that yourself but this is a great starting page it
talks about the latex structure and all sorts of other things I would go here great great resource
awesome thank you thank you okay uh let's go through as
many questions as we can get to and I promise I will share all these other questions with you Tom if we don't get
sounds a good thing okay um
so if this is anonymous question if we know we want to use a number of parameters do we have to give one of
those parameters as a default in the yaml yeah so you'll want to Define them at
least because you could imagine that when you're trying to render it if you have a if you indicate it in the body of

the document it has to have a default value so if you don't Define it in the yaml
then when the document is just rendered with like the render button or quarter render
um it's going to fail because that object doesn't exist and it's like I don't know where to find this it doesn't
exist so you will need to Define at least some default in the yaml that's
used for the first rendering and then you can modify it further that really applies to both like the
body of the document because you have to write you know params dollar sign syntax or or use in in Python you have to at
least like use that object and again if it's not defined then python or R are
just not going to know what to do with an object that's not defined I love to see people are helping
answering each other answer each other's questions on the slido as well
um but one of those I thought might be helpful to just cover here too is yeah
how can we include the current date yeah in the yaml file you don't even have to use sys.date you
can do date equals today date colon today uh one of my favorite uh someone
actually asked me about that this morning just before coming on to the presentation and it's one of my favorite tricks because it's language agnostic so
again you don't have to uh you don't have to do it you know only within using sys.date but
on quarto dates and formatting there's a built-in default for using date equals
today or date equals now or date equals last dot modified so there's multiple
things you can do with it in terms of like did you modify it today did you modify with a specific time as well as
the date or when was the last time the document was modified not rendered right
like those are all separate things and if you just did sys.date it's not as
flexible as some of those things that are just built in free options that are awesome now and I I love those things
okay so Tom you just got really excited about that one so I'm I'm curious dates
and tables are apparently my thing what is your favorite thing that you've learned in switching over to quarto
where you're like wow that was amazing I think the biggest just like kind of
like mind blown like absolute new feature for me was was freeze with projects
um there's so many little bitty things that are cool like date equals today and unified syntax and listings but if I
were to identify one or two and I just thought of two we'll stick with those one is for projects
um freeze equals true which allows me to store the computation for an entire
project like for my personal blog that's five years old I can store the computation from the first article I wrote five years ago and I don't have to re-render that one every time so that's
super powerful for collaboration or for um just like managing documents over
time I dropped a link to Rachel for that one the other one that's kind of just like mind-blowing is website listings
um because you can basically create custom listings and build out like
summaries and cards and um little grid layouts of all this
different type of content manually or from source files and just built very nice looking Pages like just remarkably
complex in a good way um I have to link out to this and we're
going to screen share it so sorry one more but this is the page that gets me excited about custom listings and I just
drop that in the chat if you want to screen share that connect widgets fake one
about listings specifically so the reason why I find this so cool is

like if you look at this page it's a summary of a large end-to-end data science task and we're hosting it on
like rstudio connect but in terms of like there's all these different things it's doing so it provides this kind of background
of where you can say like here's the diagram of all the things that are occurring there's a quarter document there's vetiver there's apis they're all
communicating together and as you navigate down the page uh if you go further to like schedule jobs and things
like that keep going there's tabs and cards and other
different things that allow you to just like move through all these assets you've created like this data science
team put together 15 different things that make up this complex robust data
science project and with listing Pages you're actually able to aggregate these into tables with filters and sorting and
cards and all that's natively built into quarto as like a first class thing so
you don't have to escape out to I've been trying to figure out this one how do you do this I got source code for
that one the answer is lots of yaml but let me grab I'll show you the GitHub
repo so we can share that out as well because this is like the mind-blowing one to me it's just like
all of that I'll drop one more link to you Rachel here in the chat
so that is a public uh repository called test Dash listing
and it goes through basically the idea of building the connect widgets our package only with quarto so you don't
have to use kind of the structure of connect widgets but you can rebuild all that formatting with cards and uh grid
and and tables so I really like this that was like a motivating listings are
really cool you should try them out that's amazing thank you okay
um I'll try to go through just a few more questions I know yes we can't we could stay on here all day but we can't
really so okay [Music] um so going over to John's question on
YouTube is there a Styler for Cordo like an R Studio add-in
so there's a package called BS lib that has
a first class plug-in for exploring themes for our markdown
um I do imagine at some point BS lib theming will probably be available in
quarto but it's not today so my answer is there is not a r Studio add-in as of
this moment in time that might change over time um boot Swatch theme picker I'm just
looking uh uh I believe okay there might be an option for just
like modifying boot Swatch in terms of like those are the 25 themes that are built in to Cordo there's probably an
application that hosts a boot Swatch theme modifier because that's really what's happening is bootstrap CSS boot
swatches the opinionated themes and you can modify them um so no there's not but that's a good idea and we're probably already working on it great thank you
okay maybe we're having too much fun with the polls over here but I just wanted to ask one more over there
um and it doesn't have to be related to quarto but I was just wondering for people listening in what is a challenge
that you've faced in the past few weeks that we could explore creating content
for and it doesn't have to be like code specific or or Cordo it could even be

about like people focus or organization um but just wanted to ask that everybody to everybody as we think about what
would be most helpful for you all um and that's just on the the slido you can access the poll there
okay uh one other question from slido
can you please quickly throw light on how to publish a quarter document to Cordo pub with the yaml attributes
setting file hear more attributes I'm
assuming I don't want to assume too much let me make sure uh
mammal attributes is kind of a loaded term in terms of
we could ask for clarification yeah um if you want to clarify in that one so quarto Pub I would say the the best
resource is that like Porto Pub is is a first class kind of description with the documentation on
quarto.org and it has a published.yaml uh example I think that's probably what
you're talking about in terms of that's a yaml file that allows you to modify settings for a specific quarter Pub
um you could include that around the context of like your existing project for most people
um the way that you would use quarto publish or quarto pubs is you would from the terminal you would do quarto publish
quarto Dash Pub and then the file you want to publish and it will then walk you through interactively the process of
like selecting your credentials and yes you want to name it this and Publishing that
with a quarto pub and the underscore published at yaml or if you're doing
that non-interactively with something like continuous integration or GitHub actions or Azure devops or our studio
connect um you can go further and basically like write some of those ahead of time
because you aren't going to be there to to do those steps
that's a again a pretty big thing to think about so there's a lot there's
like a whole section on publishing on quarto.org that's fairly new so maybe last time you were looking it wasn't
there but it has like description of the underscore published ieml the publish
command as well as the non-interactive publishing with continuous integration
um and I think I yeah I shared that with Rachel and she'll show that in the chat
foreign okay a few other questions one from
slido is there a comprehensive quarto cheat sheet or web page they clarified and said I'm finding bits and pieces of code for the yaml header and code chunks but nothing comprehensive
so the best option if you it's not it's not a cheat sheet at all but I would consider it more of like a reference or
an appendix so if you go to quarto.org docs slash reference or at the top
quarter.org there's a reference header yeah okay perfect
um and then at the top it says reference that fourth item okay
this provides for all the formats and different things you can do what is
possible so these are this is not expanding on it in great detail it's just describing it but let's say for yaml header if you click on documents the HTML the first one uh there
these are all the yaml options you can provide and then they're structured by what is like the grouping
so this is pretty Bare Bones but it like tells you what's possible and what does
it actually mean this is also duplicated so when you're writing quarto documents in our studio
or in vs code with the quarto extension if you do Control Plus space inside the
yaml header it will basically pull up this list and you can start filtering it so you could say like you know start

typing title and it will say like title is something you can auto complete and
describing it as the document title but this reference section gives you a
massive amount of just I want the highlights or the cheat sheet of like all the different uh things that are possible and then if you want to go into like how would I actually use them that's where the guide section would be
more useful okay and that was Control Plus space
yep Control Plus space inside the yaml header or inside a Hash Pipe in the code
chunk will basically like pull up a search bar essentially of like what load is possible that you can do inside our
studio great thank you
I think there are a few questions that touch upon this um to to math and apologies if I
mispronounce your name ask is there going to be a quarto dashboard offering similar to flex dashboard
yes there will be in the future coming soon TBD
okay great let's okay two more questions I promise cool
okay let me just double check on slido the most that we answered the most upvoted ones too
okay another uh yaml question for you one was the yaml file option for quarto
CLI um to set parameters was not mentioned any reason
yeah so uh execute params is a sub command there for
if you think of a qmd and I'm kind of thinking that as like the first thing that I'm working on is like a doc UMD
file um in that context every time you execute the document
it executes it from top to bottom linearly and re-executes everything for Jupiter notebooks using like python
for example a jupyter notebook can be rendered with the stored execution
because a jupyter notebook behind the scenes is actually a DOT Json file that actually stores the computation that
when it was executed um so you can either do that and use
whatever the parameters were that were in the document as is the defaults or if we use dash dash execute Dash params the
one that uh the question has been asked about that will basically like force an execution or reference an external file
in some cases like maybe you don't um you can imagine like writing out dash
param one dash param two dash param 3 Dash param four like continuing on and
through the CLI is a lot of typing so if you use like a a
external params.yaml file that will simplify uh the writing of it because
you have like a file that has all those different parameters defined that you're bringing in
great thank you okay um one other question that was asked on I
think YouTube shoot just lost it okay how do you install Cordo extensions
yep so from the uh there's two options for quarter extensions
um are text files or or zip files so you can literally download it and copy and
paste it over like I just want to say like that's totally possible but
um if you look at the extensions uh page on quarto.org it actually walks through
how to install extensions so from the quarto command line interface you could
do like quarto install extension font awesome or quarto install extension
quarter extension uh video or lightbox and it will install that extension into
a folder in your project um so the quarto command line interface is the correct way to do that but I just want to note that they are literally text files so you can download it and like copy it or move it around between
folders if you wanted to just the quarto command line interface provides a

a helper function to do that efficiently great thank you I realized I just threw
42 minutes of questions at you so really appreciate it Tom thank you for
an awesome talk as well one last question it if people are looking for
like the best way to stay up to date on new quarto features or where to go and
how to like get in touch with the Cordo team what's the best way to do that
so if I were to say stay up to date um quarto has a news let me see if
there's the getting started
all right so on the download page if you want to open this up
um
and screen share that one um so if you wanted to get to there and
we'll start from the beginning if you click on get started at the top of quarto
and you see it has the default it says like Linux Mac Windows you can install if you see that release notes and more
downloads below windows and the table uh yep
if you click on that it'll take you to the download page and then if you scroll down it'll say here's all the things
that have happened in the latest version so you can see that the team's doing a lot of really cool work and adding a lot
of things that's in the current released version and if you scroll back up it also links
out to the um kind of pre-release next version the alpha that's coming out in
the future and that also has uh release notes at the bottom of the page that
we'll talk about like what's coming down the pipeline or what are they working on
um ultimately like because certain things released quickly just like that page is going to be the best uh
broad overview of what's coming and what's happening um but if you do have questions in terms
of like that's a a pull kind of thing like I just want to read about what's happening great I can go there if you're
running into trouble or if you have questions for the dev team at the top of that page under help
it says ask a question that button will take you to uh the
quarto dev GitHub discussions and there you can ask specific questions and the
team can assist or the best thing to do is to search for your question there
I guarantee that for a lot of things other people have asked questions about your topic and you might actually be able to find the answer there before we
can put it into like documentation form or into a new feature so you can search for all sorts of
different things and say like hey there's weird Behavior or I'm not quite understanding and someone may have
already asked that question or the the dev team might have answered that question or said oh this is a problem
we'll we'll turn it into a GitHub issue for example
that's cool I didn't I didn't realize that was right from the court of sight too yeah
cool the short answer to everything is quarto.org has all of your answers or will send you
to the place to ask more questions perfect thank you so much Tom really really
appreciate it just want to let everybody know that this recording is available immediately after the talk ends it's
just at the same YouTube live link and I will try and go through probably not
within the next hour or so but I'm gonna go grab all those links that Tom shared and put them with the recording as well
but you should be able to see them in the chat too but thank you everybody thanks so much
for staying and hanging out with us as well have a great rest of the day thanks everybody thanks Rachel